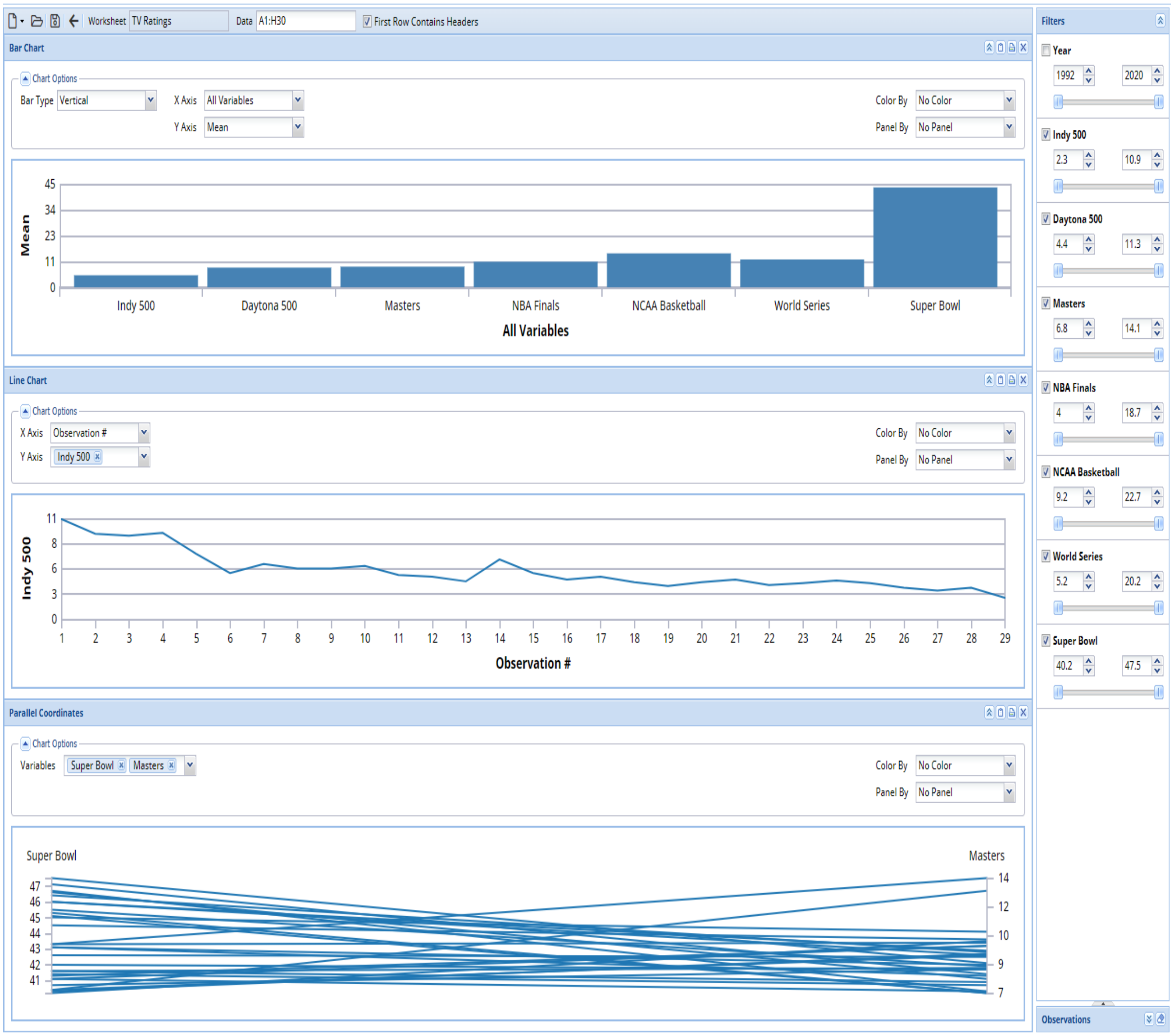
Version 2025 Q4 For Use With Excel 2013-2019

Analytic Solver

Data Science User Guide



Worksheet TV Ratings Data A1:H30 ☑ First Row Contains Headers

Bar Chart

Chart Options

Bar Type Vertical X Axis All Variables Y Axis Mean Color By No Color Panel By No Panel

Line Chart

Chart Options

X Axis Observation # Y Axis Indy 500 Color By No Color Panel By No Panel

Parallel Coordinates

Chart Options

Variables Super Bowl Masters Color By No Color Panel By No Panel

Filters

□ Year 1992 2020

☑ Indy 500 2.3 10.9

☑ Daytona 500 4.4 11.3

☑ Masters 6.8 14.1

☑ NBA Finals 4 18.7

☑ NCAA Basketball 9.2 22.7

☑ World Series 5.2 20.2

☑ Super Bowl 40.2 47.5

Observations

Copyright

Trademarks

Patent Pending

Systems and Methods for Automated Risk Analysis of Machine Learning Models.

Acknowledgements

Thanks to Dan Fylstra and the Frontline Systems development team for a 25-year cumulative effort to build the best possible optimization and simulation software for Microsoft Excel. Thanks to Frontline's customers who have built many thousands of successful applications, and have given us many suggestions for improvements.

Risk Solver Pro and Risk Solver Platform have benefited from reviews, critiques, and suggestions from several risk analysis experts:

- Sam Savage (Stanford Univ. and AnalyCorp Inc.) for Probability Management concepts including SIPs, SLURPs, DISTs, and Certified Distributions.
- Sam Sugiyama (EC Risk USA & Europe LLC) for evaluation of advanced distributions, correlations, and alternate parameters for continuous distributions.
- Savvakis C. Savvides for global bounds, censor bounds, base case values, the Normal Skewed distribution and new risk measures.

How to Order

Contact Frontline Systems, Inc., P.O. Box 4288, Incline Village, NV 89450.
Tel (775) 831-0300 Fax (775) 831-0314 Email info@solver.com Web http://www.solver.com

Table of Contents

Using Help, Licensing and Product Subsets 58

Creating & Deploying a Workflow 73

Deploying a Fitted Model to RASON, Power BI or Tableau 89

Bringing Big Data into Excel Using Apache Spark 109

Fitting a model using Feature Selection 119

Text Mining 133

Exploring a Time Series Dataset 154

Automated Risk Analysis of Machine Learning Models 164

Fitting the Best Model 177

Classifying the Iris Dataset 228

Start Here: Data Science Essentials in V2025 Q4

Getting the Most from This User Guide

Desktop and Cloud Versions

Analytic Solver V2025 Q4 comes in two versions: **Analytic Solver Desktop** – a traditional "COM add-in" that works only in Microsoft Excel for Windows PCs (desktops and laptops), and **Analytic Solver Cloud** – a modern "Office add-in" that works in Excel for Windows and Excel for Macintosh (desktops and laptops), and also in Excel for the Web using Web browsers such as Chrome, FireFox and Safari. Your license gives you access to both versions, and your Excel workbooks and optimization, simulation and data science models work in both versions, no matter where you save them (though OneDrive is most convenient).

Installing the Software

Read the chapter "Installation and Add-Ins" for complete information on installing Analytic Solver Cloud and (if you wish) Analytic Solver Desktop. This chapter also explains how the Cloud and Desktop versions interact when both are installed, and how to install and uninstall both versions.

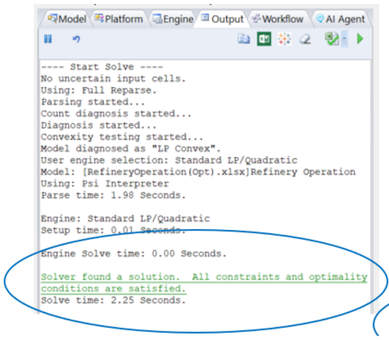
In brief, to add **Analytic Solver Cloud** version to your copy of Excel, you use the Excel Ribbon option **Insert – Get Add-ins** – no Setup program download or installation is required. To install **Analytic Solver Desktop** on Windows PCs, visit www.solver.com , login using the same email and password you used to register on Solver.com, download and run the SolverSetup program.

Understanding License and Upgrade Options

Frontline Solvers V2025 Q4 features product line that gives you access to **all** features, **all** the time for models of all sizes, and a new licensing system, tied to **you** and usable on more than one computer. Read about this in the chapter "Help, Support, Licenses and Product Versions."

Getting Help Quickly

Choose Help on the Ribbon. You'll see several options, starting with **Help – Help Center**. Support Live Chat, Example Models, and User Guides are also available here. In Analytic Solver Desktop (only) you can also get quick online Help by clicking any <u>underlined</u> caption or message in the Task Pane.

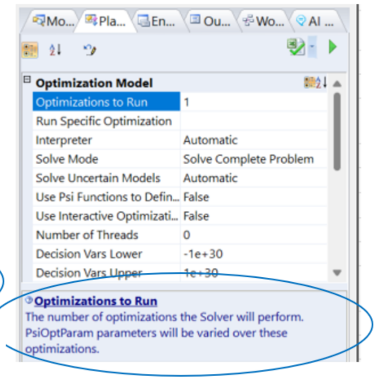


```
---- Start Solve ----
No uncertain input cells.
Using: Full Reparse.
Parsing started...
Count diagnosis started...
Diagnosis started...
Convexity testing started...
Model diagnosed as "LP Convex".
User engine selection: Standard LP/Quadratic
Model: [RefineryOperation(Opt).xlsx]Refinery Operation
Using: Psi Interpreter
Parse time: 1.98 Seconds.

Engine: Standard LP/Quadratic
Setup time: 0.01 Seconds.

Engine Solve time: 0.00 Seconds.

Solver found a solution.  All constraints and optimality
conditions are satisfied.
Solve time: 2.25 Seconds.
```



Optimization Model	
Optimizations to Run	1
Run Specific Optimization	
Interpreter	Automatic
Solve Mode	Solve Complete Problem
Solve Uncertain Models	Automatic
Use Psi Functions to Defin...	False
Use Interactive Optimizati...	False
Number of Threads	0
Decision Vars Lower	-1e+30
Decision Vars Upper	1e+30

Optimizations to Run
The number of optimizations the Solver will perform. PsiOptParam parameters will be varied over these optimizations.

Finding the Examples

Use Help – Examples on the Analytic Solver or Data Science Ribbon to open a list of example optimization and simulation models, and example data sets for data science, that you can open by clicking hyperlinks. See the chapter "Help, Support, Licenses and Product Subsets" for details. Some of these examples are used and described in the **Examples** chapters.

Using Existing Models

Models created using XLMiner 4.0 and earlier can be used in Analytic Solver Data Science without any required changes.

Getting and Interpreting Results

Learn how to interpret Analytic Solver Data Science's result messages, error messages, reports and charts using the Help file imbedded within the software. Simply go to **Help – User Guides** on the Data Science ribbon.

Installation and Add-Ins

What You Need

You can use Analytic Solver Cloud in Excel for the Web (formerly Excel Online) through a Web browser (such as Edge, Chrome, Firefox or Safari), without installing anything else. This is the simplest and most flexible option, but it requires a constant Internet connection.

To use Analytic Solver Cloud in Excel Desktop on a PC or Mac, you must have a current version of Windows or iOS installed, and **you will need the latest Excel version installed via your Office 365 subscription** – older non-subscription versions, even Excel 2019, do not have all the features and APIs needed for modern JavaScript add-ins like Analytic Solver Cloud.

To use Analytic Solver Desktop (Windows PCs only), you must have first installed Microsoft Excel 2013, 2016, 2019, or the latest Office 365 version on Windows 10, Windows 8, Windows 7, or Windows Server 2019, 2016 or 2012. (Windows Vista or Windows Server 2008 may work but are no longer supported.). It's not essential to have the standard Excel Solver installed.

Installing the Software

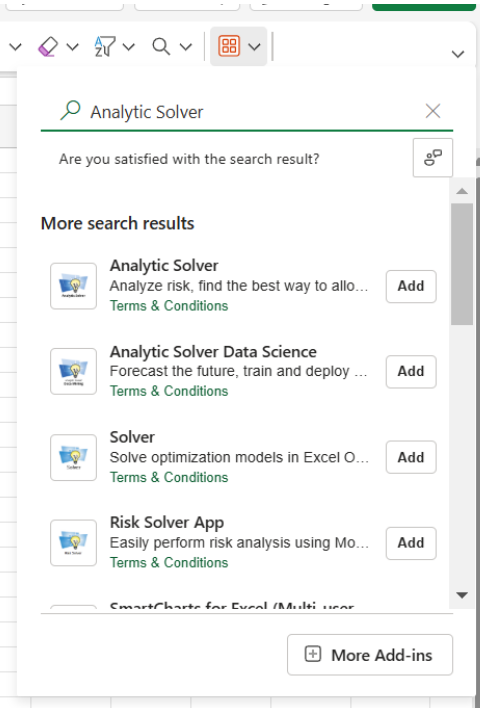
Installing Analytic Solver Cloud

Analytic Solver V2025 Q4 includes our next-generation offering, Analytic Solver Cloud – usable in the latest versions of desktop Excel for Windows and Macintosh, and in Excel for the Web. Analytic Solver Cloud is divided into **two** add-ins that work closely together (since a JavaScript add-in currently can have only one Ribbon tab): the **Analytic Solver** add-in builds optimization, simulation and decision table models, and the **Data Science** add-in builds data science or forecasting models.

Both the Analytic Solver and Data Science add-ins support existing models created in previous versions of Analytic Solver. Your license for Analytic Solver will allow you to use Analytic Solver Desktop in desktop Excel or Analytic Solver Cloud in either desktop Excel (latest version) *or* Excel for the Web.

Once you do this, the Analytic Solver and Data Science tabs will appear on the Ribbon in each new workbook you use. To use the Analytic Solver and Data Science add-ins, **you must first "insert" them** for use in your licensed copy of desktop Excel or Excel for the Web, while you are logged into your Office 365 account.

To insert the add-ins for the first time, open desktop Excel (latest version) or Excel for the Web, click the **Home** tab on the Ribbon, then click the down arrow next to the **Office Add-ins** button.



Analytic Solver

Are you satisfied with the search result?

More search results

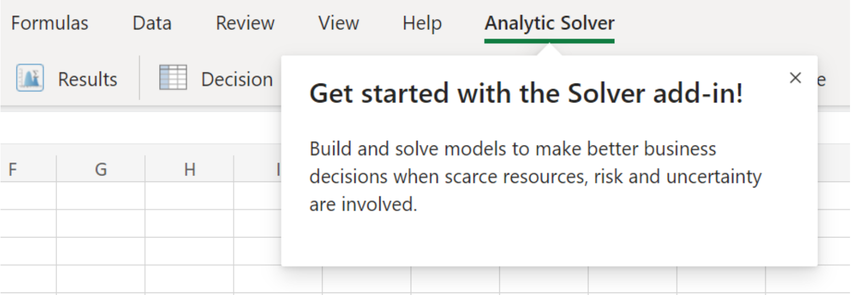
Analytic Solver
Analyze risk, find the best way to allo…
Terms & Conditions

Analytic Solver Data Science
Forecast the future, train and deploy …
Terms & Conditions

Solver
Solve optimization models in Excel O…
Terms & Conditions

Risk Solver App
Easily perform risk analysis using Mo…
Terms & Conditions

Type "Analytic Solver" in the search field. The Analytic Solver add-in will appear at the top of the search results, click **Add**. After a moment, you should see the **Analytic Solver** tab appear on the Ribbon, with a note about how to "Get started with the Solver add-in!", as shown below.

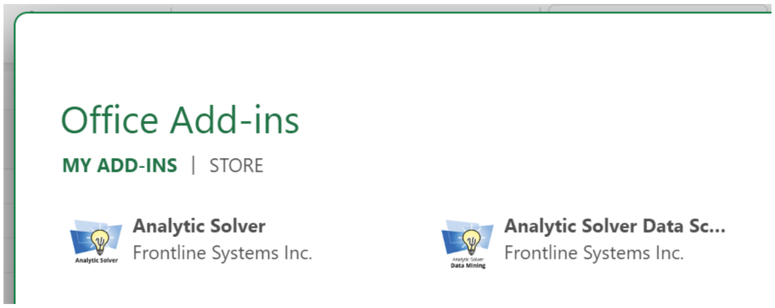


Get started with the Solver add-in!

Build and solve models to make better business decisions when scarce resources, risk and uncertainty are involved.

Repeat these steps to search for, locate and **Add** the Analytic Solver Data Science add-in. After a moment, you should see the **Data Science** tab appear on the Ribbon, with a similar "Get started" note.

After you perform these steps (one time) to insert the Analytic Solver and Analytic Solver Data Science add-ins, they will appear under "My Add-ins". If you ever need to remove the add-ins, click the "**…**" symbol to the right of the add-in name, then click the **Remove** choice on the dropdown menu that appears.



Office Add-ins

Analytic Solver
Frontline Systems Inc.

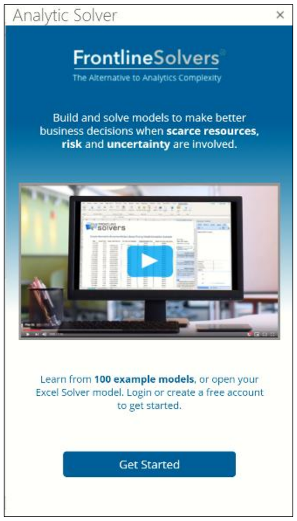
Analytic Solver Data Sc…
Frontline Systems Inc.

Note: Data Science Cloud will defer to Analytic Solver Desktop. If you find that Data Science Cloud does not appear after following these steps, you'll first need to remove Analytic Solver Desktop before installing the Cloud app. To do so simply open Excel and click File – Options – Add-ins – Com Add-ins – Go, select Analytic Solver Add-in, then click Remove. Once Analytic Solver Desktop is removed, then go back to Office Add-ins – My Add-ins and add in Data Science Cloud.

Single Sign On Functionality

Analytic Solver Cloud includes Single Sign On functionality which automatically logs in users to their Analytic Solver Cloud account using their Microsoft 365 credentials. This means that if you've signed in to your Microsoft 365 account using the same email address you used to register on www.solver.com, then you will not be asked to login to Analytic Solver Cloud. Once you insert Analytic Solver cloud, you will have immediate access to all the features and functionality of Analytic Solver. As long as you remain signed in to your Microsoft 365 account, you'll never have to login to Analytic Solver Cloud again!

If you log out of Analytic Solver or your Office 365 credentials do not match your Solver credentials, you'll see the following welcome screen in the Solver task pane. Just click the Get Started button or click License – Login/Logout on the ribbon to login.



Analytic Solver ×

FrontlineSolvers

The Alternative to Analytics Complexity

Build and solve models to make better business decisions when **scarce resources**, **risk** and **uncertainty** are involved.

Learn from **100 example models**, or open your Excel Solver model. Login or create a free account to get started.

Get Started

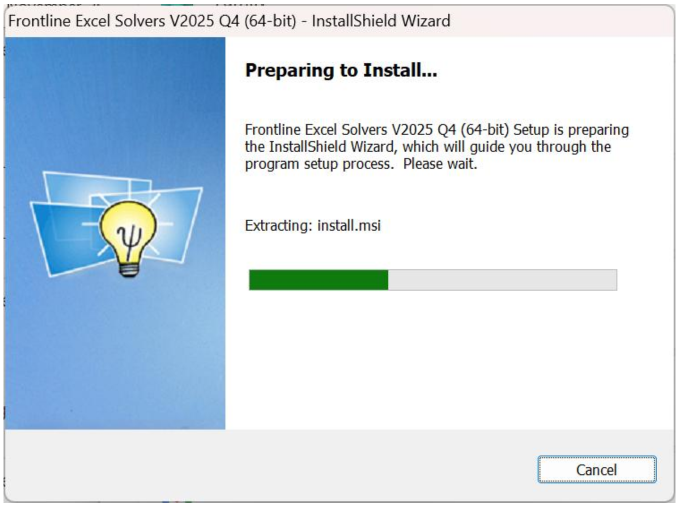
Installing Analytic Solver Desktop

To install Analytic Solver Desktop to work with any of the supported versions of Microsoft Excel (see above), simply run the program **SolverSetup.exe**, which installs all of the Solver program, Help, User Guide, and example files. SolverSetup.exe checks your system, detects what version of Office you are running (32-bit or 64-bit) and then downloads and runs the appropriate Setup program version.

Note that your copy of the Setup program will usually have a filename such as SolverSetup_12345.exe; the '12345' is your user account number on Solver.com.

When you run the Setup program, depending on your antivirus program or Windows security settings, you might be prompted with a message **"[reason**

such as new/unknown program]. Are you sure you want to run this software?" You may safely click **Run** in response to this message.



Frontline Excel Solvers V2025 Q4 (64-bit) - InstallShield Wizard

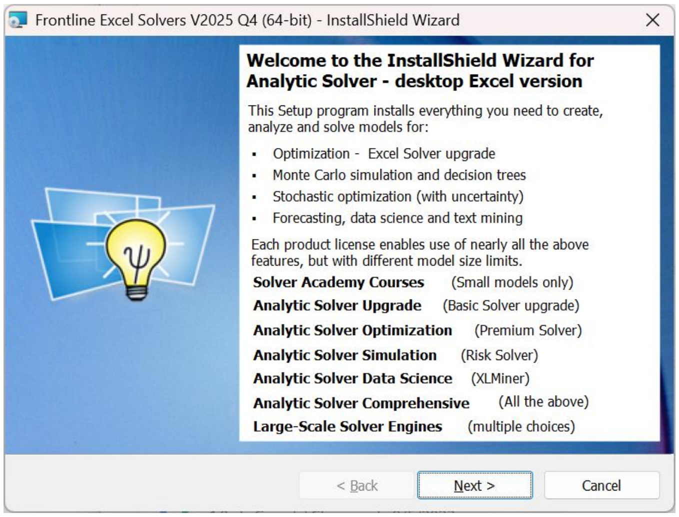
Preparing to Install...

Frontline Excel Solvers V2025 Q4 (64-bit) Setup is preparing the InstallShield Wizard, which will guide you through the program setup process. Please wait.

Extracting: install.msi

Next, you'll briefly see the standard Windows Installer dialog.

Then a dialog box like the one shown below should appear:



Frontline Excel Solvers V2025 Q4 (64-bit) - InstallShield Wizard

Welcome to the InstallShield Wizard for Analytic Solver - desktop Excel version

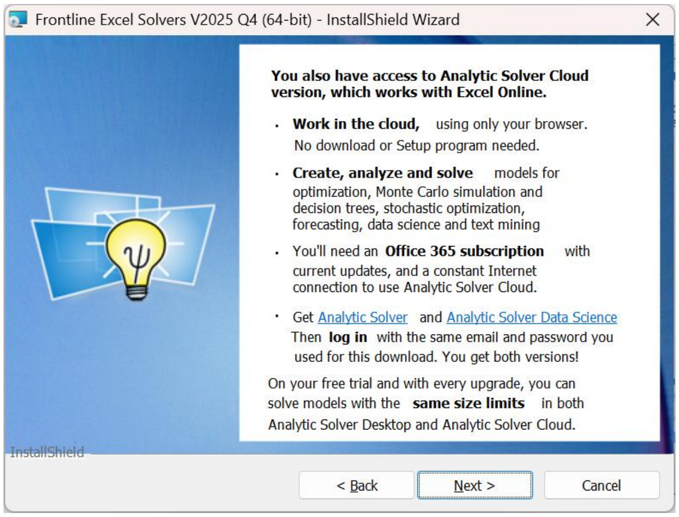
This Setup program installs everything you need to create, analyze and solve models for:

- Optimization - Excel Solver upgrade
- Monte Carlo simulation and decision trees
- Stochastic optimization (with uncertainty)
- Forecasting, data science and text mining

Each product license enables use of nearly all the above features, but with different model size limits.

Solver Academy Courses	(Small models only)
Analytic Solver Upgrade	(Basic Solver upgrade)
Analytic Solver Optimization	(Premium Solver)
Analytic Solver Simulation	(Risk Solver)
Analytic Solver Data Science	(XLMiner)
Analytic Solver Comprehensive	(All the above)
Large-Scale Solver Engines	(multiple choices)

Read this, so you know the *difference* between Analytic Solver Comprehensive and it's subsets. Then click **Next** to proceed – you'll see a dialog like the one below.

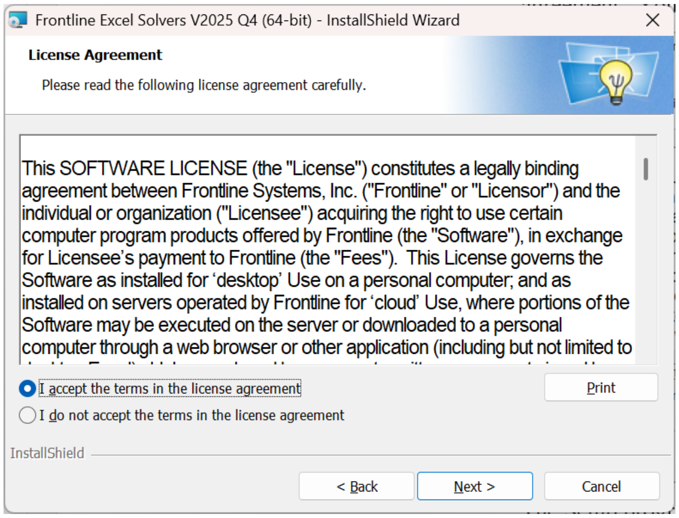

Frontline Excel Solvers V2025 Q4 (64-bit) - InstallShield Wizard

You also have access to Analytic Solver Cloud version, which works with Excel Online.

- **Work in the cloud,** using only your browser. No download or Setup program needed.
- **Create, analyze and solve** models for optimization, Monte Carlo simulation and decision trees, stochastic optimization, forecasting, data science and text mining
- You'll need an **Office 365 subscription** with current updates, and a constant Internet connection to use Analytic Solver Cloud.
- Get Analytic Solver and Analytic Solver Data Science Then **log in** with the same email and password you used for this download. You get both versions!

On your free trial and with every upgrade, you can solve models with the **same size limits** in both Analytic Solver Desktop and Analytic Solver Cloud.

Next, the Setup program will ask if you accept Frontline's software license agreement. You must click "I accept" and **Next** in order to be able to proceed.


Frontline Excel Solvers V2025 Q4 (64-bit) - InstallShield Wizard

License Agreement

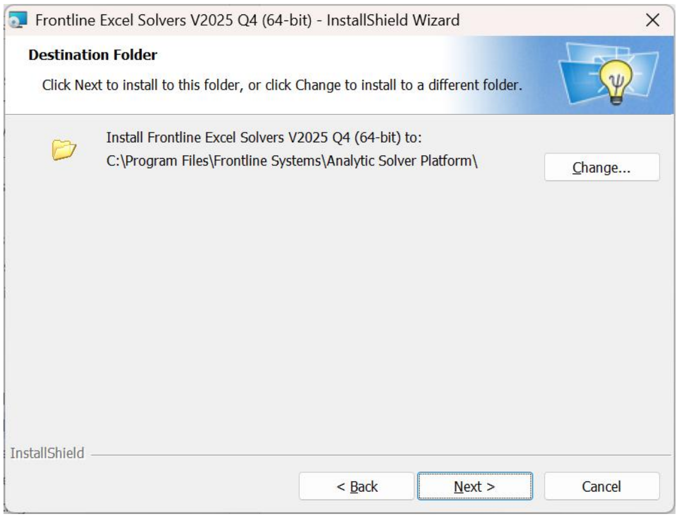
Please read the following license agreement carefully.

This SOFTWARE LICENSE (the "License") constitutes a legally binding agreement between Frontline Systems, Inc. ("Frontline" or "Licensor") and the individual or organization ("Licensee") acquiring the right to use certain computer program products offered by Frontline (the "Software"), in exchange for Licensee's payment to Frontline (the "Fees"). This License governs the Software as installed for 'desktop' Use on a personal computer; and as installed on servers operated by Frontline for 'cloud' Use, where portions of the Software may be executed on the server or downloaded to a personal computer through a web browser or other application (including but not limited to

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

The Setup program then displays a dialog box like the one shown below, where you can select or confirm the folder to which files will be copied (normally C:\Program Files\Frontline Systems\Analytic Solver Platform, or if you're installing Analytic Solver for 32-bit Excel on 64-bit Windows, C:\Program Files (x86)\Frontline Systems\Analytic Solver Platform). Click **Next** to proceed.



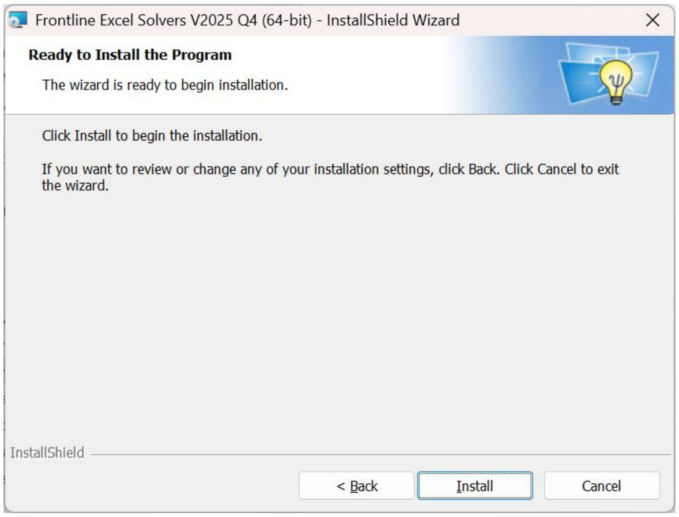
Frontline Excel Solvers V2025 Q4 (64-bit) - InstallShield Wizard

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install Frontline Excel Solvers V2025 Q4 (64-bit) to:
C:\Program Files\Frontline Systems\Analytic Solver Platform\

You'll see a dialog confirming that the preliminary steps are complete, and the installation is ready to begin.



Frontline Excel Solvers V2025 Q4 (64-bit) - InstallShield Wizard

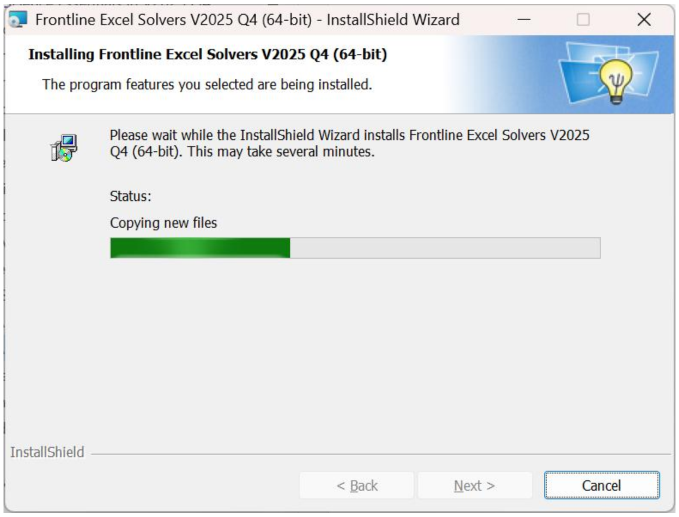
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

After you click **Install**, the Analytic Solver files will be installed, and the program file RSPAddin.xll will be registered as a COM add-in (which may take some time). A progress dialog may appear; be patient, since this process could take longer than it has in previous Solver Platform releases.



Frontline Excel Solvers V2025 Q4 (64-bit) - InstallShield Wizard

Installing Frontline Excel Solvers V2025 Q4 (64-bit)

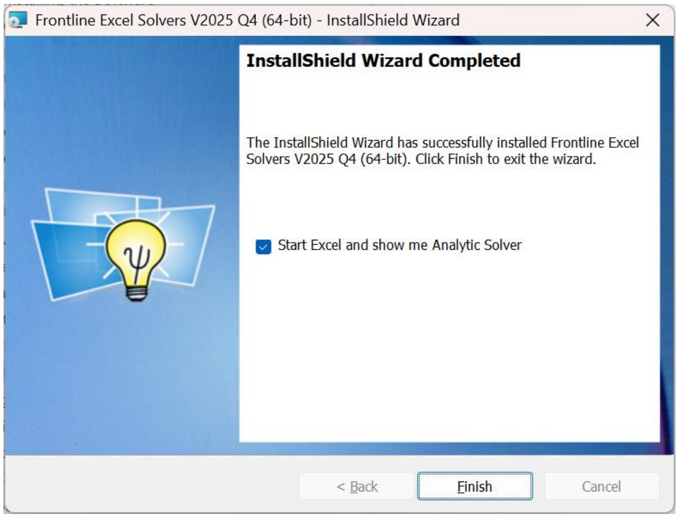
The program features you selected are being installed.

Please wait while the InstallShield Wizard installs Frontline Excel Solvers V2025 Q4 (64-bit). This may take several minutes.

Status:

Copying new files

When the installation is complete, you'll see a dialog box like the one below. Click **Finish** to exit the installation wizard.



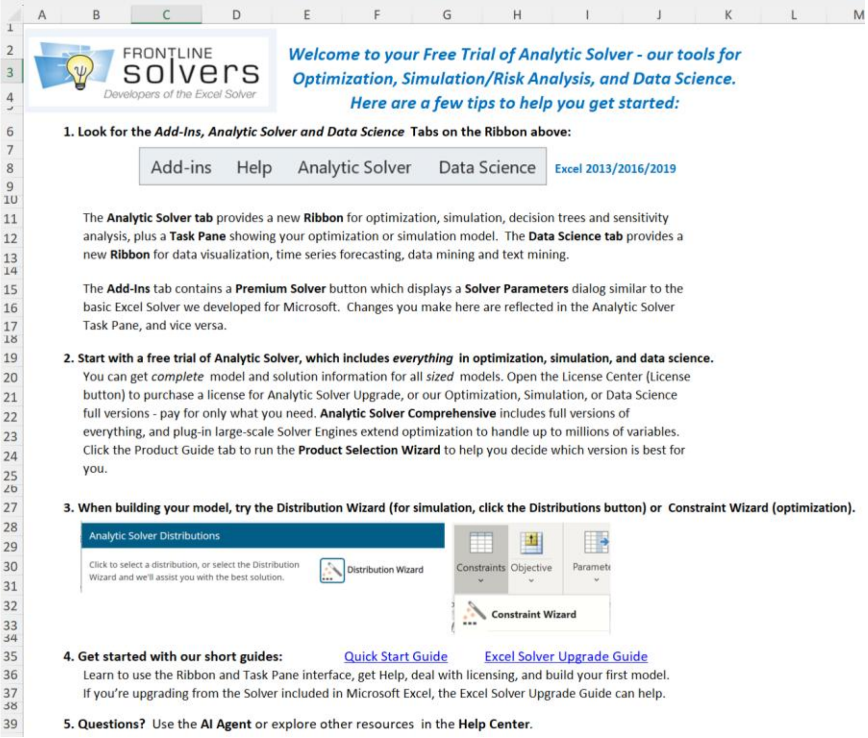
Frontline Excel Solvers V2025 Q4 (64-bit) - InstallShield Wizard

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed Frontline Excel Solvers V2025 Q4 (64-bit). Click Finish to exit the wizard.

☑ Start Excel and show me Analytic Solver

The full Analytic Solver product family is now installed. With your trial and paid license, you can access every feature of the software, including forecasting and data science, simulation and risk analysis, and conventional and stochastic optimization. Simply click "Finish" and Microsoft Excel will launch with a Welcome workbook containing information to help you get started quickly.



Welcome to your Free Trial of Analytic Solver - our tools for Optimization, Simulation/Risk Analysis, and Data Science. Here are a few tips to help you get started:

1. Look for the *Add-Ins, Analytic Solver and Data Science* Tabs on the Ribbon above:

The **Analytic Solver tab** provides a new **Ribbon** for optimization, simulation, decision trees and sensitivity analysis, plus a **Task Pane** showing your optimization or simulation model. The **Data Science tab** provides a new **Ribbon** for data visualization, time series forecasting, data mining and text mining.

The **Add-Ins** tab contains a **Premium Solver** button which displays a **Solver Parameters** dialog similar to the basic Excel Solver we developed for Microsoft. Changes you make here are reflected in the Analytic Solver Task Pane, and vice versa.

2. Start with a free trial of Analytic Solver, which includes *everything* in optimization, simulation, and data science.
You can get *complete* model and solution information for all *sized* models. Open the License Center (License button) to purchase a license for Analytic Solver Upgrade, or our Optimization, Simulation, or Data Science full versions - pay for only what you need. **Analytic Solver Comprehensive** includes full versions of everything, and plug-in large-scale Solver Engines extend optimization to handle up to millions of variables. Click the Product Guide tab to run the **Product Selection Wizard** to help you decide which version is best for you.

3. When building your model, try the Distribution Wizard (for simulation, click the Distributions button) or Constraint Wizard (optimization).

Click to select a distribution, or select the Distribution Wizard and we'll assist you with the best solution.

4. Get started with our short guides: Quick Start Guide Excel Solver Upgrade Guide
Learn to use the Ribbon and Task Pane interface, get Help, deal with licensing, and build your first model. If you're upgrading from the Solver included in Microsoft Excel, the Excel Solver Upgrade Guide can help.

5. Questions? Use the **AI Agent** or explore other resources in the **Help Center**.

Logging in the First Time

In Analytic Solver V2025 Q4, your license is associated with **you**, and may be used on **more than one PC**. For example, you can run SolverSetup to install the desktop software on your office PC, your company laptop, and your PC at home. But only **you** can use Analytic Solver, and only on **one** of these computers **at a time**. It is unlawful to "share" your license with another human user.

The first time you run Analytic Solver (Desktop or Cloud) after installing the software on a new computer, when you next start Excel and visit the Analytic Solver tab on the Ribbon, **you will be prompted to login**. Enter the **email address** and **password** that you used to register on Solver.com. Once you've done this in Analytic Solver Desktop, your identity will be "remembered," so you won't have to login every time you start Excel and go to one of the Analytic Solver tabs. In Analytic Solver Cloud, you may be asked to login more frequently.

You can login and logout at any time, by visiting the **Solver Home tab** (see "Cloud Version and Solver Home Tab" below) and clicking the **Log In** or **Log Out** button in Analytic Solver Desktop or by clicking **Help – Login/Logout** in Analytic Solver Cloud. If you share use of a single physical computer with other Analytic Solver users, be careful to **login** with your own email and password, and **log out when you're done** – if you don't, other users could access private files in your cloud account, or use up your allotted CPU time or storage.

When you move from one computer to another, you should **log out** on one and **log in** on the other. As a convenience, if you log in to Analytic Solver on a new computer when you haven't logged out on the old computer, Analytic Solver will let you know, and offer to automatically log you out on the other computer.

Uninstalling the Software

To uninstall Analytic Solver Desktop, just run the **SolverSetup** program as outlined above. You'll be asked to confirm that you want to remove the software.

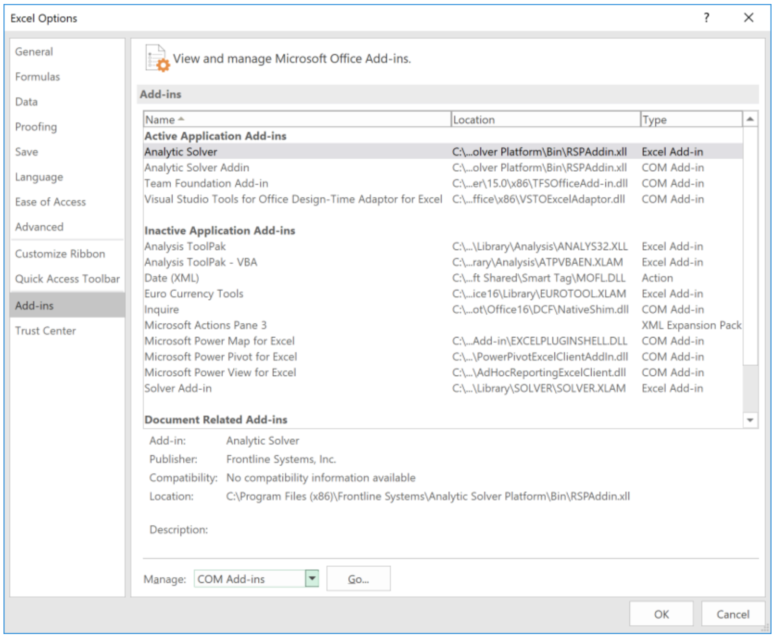
You can also uninstall by choosing **Control Panel** from the **Start** menu, and double-clicking the **Programs and Features** or **Add/Remove Programs** applet. In the list box below "Currently installed programs," scroll down if necessary until you reach the line, "Frontline Excel Solvers V2025 Q4," and click the Uninstall/Change or Add/Remove… button. Click OK in the confirming dialog box to uninstall the software.

Activating and Deactivating the Software

Analytic Solver Desktop's main program file RSPAddin.xll is a COM add-in, an XLL add-in, and a COM server. A reference to the add-in **Solver.xla** is needed if you wish to use the "traditional" VBA functions to control Analytic Solver, instead of its new VBA Object-Oriented API.

Excel 2019, 2016 and 2013

In Excel 2019, 2016 and 2013, you can manage all types of add-ins from one dialog, reached by clicking **File – Options -- Addins**.



View and manage Microsoft Office Add-ins.

Add-ins

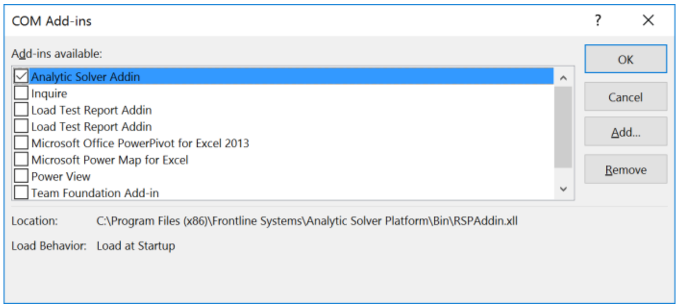
Name	Location	Type
Active Application Add-ins		
Analytic Solver	C:\...olver Platform\Bin\RSPAddin.xll	Excel Add-in
Analytic Solver Addin	C:\...olver Platform\Bin\RSPAddin.xll	COM Add-in
Team Foundation Add-in	C:\...er\15.0\x86\TFSOfficeAdd-in.dll	COM Add-in
Visual Studio Tools for Office Design-Time Adaptor for Excel	C:\...ffice\x86\VSTOExcelAdaptor.dll	COM Add-in
Inactive Application Add-ins		
Analysis ToolPak	C:\...\Library\Analysis\ANALYS32.XLL	Excel Add-in
Analysis ToolPak - VBA	C:\...rary\Analysis\ATPVBAEN.XLAM	Excel Add-in
Date (XML)	C:\...ft Shared\Smart Tag\MOFL.DLL	Action
Euro Currency Tools	C:\...ice16\Library\EUROTOOL.XLAM	Excel Add-in
Inquire	C:\...ot\Office16\DCF\NativeShim.dll	COM Add-in
Microsoft Actions Pane 3		XML Expansion Pack
Microsoft Power Map for Excel	C:\...Add-in\EXCELPLUGINSHELL.DLL	COM Add-in
Microsoft Power Pivot for Excel	C:\...\PowerPivotExcelClientAddIn.dll	COM Add-in
Microsoft Power View for Excel	C:\...\AdHocReportingExcelClient.dll	COM Add-in
Solver Add-in	C:\...\Library\SOLVER\SOLVER.XLAM	Excel Add-in
Document Related Add-ins		

Add-in: Analytic Solver
Publisher: Frontline Systems, Inc.
Compatibility: No compatibility information available
Location: C:\Program Files (x86)\Frontline Systems\Analytic Solver Platform\Bin\RSPAddin.xll

Description:

Manage: COM Add-ins

You can manage add-ins by selecting the type of add-in from the dropdown list at the bottom of this dialog. For example, if you select COM Add-ins from the dropdown list and clock the Go button, the dialog shown below appears.



COM Add-ins

Add-ins available:

☑ Analytic Solver Addin
□ Inquire
□ Load Test Report Addin
□ Load Test Report Addin
□ Microsoft Office PowerPivot for Excel 2013
□ Microsoft Power Map for Excel
□ Power View
□ Team Foundation Add-in

Location: C:\Program Files (x86)\Frontline Systems\Analytic Solver Platform\Bin\RSPAddin.xll

Load Behavior: Load at Startup

If you uncheck the box next to “Analytic Solver Addin” and click OK, you will deactivate the Analytic Solver COM add-in, which will remove the Analytic Solver tab from the Ribbon in desktop Excel, and also remove the PSI functions for optimization from the Excel Function Wizard.

Analytic Solver Data Science Overview

Overview

This Guide shows you how to use **Analytic Solver Data Science Desktop** or **Cloud** (formerly XLMiner or more recently Analytic Solver Data Mining) that combines the capabilities of data analysis, time series analysis, classification techniques and prediction techniques. Analytic Solver Data Science is included in Analytic Solver Comprehensive, or can be purchased as a stand-alone license.

Analytic Solver Data Science was originally developed and marketed by others (Cytel Corp and Statisstics.com), and was in popular use for over a decade, primarily in teaching, when Frontline Systems acquired all rights to the product in 2011. In 2012-2013, Frontline marketed and supported the existing product, while rewriting the underlying software from the ground up.

- In our V2014 release, we introduced a fundamental new capability in Analytic Solver Platform, Risk Solver Platform and Premium Solver Platform for building optimization and simulation models in Excel: Dimensional Modeling. It introduces new concepts such as dimensions and cubes, and provides all the tools you need to build and solve larger scale, better structured, more maintainable models using these concepts.
- In our V2014-R2 release, Analytic Solver Platform included a completely re-engineered, far more powerful data science and forecasting capability named XLMiner Platform. New data science algorithms are up to 100 times faster, constantly exploit multiple processor cores, and offer greater accuracy and numeric stability.
- Our V2015 release introduced a wide range of new features, including powerful text mining and ensemble methods for classification and prediction in XLMiner Platform; feature selection, partitioning "on-the-fly," ROC/RROC curves, enhanced linear and logistic regression, and more in XLMiner Pro and Platform; extensive chart enhancements, distribution fitting, and new Six Sigma functions in Risk Solver Pro and Platform; and support for "publishing" optimization and simulation models to Excel for the Web (formerly Excel Online) and Google Sheets. This feature was further extended to handle large-scale models in V2015-R2.
- Our V2015-R2 release made it easy to share results in the cloud: You can transfer optimization, simulation, or data mining results into your Microsoft Power BI online account, visualize those results with just a few clicks, and share them with others. Similarly, you can export optimization, simulation or data mining results into Tableau Data Extract (*.tde) files that can be opened in Tableau, the popular interactive data visualization software. V2015-R2 also links your Excel workbook with "Big Data": You can easily obtain sampled and summarized results drawn from the largest datasets, stored across many hard disks, in compute clusters running Apache Spark.

V2016 Release

Our V2016 release dramatically speeds the process of moving from an "analytic Excel model" to a "deployed application, available to others." With the new **Create App** feature, which translates your Excel optimization or simulation model into Frontline's new **RASON** modeling language, you can create an application that can run in a **web browser**, or a **mobile app** for phones or tablets – with just two mouse clicks! Your app solves problems via our RASON server, running 24x7 on Microsoft Azure, using its REST API.

Other enhancements in V2016 include a Task Pane **navigator for data mining** in XLMiner Pro and Platform, and a greatly enhanced **Evolutionary Solver** in Premium Pro and Platform, and Risk Solver Pro and Platform. A completely new local search algorithm called **SQP-GS** (Sequential Quadratic Programming with Gradient Sampling), and new **Feasibility Pump** methods for both continuous and integer variables help the Evolutionary Solver find better solutions, faster than ever.

V2016-R2 Release

In our V2016-R2 release, simulation models can use **compound distributions**, with either a constant or a discrete distribution as the 'frequency' element, and **correlation using copulas** (Gaussian, Student and Archimedean forms), as well as rank-order correlation, to generate samples for multiple uncertain variables.

In optimization, the **Evolutionary Solver** is further enhanced with new 'GA methods' for integer variables, often yielding much better integer solutions than previous releases in a given amount of time. And users of **Dimensional Modeling** will see major improvements in speed and memory use, thanks to new support for 'sparse cubes'. Export of analytic model results to Tableau data visualization software is more convenient than ever in V2016-R2 with support for the **Tableau Web Data Connector** introduced in Tableau 9.1.

V2016-R3 Release

With our V2016-R3 release for Excel, we're introducing **AnalyticSolver.com**, a new cloud-based platform for both predictive and prescriptive analytics models that you can use via a **web browser** – including all the optimization, simulation, and data mining power found in our desktop products. Your models are solved using our Azure-based RASON cloud servers.

The AnalyticSolver.com **user interface** works just like our Excel user interface, with a Ribbon, Analytic Solver Platform and XLMiner Platform tabs with the same icons and dropdown menus, and a Task Pane with Model, Platform, Engine and Output tabs. Both V2016-R3 in Excel and AnalyticSolver.com include a new "Solver Home" tab on the Ribbon that makes it easy to move Excel workbooks and other files between desktop and cloud. And access to AnalyticSolver.com is ***included*** with your V2016-R3 license for desktop Excel.

V2017 Release

In V2017, we are introducing commercial users to **AnalyticSolver.com**, a new cloud-based platform for both predictive and prescriptive analytics models that you can use via a **web browser** – including all the optimization, simulation, and data mining power found in the desktop version. The AnalyticSolver.com **user interface** works just like our Excel user interface, with a Ribbon and Task Pane. Both V2017 in Excel and AnalyticSolver.com include a new "Solver Home" tab on the Ribbon that makes it easy to move Excel workbooks and other files between desktop and cloud. And access to AnalyticSolver.com is ***included*** with your V2017 license for desktop Excel.

V2017 also uses a **new licensing system** that offers you more flexible ways to use the software, both desktop and cloud. Your license is associated with **you**, and may be used on **more than one PC**. For example, you can install the software on your office PC, your company laptop, and your PC at home. But only **you** can use Analytic Solver, and only on **one** of these computers **at a time**.

V2017 introduces **Analytic Solver Basic**, as described above, to give you access to **all** Analytic Solver features, **all** the time, for learning purposes using small models. It also includes a new **License/Subscription Manager** and a **Product Selection Wizard** that makes it much easier to upgrade or change your license subscription on a self-service basis, and a new **Test Run/Summary** feature that lets you see exactly how your model will run with an Analytic Solver upgrade, even a plug-in large-scale Solver Engine, before you purchase the upgrade – and do this any time, not just during a 15-day free trial.

V2017 includes major enhancements to **data mining**: Automatic support for **categorical variables** in many classification and prediction algorithms that 'normally' require continuous variables; **ensembles** that combine nearly any type of algorithm as a 'weak learner', not just for example trees; general-purpose **Rescaling** as a new Data Transformation method that can also be applied 'on-the-fly' when training a model; greatly enhanced multilayer **neural networks**; ability to export models in **PMML**; and many report and chart enhancements.

The V2017 **Evolutionary Solver** includes another set of major enhancements in its handling of non-smooth models with integer variables – enough so that *most* such models will solve *significantly* faster. And there's support for the Tableau Web Data Connector 2.0, and a new SolverSetup program that *automatically* installs the correct **32-bit or 64-bit** version of the software.

V2017-R2 Release

V2017-R2 includes major enhancements to Monte Carlo simulation/risk analysis and optimization. It's now possible to **fit copula** parameters to historical data – a complement to distribution fitting that is sometimes called "correlation fitting." You can use a new family of probability distributions, called the **Metalog distributions**, even more general than the Pearson distributions – members of the family can be chosen based directly on historical data (even just a few observations), without a distribution fitting process.

The V2017-R2 PSI Interpreter includes major **speed enhancements** for large linear and nonlinear **optimization models** – users with large models are likely to see a dramatic speedup in "Setting Up Problem…" Also part of this release are new, higher performance versions of the Gurobi Solver Engine (based on Gurobi

7.5), the Xpress Solver Engine (based on Xpress 30.1), and the Knitro Solver Engine (based on Knitro 10.3).

Creating Power BI Custom Visuals

The most exciting new feature of V2017-R2 is the ability to turn your Excel-based optimization or simulation model into a **Microsoft Power BI Custom Visual**, with just a few mouse clicks! Where others must learn JavaScript (or TypeScript) programming and a whole set of Web development tools to even begin to create a Custom Visual, you'll be able to create one right away.

You simply select rows or columns of data to serve as changeable parameters, then choose **Create App – Power BI**, and save the file created by V2017-R2. You click the Load Custom Visual icon in Power BI, and select the file you just saved. What you get isn't just a chart – it's your *full optimization or simulation model*, ready to accept Power BI data**, run on demand** on the web, and display visual results in Power BI! You simply need to drag and drop appropriate Power BI datasets into the "well" of inputs to match your model parameters.

How does *that* work? The secret is that V2017-R2 translates your Excel model into **RASON**® (RESTful Analytic Solver Object Notation, embedded in JSON), then "wraps" a JavaScript-based Custom Visual around the RASON model. See the chapter "Creating Your Own Application" for full details!

V2018 Release

V2018 extends Analytic Solver's forecasting and data mining features with a new capability called **data mining workflows** that can save a lot of time and eliminate repetitive steps. You can combine nearly any of Analytic Solver's data retrieval, data transformation, forecasting and data mining methods into a single, all-inclusive workflow, or pipeline.

Using the new Workflow tab in the Task Pane, you can either **"drag and drop"** icons onto a "canvas" to create a workflow diagram, or you can simply turn on a **workflow recorder**, carry out the steps as you've always done by choosing menu options and dialog selections, and the workflow diagram will be created automatically. Once the diagram or pipeline is created, you can "run" it in one step – each data mining method in the workflow will be executed in sequence.

In previous releases, you could use the trained model from a *single* data mining method (such as a Classification Tree or Neural Network) to "score" new data, by mapping features (columns) between the training set and new data set. In V2018, you can apply an *entire workflow* – including data transformations, partitioning, model training, and more – to a new dataset, by mapping features (columns) between the dataset used to create the workflow, and a new dataset.

Creating Tableau Dashboard Extensions

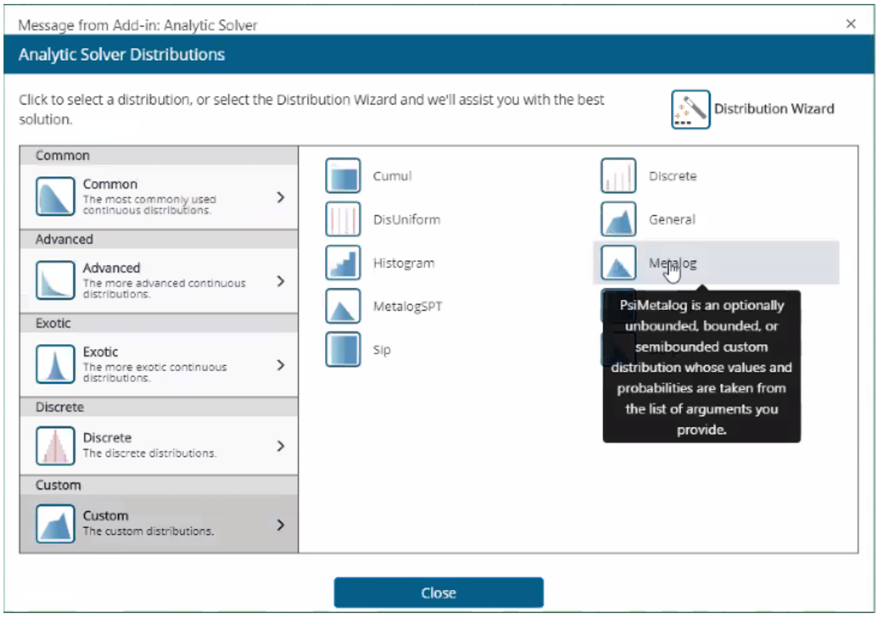
Another exciting new feature of V2018 is the ability to turn your Excel-based optimization or simulation model into a **Tableau Dashboard Extension**, with just a few mouse clicks! This is quite similar to the ability to create Power BI Custom Visuals introduced in Analytic Solver V2017-R2. It works (only) with Tableau version 2018.2 or later.

You simply select rows or columns of data to serve as changeable parameters, then choose **Create App – Tableau**, and save the file created by V2018. In Tableau, drag the **Extensions** object onto your dashboard, and choose the file you just saved. You'll be prompted to match the parameters your model needs with data in Tableau. What you get isn't just a chart – it's your *full optimization* *or simulation model*, ready to accept Tableau data**, run on demand** (using our **RASON** server), and display visual results in Tableau!

V2019 Release

In Frontline Solvers V2019, we are introducing **Analytic Solver Cloud** – a next-generation product that's the result of five years of development using new cloud technologies, that we can now bring to you – since Microsoft has released a complete set of JavaScript APIs for new Excel features, such as functions like PsiNormal() and PsiMean() used in simulation and risk analysis.

While we have new and enhanced features in development, in the V2019 release we have focused on a **consistent user experience** between Analytic Solver Desktop and Analytic Solver Cloud. In a few cases, this has involved modifications to Analytic Solver Desktop. For example, older versions of Analytic Solver Desktop used "cascading submenus" to select probability distributions, and results in Monte Carlo simulations. Since a JavaScript add-in cannot define or use cascading submenus, we have modified Analytic Solver Desktop *and* Analytic Solver Cloud so that the **Distributions** button on the Ribbon displays a dialog, rather than a cascading submenu, where you can select an appropriate probability distribution. Yet the order and layout of probability distributions remains the *same* as in previous Analytic Solver versions.



Message from Add-in: Analytic Solver

Analytic Solver Distributions

Click to select a distribution, or select the Distribution Wizard and we'll assist you with the best solution.

PsiMetalog is an optionally unbounded, bounded, or semibounded custom distribution whose values and probabilities are taken from the list of arguments you provide.

×
Distribution Wizard
Common
Common
The most commonly used continuous distributions.
Advanced
Advanced
The more advanced continuous distributions.
Exotic
Exotic
The more exotic continuous distributions.
Discrete
Discrete
The discrete distributions.
Custom
Custom
The custom distributions.
Cumul
DisUniform
Histogram
MetalogSPT
Sip
Discrete
General
Metalog
Close

V2020 Release

With each new release, Analytic Solver gives you more! Our latest enhancements apply to both Analytic Solver Desktop and Analytic Solver Cloud.

In V2020, the LP/Quadratic Solver – probably the most-used Solver Engine in Analytic Solver – features significantly improved performance on linear mixed-integer models. Prior versions of this Solver would use only one processor core at a time, but V2020 will use *multiple processor cores* to speed your solution.

The plug-in large-scale Solver Engines in Analytic Solver V2020 also feature significantly improved performance (they continue to utilize multiple processor cores). These include the **Gurobi** Solver V9.0, with a new ability to solve non-convex quadratic models; the **Xpress** Solver V35, with a new Solution Refiner, and the Artelys **Knitro** Solver V12.1, with SOCP and MIP speedups.

Analytic Solver V2020 includes 12 new probability distribution functions, *enhanced* property functions for the PSI Distribution functions, *new* property functions for the PSI Statistics functions, and *new* "theoretical" functions that return analytic moments of distributions. Full details are in the *Frontline Solvers Reference Guide*, but here's a partial list of new/enhanced functions:

PsiBurr
PsiDagum
PsiDblTriang
PsiFatigue
PsiFdist
PsiFrechet
PsiHypSecant
PsiJohnsonSB
PsiJohnsonUB
PsiKumaraswamy
PsiReciprocal
PsiLevy
PsiLevyAlt
PsiHypSecantAlt
PsiCumulD
PsiLaplace
PsiCauchy
PsiTruncate
PsiCensor
PsiLock
PsiOutput
PsiTheoMean
PsiTheoMedian
PsiTheoMode
PsiTheoMin
PsiTheoMax
PsiTheoVariance
PsiTheoStdDev
PsiTheoSkewness
PsiTheoKurtosis
PsiTheoRange
PsiTheoPercentile
PsiTheoPercentileD
PsiTheoTarget
PsiTheoTargetD
PsiCategory

These functions make it **even easier** to adapt risk analysis models developed with other popular Excel add-ins to **work with Analytic Solver**. A simple Find and Replace of the function name prefix with 'Psi' is often all you need. And unlike those other Excel add-ins, with Analytic Solver you can easily run your model in the cloud with **Excel for the Web**, translate your model to RASON, and use it in **Power BI**, **Tableau**, or your own **web or mobile** application!

What's New in Analytic Solver V2020.5

Analytic Solver V2020.5 includes significant enhancements to both Monte Carlo simulation and optimization – but the most exciting new feature is a greatly expanded **Create App** facility that makes it easy to **deploy** your Excel analytic model as a **cloud service** (thanks to RASON), usable from nearly any corporate, web or mobile application. What's more, you can manage, monitor and update your own cloud services, without ever leaving Excel!

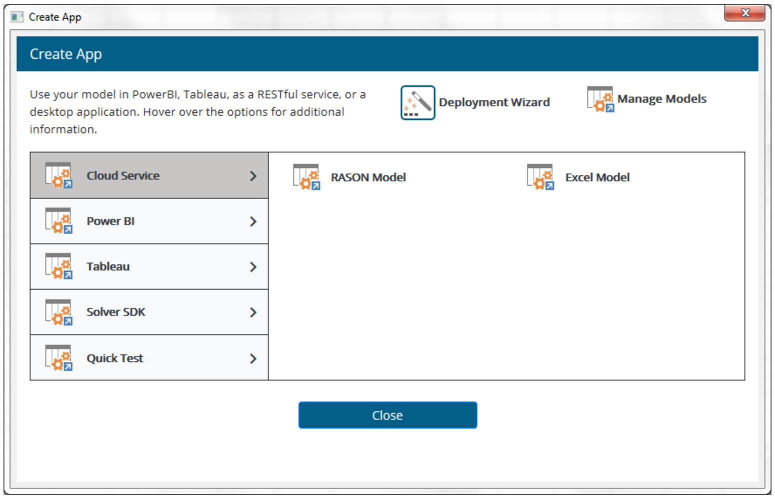
Easily Deploy Your Model as a Cloud Service

We've realized for many years that developing and testing your analytic model in Excel is often just the first step: To gain the **real business value** from the decisions that can be made with your model, it's often necessary to get the model into the hands of **other people in the business** – in a form where they can easily ensure that it has **up-to-date data, re-run** the model's optimization, simulation, or data mining process, and either **view** the results, or **plug them int**o another software application or process.

In our 2017 and 2018 Analytic Solver releases, we took the steps that were possible at that time, enabling users to get their models into the hands of **Power**

BI and **Tableau** users. And we built a facility to ***translate*** simpler Excel models into the RASON modeling language, enabling them to be solved in our cloud platform (RASON is an acronym for RESTful Analytic Solver Object Notation). But up to this point, a typical Excel user would still need help from a web developer, or would need learn JavaScript and other web development skills, to make truly effective use of this facility.

Now in our V2020.5 release of both Analytic Solver and RASON, we've gone much further to **simplify** the process of **deploying** an Excel model as a cloud service, and **connecting** it to databases and cloud data sources. The RASON cloud service will now accept and run **Excel workbook** models "on a par" with models written in the RASON modeling language. With the **Create App** menu option, you can turn your Excel model into a cloud service in seconds.



Use your model in PowerBI, Tableau, as a RESTful service, or a desktop application. Hover over the options for additional information.

As an Analytic Solver user, you can now create and test models, deploy them "to the cloud" – **point and click** – as full-fledged RESTful decision services, and even get reports of recent runs of your decision services, all without leaving Excel. Using our web portal at https://rason.com, you can go further –even embed your Excel workbook in a multi-stage "decision flow" that can combine SQL, RASON, Excel, and DMN models, passing results from stage to stage.

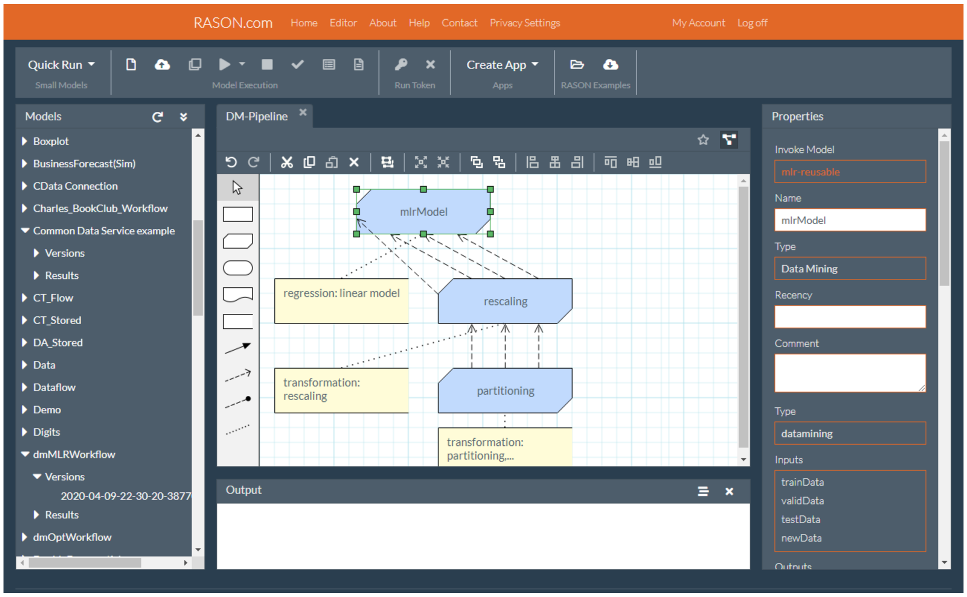
We start you out with a **RASON trial license**, so you can try out these new capabilities without purchasing anything else! (Of course, you may need an upgraded RASON license to deploy your model to many users, and re-solve it hundreds or thousands of times on our cloud servers.)

More About RASON Decision Services

RASON is an Azure-hosted cloud service that enables your company to easily embed 'intelligent decisions' in a custom application, manual or automated business process, applying the full range of analytics methods – from simple calculations and business rules to data mining and machine learning, simulation and risk analysis, and conventional and risk-based optimization.

RASON Decision Services can be used from nearly **any application**, via a series of simple REST API requests to https://rason.net. To express the full range of analytic models, RASON includes a high-level, declarative modeling language, syntactically embedded in JSON (JavaScript Object Notation), the popular structured format almost universally used in web and mobile apps. RASON results appear in JSON, or as more structured OData JSON endpoints.

RASON Decision Services also includes comprehensive data access support for **Excel, SQL Server on Azure, Power BI, Power Apps, Power Automate** (aka Microsoft Flow) and **Dynamics 365**. And it includes powerful model management tools, such as tracking model versions including "champions and challengers", monitoring model results, and automated scheduling of runs for both models and multi-stage decision flows.



How You Can Use RASON

You can use RASON to quickly and easily create and solve optimization, simulation/risk analysis, data mining, decision table, and decision flow models – instantly deployed as cloud services. You can learn RASON, create models, supply data and solve them, and even manage model versions and cloud data connections, "point and click" using https://rason.com, our "web portal" to the underlying REST API service.

If you've used another **modeling language** to build an analytic model, you'll find the RASON language to be simple but powerful and expressive – and integrating RASON models into a larger application, especially a web or mobile app, is **much easier** than with other modeling languages. Excel users will find that RASON includes virtually the entire **Excel formula language** as a subset. If you've used tools based on the DMN (Decision Model and Notation) standard, you'll find that RASON – and Analytic Solver, as shown in the chapter "Building Decision Tables" – fully support DMN and FEEL Level 2.

Unlike existing "heavyweight" Business Rule Management Systems, with year-long implementation schedules, six-figure budgets and limited analytics power, RASON Decision Services enables you to **get results in just weeks to months**, from building and testing models, to deploying them across an organization. With RASON, you can build successful POCs (Proofs of Concept) without any IT or professional developer support – yet RASON is very "IT and developer friendly" when you're ready to deploy your POC across your company.

New Time Series Simulation Functions

Analytic Solver V2020.5 includes another new set of PSI Distribution functions and related PSI property functions, focused around **time series simulation**. Earlier Analytic Solver versions supported time series simulation using functions such as PsiForecast() and PsiPredict(), and models fitted via Analytic Solver Data Mining – but V2020.5 goes further, to support time series functions found in other popular Excel add-ins, such as Palisade's @RISK. Full details are in the *Frontline Solvers Reference Guide*, but here's a partial list of new/enhanced functions:

PsiAR1
PsiAR2
PsiMA1
PsiMA2
PsiARMA11
PsiBMMR
PsiGBMJD
PsiARCH1
PsiGARCH11
PsiEGARCH11
PsiAPARCH11
PsiTSTransform
PsiTSIntegrate
PsiTSSeasonality
PsiTSSync

With these functions, virtually any risk analysis model developed with other popular Excel add-ins, such as Palisade's @RISK, can be easily made to work with Analytic Solver. An appendix in the *Frontline Solvers Reference Guide*, "@Risk to Analytic Solver Psi Function Conversion Table", explains the details. And with Analytic Solver, you can easily **deploy** your risk analysis model as a **cloud service** – usable from Tableau, Power BI, Power Apps, Power Automate, or virtually any corporate, web or mobile application!

New Optimization Result Functions

In every version of Analytic Solver (and its precedessors, such as Premium Solver and the Excel Solver), you could obtain all the properties of an optimal solution – such as initial and final values, dual values, and ranges for decision variables and constraints – via the Answer Report and Sensitivity Report, which are inserted into your Excel workbook as new worksheets. But what if you want only ***select subsets*** of these values – and you'd like to have them on the **same workshee**t as your model? That's now possible in Analytic Solver V2020.5. Just type these new functions into cells, or use the Function Wizard in Excel.

PsiInitialValue
PsiFinalValue
PsiSlackValue
PsiDualValue
PsiDualLower
PsiDualUpper
PsiCalcValue
PsiOptStatus
PsiModelDesc

These new functions have another purpose in V2020.5, when you use Analytic Solver's enhanced **Create App** facility to deploy your model as a cloud service: You can use them to determine a select subset of values from the solution that you want to ***return from your cloud service*** to a calling application.

What's New in Analytic Solver V2021

Analytic Solver V2021, our latest release, features faster startup and better support for 4K monitors in Analytic Solver Desktop, improved support for decision trees "on the spreadsheet" in Analytic Solver Cloud, and a new, faster version of the Xpress Solver (V37.1.3) – but the most exciting new features in this release give you new ways to define your own **custom functions**, in a way that works in both Excel Desktop and Excel for the Web, as well as our cloud platform RASON.

Lambda, Let and Box Functions

In years past, Excel-savvy analysts used VBA (Visual Basic for Applications) to define their own custom functions. While this still works in Excel Desktop using COM (28-year-old Component Object Model), VBA functions are *not* supported in Excel for the Web – and according to Microsoft, VBA and COM will *never* move to the cloud. If you want your custom functions to work in *both* desktop and cloud, your options have been limited – until now:

- Microsoft has introduced new Excel functions LAMBDA and LET. These are very special because you can use them in Excel formulas to define your own custom functions. The Excel community has expressed much excitement over these new functions, since they effectively make Excel a "complete programming language". (In Q1 2021, these functions are being rolled out across the different Office update channels.)
- On another front, there's the open standard known as DMN (Decision Model and Notation) – a business user-friendly "formula language" used to define business rules and decision tables, supported in "decision management" platforms from various vendors, and in Analytic Solver and RASON since 2019. DMN – now in version 1.3 – offers a way to define your own custom functions, known as "Box functions".

Analytic Solver V2021 includes support for *both* Excel's LAMBDA and LET functions, and for DMN-compatible Box functions. You'll find a new chapter in this User Guide, "Using Custom Functions", that explains how to use both approaches. Even better, LAMBDA and LET, and DMN Box functions enjoy full support from our PSI Interpreter – which means that our full range of Solver Engines, and our high-speed Monte Carlo simulation engine "understand" and take full advantage of custom functions that you define this way. This can yield better results than you've ever had with VBA-based functions that are embedded in an optimization or simulation model.

What's New in Analytic Solver V2021.5

Automate Data Mining with Find Best Model

Analytic Solver, our SDKs and RASON include comprehensive, powerful support for data mining and machine learning. Using these tools, you can "train" or fit your data to a wide range of statistical and machine learning models: Classification and regression trees, neural networks, linear and logistic regression, discriminant analysis, naïve Bayes, k-nearest neighbors and more. But the task of choosing and comparing these models, and selecting parameters for each one was up to you.

With the new Find Best Model options in V2021.5, you can automate this work as well! Find Best Model uses methods similar to those in (expensive high-end) tools like DataRobot and RapidMiner, to automatically choose types of ML models and their parameters, validate and compare them according to criteria that you choose, and deliver the model that best fits your data.

Better Simulation Models with Metalog Distributions and Fitting

Analytic Solver, our SDKs and RASON support over 60 "classical" probability distributions for Monte Carlo simulation. Since mid-2017, they've also supported the increasingly popular Metalog family of distributions, created by

Dr. Tom Keelin, and recently popularized by the nonprofit Probability Management group. Metalog distributions can closely approximate virtually any classical continuous distribution, and often they can better fit user data than classical distributions. In V2021.5 we've brought Metalog distributions to the fore, with a powerful new facility to automatically fit user data to the full range of possible (bounded and unbounded, multi-term) Metalog distributions. It's never been easier to get an accurate probability distribution that fits a real-world phenomenon.

Share Data Mining and Probability Models via RASON

In recent years, our Azure-hosted RASON Decision Services platform has offered increasingly powerful facilities to deploy models to the cloud and share them with other users – culminating in Frontline Solvers V2020.5, when we enabled deployment, sharing, versioning and management of Excel models as well as native RASON models, plus support for multi-stage decision flows, encompassing and going beyond traditional data science workflows.

In V2021.5 we've gone further: You can now deploy and share data mining and machine learning models, trained in Analytic Solver or RASON, to the Azure cloud, and use them directly for classification and prediction (without needing auxiliary "code" in R or Python, RASON or Excel). You can also deploy and share probability models, following the open Probability Management 3.0 standard. Using these Shared Information Probability resources (SIPs, also known as "Stochastic Information Packets"), you can ensure that your group or organization uses consistent data about uncertain/risky variables across simulation or decision models, enabling model results to be meaningfully compared.

And of Course, Optimization Enhancements

Analytic Solver, our SDKs and RASON have always offered rich support for conventional and stochastic optimization, improved in every new release. V2021.5 is no exception: We've made PSI Interpreter enhancements to better utilize main memory in large optimization models (with 1 million or more decision variables) in all three product lines. In our V2021.5 release, we're also shipping the latest Gurobi Solver 9.1, Xpress Solver 37.1.3.0, and new KNITRO Solver V12.4 with each of our products.

Analytic Solver Upgrade

Analytic Solver is our comprehensive tool for predictive, prescriptive and decision analytics. In recent years we've added deep support for business rules and decision tables, following the DMN standard, to Analytic Solver, and similar facilities in RASON. Since we've matched some entire products offered by competitors in the "rules" or "decision management" world, we've brought together our decision trees, DMN business rules and decision tables, and DMN Box functions, and combined them with Premium Solver Upgrade (for optimization models) to create a new product, Analytic Solver Upgrade.

In Excel, you'll see a new icon group on the Analytic Solver Ribbon Decision Model, next to the Optimization Model and Simulation Model groups. You'll also see the Tools group options moved to the Task Pane (already true in Analytic Solver Cloud version). Analytic Solver Upgrade will still be included in Analytic Solver Comprehensive and in a full RASON license, but its DMN-related and optimization features will have size limits in Analytic Solver Simulation and Analytic Solver Data Mining.

Analytic Solver Academy – Replaces Analytic Solver Basic

We've found that many purchasers of Analytic Solver Basic, who lacked a previous analytics background and didn't also purchase learning aids such as our Solver.Academy courses, experienced limited success and often didn't renew their licenses. In V2021.5 we are replacing Analytic Solver Basic with a bundle of all four current Solver.Academy courses plus a license for Analytic Solver with basic limits, if no other Solver license has been purchased.

What's New in Analytic Solver V2022

Faster Interaction, Faster Solves

In Analytic Solver V2022, our latest release, Excel startup is **faster**, and your 'regular interaction' with Excel is **faster** when Analytic Solver Desktop is loaded. Dialog rendering is improved for users with very-high-resolution monitors, as well as monitors with limited vertical depth.

Most impactful, the process of interpreting the model – when the model type is diagnosed and when "Setting Up Problem…" appears in the Task Pane status bar – is **faster** for most models, and *significantly* **faster** for larger models, especially on models with very deep "chains" of formulas that depend on other formulas. We're also including new, **faster** versions of the Gurobi Solver for linear mixed-integer models, and the Knitro Solver for nonlinear models.

V2022 re-introduces the **Freeze** and **Thaw** options (now located on the Task Pane Tools tab), which allow you to share Analytic Solver models containing PSI function calls with users who don't have Analytic Solver installed. ("Freeze" will save PSI function call formulas in cell comments, and "Thaw" will restore them later as formulas.)

Deploy Model functionality in V2022 is enhanced on our RASON cloud server, and **Microsoft Teams** messaging is improved for several account types.

In case you didn't know: Recent Analytic Solver releases, including V2022, allow you to **Test Run** models that *exceed* the limits of your current license. (You don't get full results for all variables and constraints, but you do get the final objective value and solution time.) You can even use optional large-scale Solver Engines, like the Gurobi and Knitro Solvers, in a "Test Run". To make it clear what your current license does and doesn't include, in V2022 you'll see "(Test Run)" next to the names of optional Solver Engines in the Task Pane.

What's New in Analytic Solver V2023

Automated Risk Analysis of Machine Learning Models

Besides our "usual enhancements" such as faster Solver Engines and new PSI functions like PsiCalcParam(), Analytic Solver V2023, our latest release, features an innovative (and patent pending) new capability for **automated risk analysis** of **machine learning** models on the Data Mining tab.

A further benefit of this new feature is a general-purpose, easy to use facility for **synthetic data generation**, to augment the data you already have. Analytic

Solver Data Mining and Comprehensive users are able to use these new features with size limits constrained only by memory, but *all* Analytic Solver users have access to these features with “Basic size limits” for your datasets.

Until Analytic Solver V2023, data science and machine learning (DSML) tools – including ours – had no facility for **risk analysis** of machine learning (ML) models, prior to their production use. Most tools (including ours) had facilities for ‘training’ the model on one set of data, ‘validating’ its performance on another set of data, and ‘testing’ it versus other ML models on a third set of data. But this is not *risk analysis*.

Synthetic Data Generation ‘For Free’

Synthetic data generation has come into use in recent years to augment available datasets, when the available data is limited, or is restricted by law or regulation, such as with personal health information (PHI). In Analytic Solver V2023, you have a powerful, general-purpose tool for synthetic data generation, using the new **Generate Data** button on the Data Mining Ribbon tab. So far, we’re keeping Analytic Solver current with the “state of the art”.

But with the new **Simulation** tab that you’ll find in every ML model training dialog in Analytic Solver, we’ve gone *beyond* the “state of the art”, to bring you a new way to assess your trained ML model’s performance – not to determine how well it **has performed** on data you have, but to quantify how it **may perform differently** on data it will encounter in the future. We generate synthetic data “on the fly” and use it in a Monte Carlo simulation of your ML model’s performance – and we highlight **differences** in model performance in training versus simulated production use.

The beauty of this approach is that **you don’t have to do any work** to obtain a risk analysis of your model’s performance, beyond a few mouse clicks on the **Simulation** tab to enable the analysis, which is entirely **automated**. You don’t even have to be familiar with the features of Analytic Solver Simulation to use this capability for machine learning!

(You *can* adjust several settings that involve distribution fitting, correlation and copulas, Monte Carlo sample generation, etc. – but the default settings work very well.) Typically, the analysis adds only **seconds** to a perhaps a **minute** to the time taken to train and validate your ML model. So you can make this a routine **part of the process** of training and assessing new ML models.

As noted earlier (opposite the title page) in this User Guide, we have a patent application pending titled “Automated Risk Analysis of Machine Learning Models”. But as an Analytic Solver (or Solver SDK or RASON) user, you gain first access to this new capability at no extra cost.

What’s New in Analytic Solver V2023 Q1

We released Analytic Solver V2023 in September 2022, with the innovative enhancements described above – but we surprised even ourselves by how much more we could offer our customers by December 2022! So we’ve christened this new release “Analytic Solver V2023 Q1”.

Faster LP/Quadratic Solver and Large-Scale LP/QP Solver Engine

Our “headline feature” for this release is a new, higher performance version of the built-in LP/Quadratic Solver, as well as its “cousin” the Large-Scale LP/QP

Solver Engine which has much higher problem size limits. We expect most users will see **faster – sometimes *much* faster** solutions with this new version, both for LP and QP (linear programming and quadratic programming) models, and for LP and QP models with integer constraints.

And there's more: Now you can solve models with (convex) **quadratic constraints**, with or without a quadratic objective, using this Solver Engine.

(There are always tradeoffs: We've found that for about 10% of models we've tested, the new version is not faster, and can even be *slower* for models with integer constraints. If you encounter this, just select the **Classic Search** option to get exactly the same performance you were getting before.)

More Plug-in Solver Engine Improvements

V2023 Q1 also includes new versions of other large-scale Solver Engines. The **Gurobi Solver Engine** is upgraded (to their V10.0) with a range of enhancements, yielding solution times faster by 3-10% to 25% on a wide range of linear and quadratic mixed-integer models. The **Xpress Solver Engine** is upgraded (to their V9.0) with enhancements to strong branching, separation of cutting planes, and a new heuristic method, run at the branch & bound root. The **KNITRO Solver Engin**e is upgraded (to their V13.2) with a range of enhancements for smooth nonlinear mixed-integer models, including new presolve, cut selection and heuristic methods.

Greatly Improved "Deploy Your Model to Teams" Capability

We learned from surveys that a large majority of our customers work in companies using **Microsoft Teams –** so we've significantly enhanced a feature introduced in Analytic Solver V2022, that makes it easy to **share** your Excel model results with colleagues in your company, using Teams. The V2022 feature was designed to share your (entire) model, but we realize that many users want or need to share just the **model results** – not the full model with all its optimization and/or simulation model elements.

In V2023 Q1, when you choose **Deploy Model** from the Ribbon and click **Teams – Teams Report**, Analytic Solver will automatically create a new workbook holding only **model results**, with external links to your **model** workbook. You can choose exactly which optimization and/or simulation results you want to include in this workbook. The new workbook will be saved online, and made available to the users you want, through a "Teams channel" that you select. Your colleagues, using **just Teams**, will be able to open the workbook and view, copy or work with the results you're providing. And perhaps the best part: When you **re-run** your optimization or simulation model with new data, the workbook in Teams will be **automatically updated** (via those external links) with the latest model results! See the chapter "Deploying Your Model" in this Guide for full details.

Risk Analysis of Machine Learning Models Created in Other Software

Analytic Solver V2023 introduced an innovative (and patent pending) new capability for **automated risk analysis** of **machine learning** models – see "What's New in Analytic Solver V2023 Q1" above for a complete summary, and see our Data Mining User Guide for full details. But we realize that many people create and test machine learning models using other software. Those folks just don't have the ability to quantify how their ML models **may perform differently** on data they will encounter in the future … until now. But **you can help them**, using Analytic Solver V202 Q1 (or using RASON V2023 Q1, if they prefer to use a cloud platform).

In V2023 Q1, we've made it easy to perform **automated risk analysis** of models **created in other software** and saved in **PMML** (Predictive Modeling Markup Language) format. PMML is an open standard that is widely supported by software for machine learning. Analytic Solver and RASON will also save a trained machine learning model in PMML form. But now you can bring a PMML model into Analytic Solver, plus some of the data you used to train the model (just copy the PMML text and the data onto worksheets) – then use simple menu options to quickly get insights into the future performance of this ML model, from our **automated risk analysis** methods.

There are plenty of other small enhancements and fixes in V2023 Q1 – and we have more new features "in the works". But you can see why we felt we were ready to deliver another major version to our customers – in time for Christmas!

What's New in Analytic Solver V2023 Q3

In Analytic Solver V2023 Q3, we've made it easier than ever to create analytic models, and easier to deploy them for ongoing use, through two major new features: our new conversational "AI Agent", with a button next to "Help" on the Analytic Solver Ribbon, and a new "Identify Inputs" feature that simplifies the task of updating key input parameters when it's time to re-run your model.

AI Agent: Ask for Help from ChatGPT "Trained on Analytic Solver"

By now, nearly everyone who's been following developments in software is aware of ChatGPT, the conversational agent using "Generative AI" methods, developed by the OpenAI nonprofit closely affiliated with Microsoft. You can ask ChatGPT questions on almost any topic – including analytic methods – and get meaningful and interesting (though not always 100% correct!) answers.

That's great – but what if you could have a ChatGPT "technical support agent" that had studied all 2,300 pages of Frontline's User Guides (including this one), Reference Guides and QuickStart Guides – everything about optimization, Monte Carlo simulation, data science and machine learning, and business rules – and was instructed to use its knowledge to answer your questions? Well, that's what you have in Analytic Solver V2023 Q3!

Actually making this work involves a fair amount of software engineering: Generative AI tools need the right "context" to respond to your question – so we've built an online resource where all those 2,300 pages of Guides are represented via "vector embeddings" that enable searches by meaning, not just "keyword matches". Our AI Agent automatically searches this online resource to create "context' for your queries to ChatGPT – and you can also search this online resource directly.

We're using the "real" online ChatGPT 3.5 Turbo version for its full conversational capabilities, which costs us money for every query – so you will find some limits placed on query length and number of queries per month, depending on your software license type. But we expect you'll be able to use our AI Agent to amplify your own efforts, ultimately building more capable and effective analytic models. We look forward to your feedback!

Identify Inputs: Easily Set Up your Model for Data Updates and New Solves

When you're first building and solving a model in Analytic Solver, usually your focus is on getting a solution for a specific instance of a business problem, with data you have today – gathered from one or several external sources into Excel. After some effort, you're now getting solutions from optimization, simulation, or data science and machine learning. That's a success by itself … but it leads to a desire to "do it again (and again) in the future."

But … those data and parameter values that *should be updated* for a future run may be scattered around your Excel spreadsheet, which also includes cells with calculated formulas, cells with constant data that doesn't need updating, and cells with data that you want to be visible, but that doesn't actually affect or "participate in" the analytic model. How do you find *just* the parameter values that *should* be updated – and then, how do you *actually update* them?

Now you can choose **Tools – Identify Inputs** and get help doing exactly that. This tool uses our PSI Interpreter to scan your model formulas, identify and list only those cell ranges that are candidates for data updates that will affect the model when solved. You can then pick and choose from a list of cell ranges, which ones you actually need to update for a new run.

Then you can either (i) let Analytic Solver "highlight" those cells with colors and backgrounds, or even better, (ii) automatically add calls to our PsiInput() function that reference those cells. These cells will automatically appear in the Task Pane as part of your model (under the **Input Data** heading) – but when you deploy your model for use *beyond Excel*, via our RASON and Solver SDK tools, you'll have easy ways to supply new values for exactly those input cells.

If you've planned in advance for data updates and re-solves, Tools – Identify Inputs can help you check and validate your work. But if – like most of us – you were focused on getting a first-time solution and 'put off' the task of data updates and re-solves, Tools – Identify Inputs can be a huge time-saver!

What's New in Analytic Solver V2024 Q2

In **Analytic Solver V2024 Q2**, our overall theme is "Core Performance Improvements" -- indeed we think every user with a model of nontrivial size and complexity should **upgrade** (at no extra cost), since you're likely to see performance improvements -- especially faster **"Setting Up Problem..."**.

Solving Outside Excel

The most visible change -- though *not* the one impacting the most users -- is on the menu for the **Optimize** button on the Ribbon: the last dropdown choice, **Solve on Solver Server**, has been greatly extended. In past Analytic Solver releases, this choice allowed you to solve your model (actually perform the optimization) **outside** the running Excel program, in a separate program called Solver Server (part of our Solver SDK product), that can run on your PC or on another PC on your local area network (LAN). But the Excel-based Analytic Solver add-in continued to "monitor" the solving process, which meant that you couldn't do *something else* with Excel, while waiting for a "long solve" to complete.

In Analytic Solver V2024 Q2, this choice allows you to solve your model (again, actually perform the optimization) **outside** of Excel, on any of three alternatives: the SDK-based **Solver Server**, our public, Azure-based **RASON server**, or on a new offering: our **Containerized RASON Server --** which brings all the capabilities of RASON to your own PC, another physical or virtual PC on your LAN, or your own cloud account (we'll say more about this in an upcoming blog post). And unlike in previous releases, V2024 Q2 is able to "monitor" the solving process **without** blocking your use of Excel for *something else*, even working on a different workbook. When the final solution for your long-running model is available, you'll be notified in Excel, and your workbook containing the original model will be automatically updated, without disturbing anything else.

New Versions of Gurobi and OptQuest Solvers

Analytic Solver V2024 Q2 also includes new (minor) releases of the **Gurobi** Solver and **OptQuest** Solver (again at no extra cost), that improve performance on a range of models using these plug-in Solver Engines. The Gurobi Solver (their version 11.0.2) has typical speed improvements of 8% to 18% compared to pre-V11 versions. The OptQuest Solver (their version 9.1.2.9) includes an improved "diversity search" algorithm, among other enhancements.

PSI Interpreter Improvements

In Analytic Solver V2024 Q2, we expect the biggest impact on performance for most users will come from some **deep** improvements in the **PSI Interpreter**, our "Polymorphic Spreadsheet Interpreter". This is the part of Analytic Solver (and RASON) that takes your model as expressed in Excel or RASON formulas, analyzes the model, and converts it into a form **usable by** the Solver Engines -- whether it's linear or quadratic (for the LSLP or Gurobi Solvers), nonlinear (for the LSGRG, LSSQP and KNITRO Solvers), or "non-smooth / arbitrary" (for the Evolutionary and OptQuest Solvers). The PSI Interpreter is hard at work when you see "Setting Up Problem..." in the Task Pane, but also **during the solution** of nonlinear and non-smooth models.

In Analytic Solver V2024 Q2, the PSI Interpreter has been re-engineered to use "sparse methods in automatic differentation" (take a look at the Wikipedia article we've linked, if you're curious about the methods from algebra and calculus). The practical impact for users with large models is **faster** end-to-end solution times for linear models, less time spent in **"Setting Up Problem"**, and big savings in **memory use** (which translates to faster solutions, or solutions for models that exhausted memory in the past).

All you really need to know is "much better performance", but if you're curious, the rest of this post will seek to explain just some of what the PSI Interpreter does for you.

More About the PSI Interpreter: The Jacobian

You might be surprised to learn that the Solver Engines, or optimization algorithms, don't work with your Excel formulas at all. They require as input *tables of numbers* (constant or changing) that describe your model, at a rather low level. The PSI Interpreter reads, parses, and interprets your formulas and produces those tables of numbers.

The most fundamental such table is called the **Jacobian matrix** -- see the Wikipedia article (rather technical) or "A Gentle Introduction to the Jacobian" which comes from machine learning, where the same matrix arises. In optimization models, this matrix has a row for the objective and each constraint, and a column for each decision variable -- so its size grows with variables *times* constraints. Each matrix element is the **partial derivative** (or *rate of change*) of one constraint (or the objective) with respect to one decision variable.

In linear models, the partial derivatives are all **constant** numbers and are referred to as "LP coefficients" -- but these numbers may not appear explicitly anywhere in your Excel model. They are computed by the PSI Interpreter. In nonlinear and non-smooth models, the partial derivatives are **not constant** -- their values depend on the current values of (potentially many) decision variables. So the PSI Interpreter must **re-compute** them, and re-supply them to the Solver Engine, each time the Solver Engine tests a new set of values for the decision variables -- and this can happen thousands or even millions of times while a model is being solved. So the PSI Interpreter has a lot of work to do.

Let's say your model has 32,000 variables and 32,000 constraints (the upper limits of our *Standard* Large-Scale LP/QP Solver -- the *Extended* version removes the limits). That's not very large by our customers' standards -- but it means the Jacobian matrix has 32,000 x 32,000 = 1,024,000,000 (just over **one billion**) elements. It takes a *lot* of memory, and a *lot* of computing to calculate a billion different values from your Excel model! The saving grace is that in *most* models, *most* of these matrix elements will be **zero**: A typical constraint will depend on a **small subset** of the decision variables, so its rate of change is zero with respect to all the other decision variables. The challenge for the PSI Interpreter is to **figure out which** elements will be zero (the "sparsity pattern"), **without** consuming a lot of memory and time just doing this. In V024 Q2, the PSI Interpreter is far better at this!

Beyond the Jacobian: The Hessian

When your model goes "beyond linear", even with just a quadratic objective (as in portfolio optimization), the Solver Engines will typically require a further large table of numbers, called the **Hessian matrix** (again see the Wikipedia article if you're interested). The Hessian has as many rows and as many columns as the number of **decision variables** in your model; it is typically computed for the objective function (at least). Each matrix element is the **second partial derivative** of one function (usually the objective) with respect to **a pair** of decision variables. To compute this, the PSI Interpreter must do even more work, and it must deal with the same issues of memory and computing time, for perhaps a billion values. Again *some* of these elements will be **zero**, since the function value may not depend on *all possible pairs* of variable values -- the challenge is figuring out **which ones!** And again, in V2024 Q2 we've made major progress on this. We've seen some large customer models that formerly would always exhaust memory after running for hours, suddenly solve in a minute or less!

And There's (Much) More...

There's much more to the PSI Interpreter, that we don't have time or space to fully describe here. Whenever Microsoft adds **new built-in functions** or formula features (such as "spilled arrays") to Excel, we've been hard at work supporting those new features in the PSI Interpreter. Whenever you use menu options such as "Analyze Original Model" or "Analyze Transformed Model", notice your **Model Type** (LP for linear, QP for quadratic, etc.) or the counts of model **Dependencies** in the Task Pane, or create a Linearity Report, that's the PSI Interpreter at work. And the PSI Interpreter is also the key to **super-fast Monte Carlo simulation** in Analytic Solver, usually ten times faster (or more) than other Excel add-ins for simulation. The PSI Interpreter also parses and interprets **DMN** (Decision Model and Notation) for business rules and decision tables.

You might be surprised to learn how many of our competitors offering Excel add-ins (especially firms that have been acquired by other companies) have been "resting on their laurels" for a decade or more. At Frontline Solvers, ***we never rest!*** We spend millions each year on real R&D, to bring you the very best in modeling and analytic software tools.

What's New in Analytic Solver V2024 Q3

In **Analytic Solver V2024 Q3**, we've gone beyond our conversational "AI Agent", introduced one year earlier in Analytic Solver V2023 Q3: Our new "AI Assistant" doesn't just consult our User Guides and publicly available info – it seeks to "understand" the *model you are working on*, so it can assist you in improving your model, or fixing problems with it!

But we don't just focus on the latest AI methods – we also improve the usability and performance of the optimization and simulation solvers you use every day.

New AI Assistant

The most visible change in V2024 Q3 is on the Analytic Solver Ribbon, which now sports two "AI icons": On the far right next to the **Help** icon is **AI Ask a Question**, similar to the capability we introduced in Analytic Solver V2023 Q3 (but enhanced): Ask it a question, and it will answer conversationally, with links to our example models, User Guides and Reference Guides. On the far left next to the **Model** icon is a new **AI Assist** icon: When you click this icon, the AI Assistant will examine and use "as context" your currently-open model in Excel (this also works in our cloud platform RASON): You can then ask a question and engage in a dialog with the Assistant about *your model*. For example, you can ask "is my model missing an essential element?" or "my model is building products from parts in inventory, how can I add another part?"

Optimization Improvements

In this release, we've included the latest versions of the **Gurobi Solver** (their 11.0.3), the **XPRESS Solver** (their 43.1.3.0), and the **MOSEK Solver** (their 10.2.0.3). For all the Solver Engines, we've expanded "Verbose" logging (enabled via the Task Pane Platform tab **General – Log Level** option) to include all the solution progress information that each Solver can provide. And in the case of the Gurobi Solver working on a non-integer, non-convex QCP

(Quadratically Constrained Problem), Verbose logging now provides "branch level" solution progress info that just wasn't available before.

Simulation Improvements

In Monte Carlo simulation, we've added new "alternative parameter" distribution functions named PsiFatigueLifeAlt() and PsiFrechetAlt(). The first of these, the "fatigue life" or Birnbaum–Saunders distribution, is extensively used in reliability problems to model failure times. The second, the Frechet or inverse Weibull distribution, is used to model extreme events, for example in weather forecasting and hydroelectric power application.

What's New in Analytic Solver V2025 Q1

We've just released **Analytic Solver V2025 Q1**, with new versions of Analytic Solver Desktop and Cloud (for Excel users), Solver SDK (for developers), and our cloud platform RASON. This release features both **usability** improvements and a powerful new way to solve a **wide range** of **nonlinear** optimization models using the **Gurobi Solver** Engine -- traditionally known for solving linear (and quadratic) mixed-integer models. This new approach yields **globally optimal** solutions **every time** -- not possible with other nonlinear Solvers -- and often **much faster** solutions.

Usability: New Flexible Dialogs

Analytic Solver has always offered an easy user interface with an Office-style Ribbon, Task Pane, and dialogs that you can move around and resize. But the **content** of many dialogs used fixed-size fonts and graphics. To help with both **accessibility**, and better use of **larger, high-resolution screens** when you're "consuming" this content, in Analytic Solver V2025 Q1 we've revised the Task Pane and scores of other dialogs with **"zoomable"** content. You can now press the CTRL key, then use the mouse wheel to enlarge or shrink the text and graphics in almost any dialog or pane of Analytic Solver. At the bottom of this page is an example, where we've enlarged the text in the top and bottom parts of the Task Pane, but left the middle part (Model Diagnosis) unchanged – so you can see the difference.

Performance: New Nonlinear Solver Power

Analytic Solver V2025 Q1 features a *very special* level of support for the just-released (November 19) **Gurobi Solver 12.0**. This popular Solver again improves performance on linear programming (about +4%) and mixed-integer programming (about 13%) models, and even better performance (about +28%) on non-convex mixed-integer quadratic models. But most interesting for our Excel users is Gurobi's improved support for (often non-convex) **mixed-integer nonlinear** optimization problems. Such problems arise in specialty chemicals, hydroelectric power, and some aeronautics and space applications.

Unlike other nonlinear Solvers (such as our LSGRG, LSSQP and KNITRO Solvers), Gurobi takes a different approach, that can yield **globally optimally** solutions **IF** you can formulate your model with (only) certain algebraic expressions -- defined via special Python API calls -- that use just over a dozen mathematical operators (plus, minus, times, divide, square, square root, exponential and logarithm, etc.). If you're really good at Python coding and really good with algebraic manipulation, you can code this yourself. But in Analytic Solver V2025 Q1, **you don't have to do any extra work** to define your nonlinear model for the Gurobi Solver.

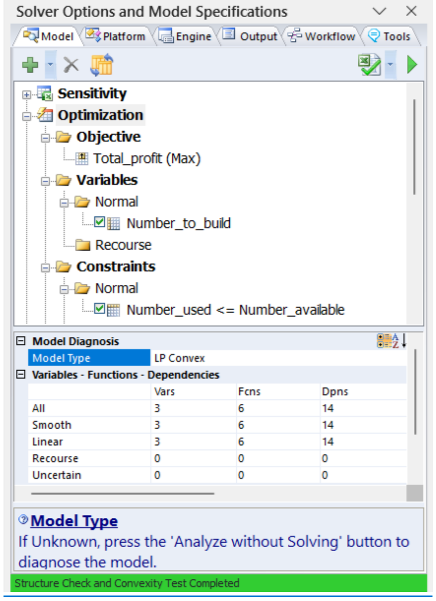
Standard Excel Formulas, New Solving Speed

You can write your model in Excel, using any of Excel's mathematical operators and any of Excel's 470+ built-in functions. Analytic Solver's **PSI Interpreter** will **analyze** your formulas (including formulas that depend on other formulas, to any level), **diagnose** your model as linear, quadratic, smooth nonlinear or non-smooth, and -- In V2025 Q1 -- tell you if your model is "nonlinear and **solvable via the new Gurobi Solver**". Analytic Solver can generate the internal "expression trees" that the Gurobi Solver requires.

Analytic Solver will even expand various Excel functions, based on their mathematical definitions, into chains of "Gurobi-supported primitive operators". For example, you can just go ahead and use the STDEV Excel function -- Analytic Solver will recognize this as "Gurobi-eligible" and create (internally) an expression tree using square, plus and square root operators, based on the definition of "standard deviation". The bottom line: With Analytic Solver, you can use the Gurobi 12.0 Solver to solve a much **wider range** of nonlinear models, much **more quickly and easily**, than you could any other way.

For "eligible" nonlinear optimization models, the Gurobi 12.0 Solver is often **faster** than other nonlinear Solvers, especially when **integer variables** are also used in the model, and when you require a **globally optimal** solution. This isn't always true however -- occasionally Gurobi is (much) slower -- so it's good that you have alternative nonlinear Solvers available to you, even in the basic Analytic Solver Optimization product. We do hope and expect to see even better performance in future versions of the Gurobi Solver.

Analytic Solver V2025 Q1 is a **free upgrade** for anyone with a paid Analytic Solver Optimization (or higher) license. If you've been a customer for years, you're probably accustomed to such dramatic enhancements -- but we're always working hard to bring you **more value, sooner** than anyone else in our "world" of advanced analytics.



Sensitivity
Optimization
Objective
Total_profit (Max)
Variables
Normal
☑ Number_to_build
Recourse
Constraints
Normal
☑ Number_used <= Number_available

Model Diagnosis

Model Type	LP Convex

Variables - Functions - Dependencies

	Vars	Fcns	Dpns
All	3	6	14
Smooth	3	6	14
Linear	3	6	14
Recourse	0	0	0
Uncertain	0	0	0

Model Type
If Unknown, press the 'Analyze without Solving' button to diagnose the model.

Structure Check and Convexity Test Completed

What's New in Analytic Solver V2025 Q2

We've just released **Analytic Solver V2025 Q2**, with new versions of Analytic Solver Desktop and Cloud (for Excel users), Solver SDK (for developers), and our cloud platform RASON. This release features both **speed** and **usability** improvements, a **smarter** AI Agent to assist with model building, and support for the new/improved **Metalog 2** family of probability distributions.

Faster When Using, Not Just Solving

Customers (and prospective customers) regularly bring us large models that **stretch limits** in different ways. We've long supported models with millions of decision variables and constraints. But for example, customers have brought us models where one constraint uses a formula that depends on several other cells -- where each of these depends on several other cells -- to **10,000 or more** levels -- and we've found ways to make our analysis of those "deep formula chains" faster. When we find a "math + computer science" solution for these "stretch" cases, all customers end up benefiting.

More recently, customers have brought us models with so many constraints, formulas or Excel defined names that it took a long time to display or update the **Task Pane**. Other customers have brought us models with large numbers of **quadratic constraints** (each requiring a "Hessian evaluation"), or with constraint **right hand sides** deeply dependent on decision variables. In our V2025 Q2 release, Analytic Solver is **much faster** in all these cases.

And for those solving with the **Gurobi** Solver Engine, our V2025 Q2 release includes the latest 12.0.2 release from our friends at Gurobi Optimization, with their latest fixes and speed improvements.

Enhanced Excel Support, Smarter Human Support

Analytic Solver V2025 Q2 includes improved support for Excel models using the new **LET function** (the LAMBDA function was already supported). We've also improved support for recalculation of optimization and simulation models that use certain advanced Excel functions, notably our Psi functions and our support for DMN-style **decision tables** and "**box functions**".

And our **AI Agent** in V2025 Q2 is even better at **helping you** enhance and even find and fix problems in your model: It now takes advantage of both the latest ChatGPT large language model with our RAG support, and "**model diagnostics**" from our PSI Interpreter to better understand the structure of your model, and how it might be fixed or improved.

New Support for Metalog2 Distributions

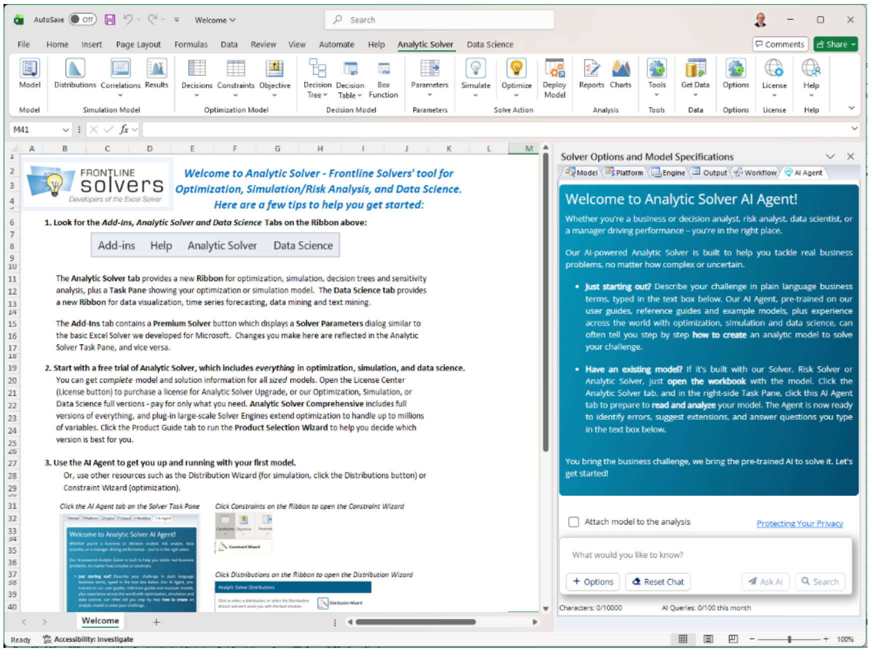
Way back in 2017, Frontline Systems was among the very first software vendors to implement support for the new (introduced in 2016) **Metalog** family of probability distributions -- see also the Metalog YouTube Channel. This family of distributions has some **remarkable properties**: virtually unlimited shape flexibility, flexible bounds, simple, closed-form quantile function (inverse CDF) equations, a simple, closed-form PDF (probability density function), and **Bayesian updating in closed form** in light of new data(!)

Over the last 8 years, the Metalog distributions have been **widely adopted**, in applications ranging from astrophysics and cosmology, crystallography, integrated circuit design and nanomaterials, to volcanic discharges in earth science, toxin levels in drinking water, oil-field production forecasting, electric vehicles and hydrogen production in renewable energy, cybersecurity risk mitigation, and conditional value at risk in quantitative finance.

Now in 2025, **Dr. Tom Keelin**, inventor of Metalog distributions, and two other authors are publishing a new paper *On the Properties of the Metalog Distributions* which refines and, in special cases, *redefines* the Metalog family of distributions (improving cases where the fitting process could yield infeasible results), earning the title "**Metalog 2**". And in Analytic Solver V2025 Q2, the refinements in this paper are **fully implemented** via our PsiMetalog2() function and our built-in automatic distribution fitting. As in so many cases in the past, Analytic Solver users are **first to gain access** to this new/improved capability -- at no extra cost for current licensees!

What's New in Analytic Solver V2025 Q4

We've just released **Analytic Solver V2025 Q4**, with new versions of Analytic Solver Desktop and Cloud (for Excel users), Solver SDK (for developers), and our cloud platform RASON. In this release, we've focused on **ease of use** and better support for **model creation**. Most prominent is the new **AI Agent tab** in the Task Pane, which appears (in lieu of the Model tab) when you visit Analytic Solver on the Ribbon.



Welcome to Analytic Solver - Frontline Solvers' tool for Optimization, Simulation/Risk Analysis, and Data Science.
Here are a few tips to help you get started:

1. Look for the *Add-Ins, Analytic Solver and Data Science* Tabs on the Ribbon above:

The **Analytic Solver tab** provides a new **Ribbon** for optimization, simulation, decision trees and sensitivity analysis, plus a **Task Pane** showing your optimization or simulation model. The **Data Science tab** provides a new **Ribbon** for data visualization, time series forecasting, data mining and text mining.

The **Add-Ins** tab contains a **Premium Solver** button which displays a **Solver Parameters** dialog similar to the basic Excel Solver we developed for Microsoft. Changes you make here are reflected in the Analytic Solver Task Pane, and vice versa.

2. Start with a free trial of Analytic Solver, which includes *everything* in optimization, simulation, and data science.

You can get *complete* model and solution information for all *sized* models. Open the License Center (License button) to purchase a license for Analytic Solver Upgrade, or our Optimization, Simulation, or Data Science full versions - pay for only what you need. **Analytic Solver Comprehensive** includes full versions of everything, and plug-in large-scale Solver Engines extend optimization to handle up to millions of variables. Click the Product Guide tab to run the **Product Selection Wizard** to help you decide which version is best for you.

3. Use the AI Agent to get you up and running with your first model.

Or, use other resources such as the Distribution Wizard (for simulation, click the Distributions button) or Constraint Wizard (optimization).

Welcome to Analytic Solver AI Agent!

Whether you're a business or decision analyst, risk analyst, data scientist, or a manager driving performance – you're in the right place.

Our AI-powered Analytic Solver is built to help you tackle real business problems, no matter how complex or uncertain.

- **Just starting out?** Describe your challenge in plain language business terms, typed in the text box below. Our AI Agent, pre-trained on our user guides, reference guides and example models, plus experience across the world with optimization, simulation and data science, can often tell you step by step **how to create** an analytic model to solve your challenge.
- **Have an existing model?** If it's built with our Solver, Risk Solver or Analytic Solver, just **open the workbook** with the model. Click the Analytic Solver tab, and in the right-side Task Pane, click this AI Agent tab to prepare to **read and analyze** your model. The Agent is now ready to identify errors, suggest extensions, and answer questions you type in the text box below.

You bring the business challenge, we bring the pre-trained AI to solve it. Let's get started!

The **AI Agent**'s ability to help you **build a model** in the first place is greatly improved in this release. You can *type in plain language* what *you want to accomplish*: For example "My company produces material in large sheets, that we need to cut into smaller sizes that our customers need. We'd like to do this while minimizing waste." The AI

Agent can recognize this as a “cutting stock optimization problem”, tell you how to formulate a model with the right variables and constraints, advise you what data you’ll need, and give you link(s) to example model(s) relevant to what you’re trying to accomplish.

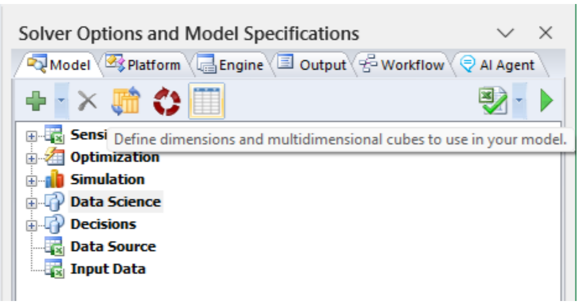
You can also use the **AI Agent** to **diagnose a model** that you’ve previously created – looking for missing elements, possible errors, and suggested enhancements. Just open your model workbook, then go to the AI Agent tab in the Task Pane.

Semantic Search Without Using AI

Some customers are uncomfortable with the idea that an external agent like ChatGPT-5 might “read and understand” their model. If that’s your situation, you can still use the **Search** button (instead of the **Ask AI** button) at the bottom of the **AI Agent tab**. This performs a semantic search (*much better* than just searching for keyword matches) across all 3,500 pages of Frontline’s User and Reference Guides, and our hundreds of example models. It’s also much faster than “Ask AI”.

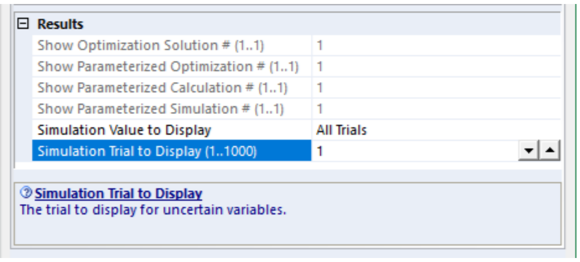
A Shorter Ribbon, and Other Simplifications

Note that, with the AI Agent “up front” in the Task Pane, we no longer need the **AI Assist** and **Ask Question** icons on the Ribbon. We’ve also moved the **Dimensional Modeling** icon from the Ribbon to the Task Pane Model tab, where we’ve also placed the **Reset All** icon (in red):


Solver Options and Model Specifications

Define dimensions and multidimensional cubes to use in your model.

And we’ve removed the **Tools tab** from the Task Pane: The controls allowing you to select (and display on the worksheet) specific Simulation Trials, or specific runs from multiple parameterized Simulations or Optimizations, are now – more conveniently – available at the bottom of the **Output tab**:


⊟ Results

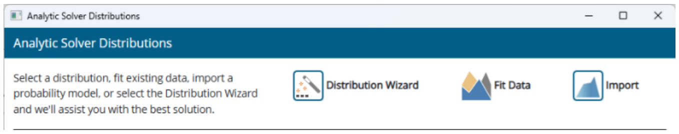
Show Optimization Solution # (1..1)	1
Show Parameterized Optimization # (1..1)	1
Show Parameterized Calculation # (1..1)	1
Show Parameterized Simulation # (1..1)	1
Simulation Value to Display	All Trials
Simulation Trial to Display (1..1000)	1

Simulation Trial to Display
The trial to display for uncertain variables.

Easier to Find Distribution Fitting

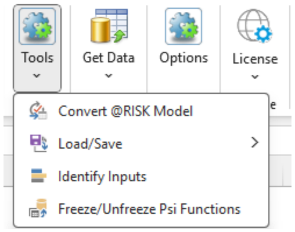
In this release, we’ve *swapped the positions* of two popular features, based on customer feedback about where they *expected* to find them:

If you're **defining probability distributions**, and you have sample data that you'd like to fit to a well-known type of distribution, you'll find a **Fit Data** icon at the top of the main Distributions dialog, displayed when you click the **Distributions** icon on the Ribbon:



Select a distribution, fit existing data, import a probability model, or select the Distribution Wizard and we'll assist you with the best solution.

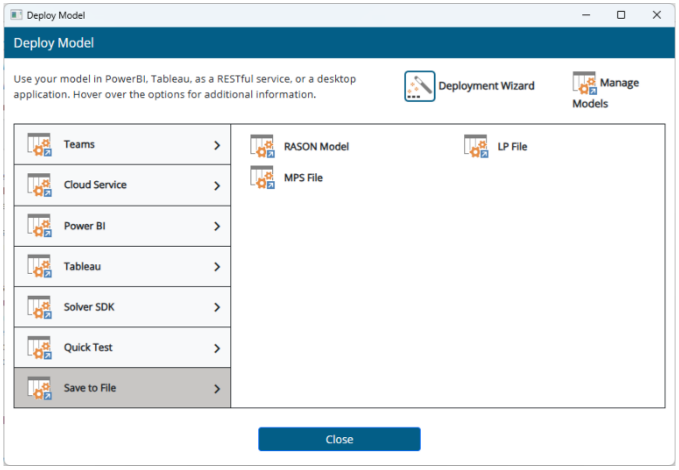
Fit Data (formerly available via the Tools icon, now here) replaces another popular feature, **Convert @RISK Model**. This option (which you'll need only once, if at all) is now available via the **Tools** icon.



Easier to Find "LP or MPS Model Format"

Another popular feature formerly available under the **Tools** icon, often used to help "debug" linear or linear mixed-integer programming models, allow you to *write out* your model as an "**LP format file**". This is a text file where every constraint is written in "algebraic linear coefficient form". An older and even more basic file format, called a "MPS format file", can also be written out.

From customer feedback, we concluded that this feature belonged with our *full array* of m*odel deployment* features, accessible in the dialog that appears for the **Deploy Model** icon:



Use your model in PowerBI, Tableau, as a RESTful service, or a desktop application. Hover over the options for additional information.

If you haven't previously spent time with the **Deploy Model** dialog, we *encourage* you to spend some time here: It's the **easy way to share** your model results, or even your full solvable model, with **others in your company**, using Microsoft Teams, Power BI or Tableau, or our RASON Cloud Service.

With a ***few mouse clicks***, you can turn your Excel model into a visual working model on your Power BI or Tableau dashboard – or *even* turn it into a “cloud service” readily *usable* from other *web* or *mobile* apps. If you’re not sure, just click the **Deployment Wizard** icon at the top of the dialog!

Analytic Solver Product Line

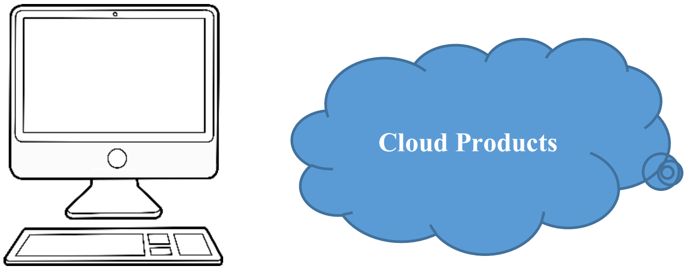
This Guide shows you how to create and evaluate forecasting and data science models using Analytic Solver.

In Frontline Solvers V2025 Q4, every license starts with **Analytic Solver Upgrade**, which allows you to **use every feature** described in this User Guide and the companion Reference Guide, for learning purposes with **small models**. Upgrade versions enable you to ‘scale up’ and solve commercial-size models for optimization, simulation or data science, paying for only what you need – but you ***keep*** access to all the features of Analytic Solver at the basic problem limits.

Analytic Solver combines and integrates the features of Frontline’s products for conventional optimization (formerly called **Premium Solver Pro** and **Premium Solver Platform**), Monte Carlo simulation and stochastic optimization (formerly **Risk Solver Pro** and **Risk Solver Platform**), and forecasting and data science (formerly **XLMiner Pro** and **XLMiner Platform**), in a common user interface that’s available both in Excel (desktop) and in your browser (cloud).

Analytic Solver’s **optimization** features are **fully compatible upgrades for the Solver** bundled within Microsoft Excel, which was developed by Frontline Systems for Microsoft. Your Excel Solver models and macros will work without changes; you can use either the classical **Solver Parameters dialog**, or a newer Task Pane user interface to define optimization models.

Desktop and Cloud versions



Analytic Solver V2025 Q4 includes the use of Analytic Solver Cloud and Data Science Cloud – accessible through Excel for the Web or desktop Excel. Use Analytic Solver Cloud to solve optimization and simulation models and use the Data Science Cloud app to perform data science or forecasting. Both the Analytic Solver and Data Science Cloud apps support existing models created in previous versions of Analytic Solver. Your license for Analytic Solver will allow you to use Analytic Solver Desktop in desktop Excel or Analytic Solver Cloud in either desktop Excel[1] *or* Excel for the Web. A license for a desktop product will grant you a license for the corresponding product in a cloud app. For example, a license for Analytic Solver Optimization in Anlaytic Solver

[1] Must be using Microsoft Excel V2020 or later.

Desktop will also grant you a license for Analytic Solver Optimization in the Analytic Solver Cloud app. The overwhelming majority of features in Analytic Solver Desktop are also included in Analytic Solver Cloud and Data Science Cloud. However, there could be slight differences in the way users execute a function in the cloud apps.

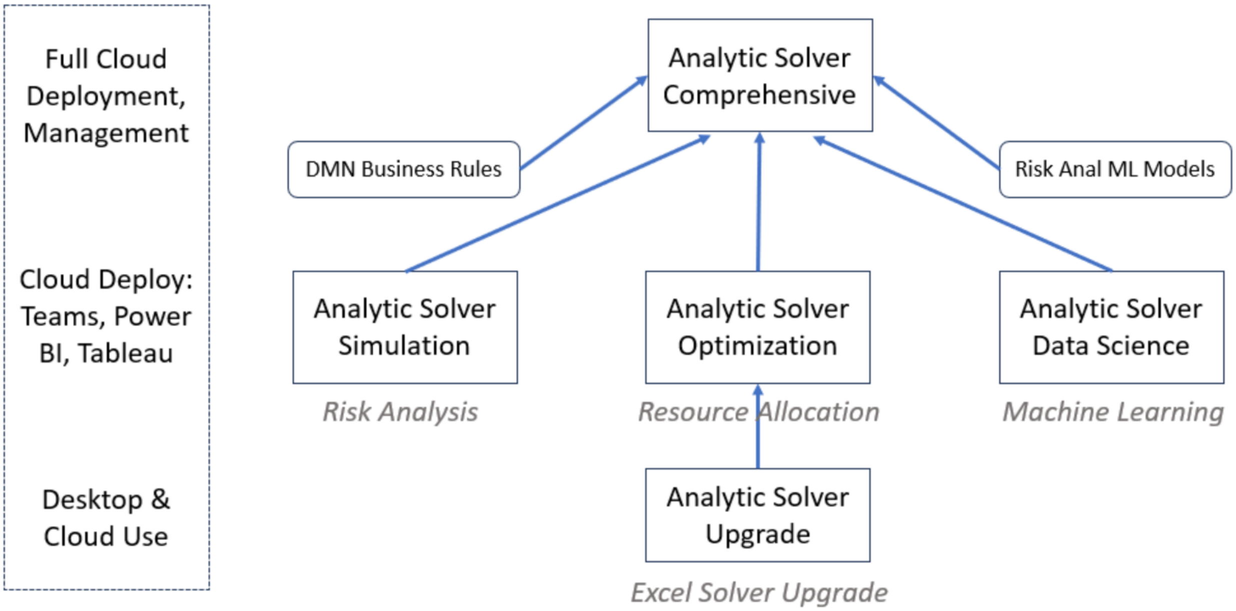
Analytic Solver for desktop or cloud may be purchased in several different ways starting with the most basic version, Analytic Solver Upgrade, up to our most complete version, Analytic Solver Comprehensive. Continue reading to see which product will best meet your needs.

Analytic Solver Academy

Analytic Solver Academy allows access to Frontline Systems' Solver Academy courses: Optimization 101, Simulation 101, Data Science 101 and Data Science 201. These four courses will quickly bring users up to speed on the complete functionality of Analytic Solver. If no other license exists, this product offers a basic course license with size limits for optimization, Monte Carlo simulation, data science and text mining, sufficient to run all installed example models and complete all course work.

Product Licenses

Frontline Systems offers five annual licenses for our Desktop Excel and Excel Online product suite: Analytic Solver Upgrade, Analytic Solver Optimization, Analytic Solver Simulation, Analytic Solver Data Science, and Analytic Solver Comprehensive. The graphic below illustrates how each of the subsets roll into Analytic Solver Comprehensive, our flagship product.


Full Cloud Deployment, Management
Cloud Deploy: Teams, Power BI, Tableau
Desktop & Cloud Use
Analytic Solver Comprehensive
DMN Business Rules
Risk Anal ML Models
Analytic Solver Simulation
Analytic Solver Optimization
Analytic Solver Data Science
Risk Analysis
Resource Allocation
Machine Learning
Analytic Solver Upgrade
Excel Solver Upgrade

Analytic Solver Upgrade

Frontline Systems is Frontline's basic upgrade for the Excel Solver – enabling you to solve linear up to 10 times larger (up to 2,000 variables), quadratic (QPs) and quadratically constrained problems (QCPs) and nonlinear models 2.5 times larger (up to 500 variables). It includes faster versions of the LP/Quadratic, GRG Nonlinear, and Evolutionary Solvers), but it doesn't support plug-in large-scale Solver Engines, decision tables, box functions, model deployment or the AI Agent. This product includes basic level support for simulation and data science models.

Analytic Solver Optimization, Simulation, Data Science and Comprehensive

Users of the next three products are granted a license to share the results of their optimization or simulation model to Power BI, Tableau or Microsoft Teams via the Deploy Model button on the Analytic Solver tab. See the *Deploying Your Model* chapter within the Analytic Solver User Guide for more details.

Analytic Solver Optimization

Analytic Solver Optimization (formerly **Premium Solver Platform**) is Frontline's most powerful product for *conventional* optimization. It includes the PSI Interpreter, five built-in Solvers (LP/Quadratic, SOCP Barrier, GRG Nonlinear, Interval Global, and Evolutionary), solves linear, quadratic and quadratically constrained models up to 8,000 variables and nonlinear models up to 1,000 variables, and it supports plug-in large-scale Solver Engines to handle much larger models. When used with Analytic Solver Simulation, you can also solve models *with uncertainty* using simulation optimization, stochastic linear programming, and robust optimization. This product, alone, includes all the features of Analytic Solver Upgrade with limited size constraints.

Analytic Solver Simulation

Analytic Solver Simulation (***expanded*** from **Risk Solver Pro**) is Frontline's full-function product for Monte Carlo simulation and simulation optimization. It includes decision tree capabilities and the PSI Interpreter – which gives you the fastest Monte Carlo simulations available in any Excel-based products, unique interactive simulation capabilities, multiple parameterized simulations, and simulation optimization using the Evolutionary Solver. When coupled with Analytic Solver Optimization, you can also solve models with uncertainty using stochastic linear programming and robust optimization. This product, alone, includes all the features of Analytic Solver Upgrade with limited size constraints.

Automated Risk Analysis of Machine Learning Models

Analytic Solver Data Science and Analytic Solver Comprehensive offer capability for **automated risk analysis** of **machine learning** models along with **synthetic data generation**, to augment the data you already have. The beauty of these new features is that **you don't have to do any work** to obtain a risk analysis of your model's performance, beyond a few mouse clicks on the **Simulation** tab to enable the analysis, which is entirely **automated**. You don't even have to be familiar with the features of Analytic Solver Simulation to use this capability for machine learning!

Analytic Solver Data Science

Analytic Solver Data Science (formerly XLMiner Platform and more recently Analytic Solver Data Mining) is Frontline's most powerful product for data science, text mining, forecasting and predictive analytics. It includes data access and sampling, data exploration and visualization, text mining, data transformation, and feature selection capabilities; time series forecasting with ARIMA and exponential smoothing; and a wide range of data science methods for classification, prediction and affinity analysis, from multiple regression to neural networks. This product includes all the features of Analytic Solver Upgrade.

Analytic Solver Comprehensive

Analytic Solver Comprehensive (formerly **Analytic Solver Platform**) combines the optimization capabilities of Analytic Solver Optimization, the simulation capabilities of Analytic Solver Simulation, and the data science capabilities of Analytic Solver Data Science (formerly XLMiner and more recently Analytic Solver Data Mining). It includes the PSI Interpreter, five built-in Solvers (LP/Quadratic, SOCP Barrier, GRG Nonlinear, Interval Global, and Evolutionary) and it accepts a full range of plug-in large- scale Solver Engines. It supports optimization, Monte Carlo simulation, simulation optimization, stochastic programming and robust optimization, and large-scale data science and forecasting capabilities.

See the chart below to compare the data handling limits between Analytic Solver Data Science and the remaining subsets: Analytic Solver Optimization/Simulation. (Analytic Solver is available for classroom use or by purchasing a textbook that contains the software. For a complete list of textbooks that include Analytic Solver, see our Website at www.solver.com.) "Unlimited" indicates that no limit is imposed, however, other limits may apply based on your computer resources and Excel version.

bn	Data Science/Comprehensive	Subsets
Risk Analysis and Synthetic Data Generation	Unlimited	Not supported
Partitioning		
# of Records (original data)	Unlimited	65,000
# of Records (training partition)	Unlimited	10,000
# of Variables (output)	Unlimited	50
Sampling		
# of Records (original data)	Unlimited	65,000
# of Variables (output)	Unlimited	50
# of Strata (Stratified Sampling)	Unlimited	30
Database		
# of Records (table)	Unlimited	1,000,000
# of Records (output)	Unlimited	65,000
# of Variables (table)	Unlimited	Unlimited
# of Variables (output)	Unlimited	50
# of Strata (Stratified Sampling)	Unlimited	30
File System		
# of Files	Unlimited	100
Text Mining		
# Documents	Unlimited	100

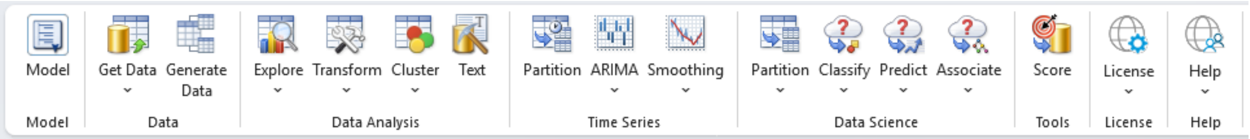
# Characters (per document)	Unlimited	5,000
# Terms in final vocabulary	Unlimited	50
# Text columns	Unlimited	1
Transformation		
Common		
# Records	Unlimited	10,000
# Variables	Unlimited	50
Missing Data Handling		
# of Records	Unlimited	65,000
# of Variables	Unlimited	50
Binning Continuous Data		
# of Records	Unlimited	65,000
Transforming Categorical Data		
# of Records	Unlimited	65,000
# of Variables (data range)	Unlimited	50
# of Distinct values	Unlimited	30
Time Series Analysis		
# of Records	Unlimited	1,000
Classification and Prediction		
# of Records (total)	Unlimited	65,000
# of Records (training partition)	Unlimited	10,000
# of Records (new data for scoring)	Unlimited	65,000
# of Variables (output)	Unlimited	50
# of Distinct classes (output variable)	Unlimited	30
# of Distinct values (categorical input variables)	Unlimited	15
k-Nearest Neighbors		
# of Nearest neighbors	50	10
Regression/Classification Trees		
# of Splits	Unlimited	100
# of Nodes	Unlimited	100
# of Levels	Unlimited	100
# Levels in Tree Drawing	Unlimited	7
Ensemble Methods		
# Weak learners	Unlimited	10
Feature Selection		
# of Records	Unlimited	10,000
# of Variables	Unlimited	50
# of Distinct classes (output variable)	Unlimited	30
# of Distinct values (input variables)	Unlimited	100
Association Rules		
# of Transactions	Unlimited	65,000

# of Distinct items	5,000	100
Clustering		
K-Means		
# of Records	Unlimited	10,000
# of Variables	Unlimited	50
# of Clusters	Unlimited	10
# of Iterations	Unlimited	50
Hierarchical		
# of Records	Unlimited	10,000
# of Variables	Unlimited	50
# of Clusters in a Dendrogram	Unlimited	10
Size of distance matrix	Unlimited	1,000 x 1,000
Charts		
# of Records	Unlimited	65,000
# of Variables (original data)	Unlimited	100
General		
Model pane	Included	Included
Big Data sampling/summarization	Included	Included
Model storage and scoring	Included	Included

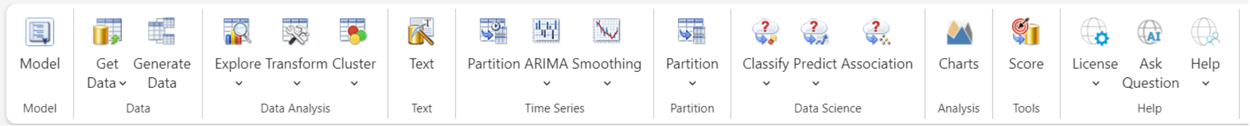
Data Science Ribbon Overview

Analytic Solver Data Science software offers over 30 different methods for analyzing a dataset in order to forecast future events. The Data Science ribbon is broken up into five different segments as shown in the screenshot below.

Desktop Analytic Solver Data Science



Data Science Cloud



- Click the Model button to display the Solver Task Pane. This new feature (added in V2016) allows you to quickly navigate through datasets and worksheets containing Analytic Solver Data Science results.
- Click the **Get Data** button to draw a random sample of data, or summarize data from a (i) an Excel worksheet, (ii) the PowerPivot "spreadsheet data model" which can hold 10 to 100 million rows of data in Excel, (iii) an external SQL database such as Oracle, DB2 or SQL Server, or (iv) a dataset

with up to billions of rows, stored across many hard disks in an external **Big Data** compute cluster running Apache Spark (https://spark.apache.org/).

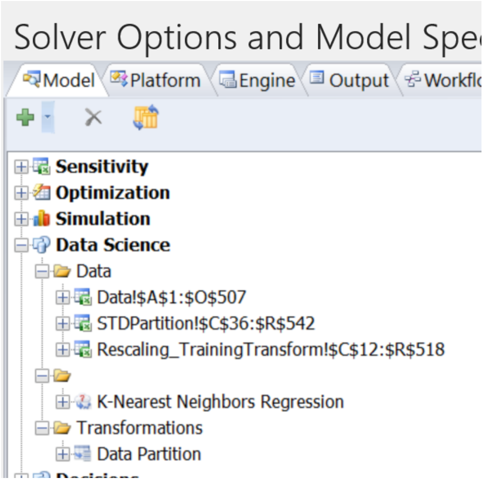
- You can use the **Data Analysis** group of buttons to explore your data, both visually and through methods like cluster analysis, transform your data with methods like Principal Components, Missing Value imputation, Binning continuous data, and Transforming categorical data, or use the Text Mining feature to extract information from text documents.
- Use the **Time Series** group of buttons for time series forecasting, using both Exponential Smoothing (including Holt-Winters) and ARIMA (Auto-Regressive Integrated Moving Average) models, the two most popular time series forecasting methods from classical statistics. These methods forecast a single data series forward in time.
- The **Data Science** group of buttons give you access to a broad range of methods for prediction, classification and affinity analysis, from both classical statistics and data science. These methods use multiple input variables to predict an outcome variable or classify the outcome into one of several categories. Introduced in V2015, Analytic Solver Data Science and now the Data Science Cloud app, offer Ensemble Methods for use with Classification Trees, Regression Trees, and Neural Networks.
- Use the **Predict** button to build prediction models using Multiple Linear Regression (with variable subset selection and diagnostics), k-Nearest Neighbors, Regression Trees, and Neural Networks. Use Ensemble Methods with Regression Trees and Neural Networks to create more accurate prediction models.
- Use the **Classify** button to build classification models with Discriminant Analysis, Logistic Regression, k-Nearest Neighbors, Classification Trees, Naïve Bayes, and Neural Networks. Use Ensemble Methods with Classification Trees and Neural Networks to create more accurate classification models.
- Use the **Associate** button to perform affinity analysis (“what goes with what” or market basket analysis) using Association Rules.
- Use the **Score** button to score new data using a fitted forecasting, classification or prediction model.
- Use the **License** button to manage your account and licenses.
- Use the **Help** button to open example models, open the Help Center, where you can find pre-recorded webinars or access our Knowledge Base or explore our User Guides.

If you’d like to learn more and get started as a ‘data scientist,’ consult the excellent book *Data Mining for Business Intelligence*, which was written by the original Data Science (formally known as XLMiner and most recently Analytic Solver Data Mining) designers and early academic users. You’ll be able to run all the Data Science examples and exercises in Analytic Solver.

Analytic Solver Data Science, along with the Data Science Cloud app, can be purchased as a stand-alone product. A stand-alone license for Analytic Solver Data Science includes all of the data analysis, time series data capabilities, classification and prediction features available in Analytic Solver Comprehensive but does not support optimization or simulation. See the Analytic Solver Data Science User Guide Data Specifications for each product.

Model

Click the Model button in Analytic Solver Desktop to display the Solver Task Pane within Analytic Solver Data Science. From the Model tab, you can easily navigate between worksheets containing data and results. Note that in the Data Science Cloud app, only the Workflow tab is present on the task pane.



Solver Options and Model Spe

Model | Platform | Engine | Output | Workflc

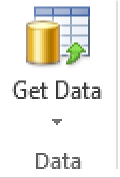
- Sensitivity
- Optimization
- Simulation
- Data Science
  - Data
    - Data!$A$1:$O$507
    - STDPartition!$C$36:$R$542
    - Rescaling_TrainingTransform!$C$12:$R$518
  -
    - K-Nearest Neighbors Regression
  - Transformations
    - Data Partition

All fields contained in the dataset are listed under the name of each data containing worksheet (for example Data!$A$1:$O$507) while results obtained from Analytic Solver Data Science are listed under Reports or Transformations by type and run number.

See the next chapter, Creating Workflows for information on how to create a workflow in Analytic Solver Data Science.

Get Data

Analytic Solver Data Science includes several different methods for importing your data including Sampling from either a Worksheet or Database or Importing from a File Folder.

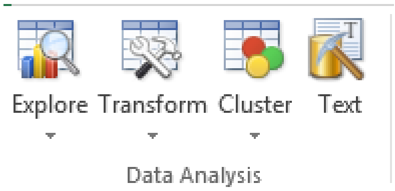


Click the **Get Data** icon to take a representative sample from a dataset included in either an Excel workbook or an Oracle, SQL Server, MS-Access, or Power Pivot database. Users can choose to sample with or without replacement using simple or stratified random sampling. Click **Get Data – File Folder** to import and or sample from a collection of text documents for use with Text Mining.

Click the **Get Data - Big Data** to sample from or summarize from a dataset with up to billions of rows, stored across many hard disks in an external compute cluster running Apache Spark. Results may be requested immediately upon completion or at a later time using **Get Results**.

Data Analysis

Analytic Solver Data Science includes several different methods for data analysis, including Sampling from either a Worksheet or Database (not supported in the Cloud app), Charting with 8 different types of available charts, Transformation techniques which handle missing data, binning continuous data, creating dummy variables and transforming categorical data, and using Principal Components Analysis to reduce and eliminate superfluous or redundant variables; along with two different types of Clustering techniques, k-Means and Hierarchical.



Click the Explore icon to use Feature Selection to help decide which variables should be included in your classification or prediction models or use the Chart Wizard to create one or more charts of your data. The new Feature Selection tool can help give insight into which variables are the most important or relevant for inclusion in your classification or prediction model using various types of statistics and data analysis measures. Analytic Solver Data Science includes 8 different types of charts to choose from, including: bar charts, line charts, scatterplots, boxplots, histograms, parallel coordinates charts, scatterplot matrix charts or variable charts. Click this icon to edit or view previously created charts as well.

Click the Transformation icon when data manipulation is required. In most large databases or datasets, a portion of variables are bound to be missing some data. Analytic Solver Data Science includes routines for dealing with these missing values by allowing a user to either delete the full record or apply a value of her/his choice.   Analytic Solver Data Science also includes a routine for binning continuous data for use with prediction and classification methods which do not support continuous data. Continuous variables can be binned using several different user specified options. Non-numeric data can be transformed using dummy variables with up to 30 distinct values. If more than 30 categories exist for a single variable, use the Reduce Categories routine to decrease the number of categories to 30. Finally, use Principal Components Analysis to remove highly correlated or superfluous variables from large databases.

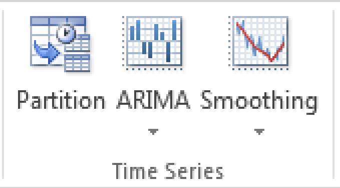
Click the Cluster icon to gain access to two different types of clustering techniques: k-Means clustering and hierarchical clustering. Both methods allow insight into a database or dataset by performing a cluster analysis. This type of analysis can be used to obtain the degree of similarity (or dissimilarity) between the individual objects being clustered.

Click the Text icon to use the Text Miner tool to analyze a collection of text documents for patterns and trends. (In the Cloud app, this tool is included in the Text section of the Ribbon.) These algorithms can categorize documents, provide links between documents that were not otherwise noted and create visual maps of the documents. Analytic Solver Data Science takes an integrated approach to text mining by combining text processing and analysis in a single package. While Analytic Solver Data Science is effective for mining “pure text” such as a set of documents, it is especially useful for “integrated text and data science” applications such as maintenance reports, evaluation forms, or any situation where a combination of structured data and free-form text data is available.

Time Series Analysis

Analytic Solver Data Science also supports the analysis and forecasting of datasets that contain observations generated sequentially such as predicting next year's sales figures, monthly airline bookings, etc. through partitioning, autocorrelations or ARIMA models and through smoothing techniques.

A time series model is first used to obtain an understanding of the underlying forces and structure that produced the data and then secondly, to fit a model that will predict future behavior. In the first step, the analysis of the data, a model is created to uncover seasonal patterns or trends in the data, for example bathing suit sales in June. In the second step, forecasting, the model is used to predict the value of the data in the future, for example, next year's bathing suit sales. Separate modeling methods are required to create each type of model.



Typically, when using a time series dataset, the data is first partitioned into training and validation sets. Click the Partition icon within the Time Series ribbon segment to utilize the Time Series Data Partitioning routine. Analytic Solver Data Science features two techniques for exploring trends in a dataset, ACF (Autocorrelation function) and PACF (Partial autocorrelation function). These techniques help the user to explore various patterns in the data which can be used in the creation of the model. After the data is analyzed, a model can be fit to the data using the ARIMA method. All three of these methods can be found by clicking the ARIMA icon on the Data Science ribbon.

Data collected over time is likely to show some form of random variation. "Smoothing techniques" can be used to reduce or cancel the effect of these variations. These techniques, when properly applied, will “smooth” out the random variation in the time series data to reveal any underlying trends that may exist.

Click the Smoothing icon to gain access to Analytic Solver Data Science's four different smoothing techniques: Exponential, Moving Average, Double Exponential, and Holt Winters. The first two techniques, Exponential and Moving Average, are relatively simple smoothing techniques and should not be performed on datasets involving trends or seasonality. The third technique, Double Exponential, should be used when a trend is present in the dataset, but not seasonality. The last technique, Holt Winters, is a more advanced technique and should be selected when working with datasets involving seasonality.

Data Science

The Data Science section of the Data Science ribbon contains four icons: Partition, Classify, Predict, and Associate. Note that in the Cloud app, Partition has been cordoned off into it's own "Partition" section. Click the Partition icon to partition your data into training, validation, and if desired, test sets. Click the

Classify icon to select one of six different classification methods. Click the Predict icon to select one of four different prediction methods. Click the Associate icon to recognize associations or correlations among variables in the dataset.

Analytic Solver Data Science supports six different methods for predicting the class of an outcome variable (classification) along with three ensemble methods which use these six methods as weak learners, and four different methods, along with three ensemble methods, for predicting the actual (prediction) of an outcome variable. Classification can be described as categorizing a set of observations into predefined classes in order to determine the class of an observation based on a set of variables. A prediction method can be described as a technique performed on a database either to predict the response variable value based on a predictor variable or to study the relationship between the response variable and the predictor variables. For example, when determining the relationship between the crime rate of a city or neighborhood and demographic factors such as population, education, male to female ratio, etc.

One very important issue when fitting a model is how well the newly created model will behave when applied to new data. To address this issue, the dataset can be divided into multiple partitions before a classification or prediction algorithm is applied: a training partition used to create the model, a validation partition to test the performance of the model and, if desired, a third test partition. Partitioning is performed randomly, to protect against a biased partition, according to proportions specified by the user or according to rules concerning the dataset type. For example, when creating a time series forecast, data is partitioned by chronological order.

The six different classification methods are:

Discriminant Analysis - Constructs a set of linear functions of the predictor variables and uses these functions to predict the class of a new observation with an unknown class. Common uses of this method include: classifying loan, credit card or insurance applicants into low or high risk categories, classifying student applications for college entrance, classifying cancer patients into clinical studies, etc.

Logistic Regression – A variant of ordinary regression which is used to predict the response variable, or the output variable, when the response variable is a dichotomous variable (a variable that takes only two values such as yes/no, success/failure, survive/die, etc.).

k-Nearest Neighbors – This classification method divides a training dataset into groups of *k* observations using a Euclidean Distance measure to determine similarity between “neighbors”. These classification groups are used to assign categories to each member of the validation training set.

Classification Tree – Also known as Decision Trees, this classification method is a good choice when goal is to generate easily understood and explained “rules” that can be translated in an SQL or query language.

Naive Bayes – This classification method first scans the training dataset and finds all records where the predictor values are equal. Then the most prevalent class of the group is determined and assigned to the entire collection of observations. If a new observation’s predictor variable equals the predictor variable of this group, the new observation will be assigned to this class. Due to the simplicity of this method a large number of records are required to obtain accuracy.

Neural Network – Artificial neural networks are based on the operation and structure of the human brain. These networks process one record at a time and "learn" by comparing their classification of the record (which as the beginning is largely arbitrary) with the known actual classification of the record. Errors from the initial classification of the first records are fed back into the network and used to modify the networks algorithm the second time around. This continues for many, many iterations.

The four different predictive methods are:

Multiple Linear Regression – This method is performed on a dataset to predict the response variable based on a predictor variable or used to study the relationship between a response and predictor variable, for example, student test scores compared to demographic information such as income, education of parents, etc.

k-Nearest Neighbors – Like the classification method with the same name above, this prediction method divides a training dataset into groups of *k* observations using a Euclidean Distance measure to determine similarity between "neighbors". These groups are used to predict the value of the response for each member of the validation set.

Regression Trees - A Regression tree may be considered a variant of a decision tree, designed to approximate real-valued functions instead of being used for classification methods. As with all regression techniques, Analytic Solver Data Science assumes the existence of a single output (response) variable and one or more input (predictor) variables. The output variable is numerical. The general regression tree building methodology allows input variables to be a mixture of continuous and categorical variables. A decision tree is generated when each decision node in the tree contains a test on some input variable's value. The terminal nodes of the tree contain the predicted output variable values.

Neural Network – Artificial neural networks are based on the operation and structure of the human brain. These networks process one record at a time and "learn" by comparing their prediction of the record (which as the beginning is largely arbitrary) with the known actual value of the response variable. Errors from the initial prediction of the first records are fed back into the network and used to modify the networks algorithm the second time around. This continues for many, many iterations.

Three ensemble methods, bagging, boosting, and random trees, are also available for both classification and prediction. Each of these methods uses a classification or prediction method as a weak learner. Each of these can be accessed on the button of either the Classify or Predict menus.

The goal of association rule mining is to recognize associations and/or correlations among large sets of data items. A typical and widely-used example of association rule mining is the Market Basket Analysis. Most 'market basket' databases consist of a large number of transaction records where each record lists all items purchased by a customer during a trip through the check-out line. Data is easily and accurately collected through the bar-code scanners. Supermarket managers are interested in determining what foods customers purchase together, like, for instance, bread and milk, bacon and eggs, wine and cheese, etc. This information is useful in planning store layouts (placing items optimally with respect to each other), cross-selling promotions, coupon offers, etc.

Scoring

Click the Score icon to score new data in a database or worksheet with any of the Classification or Prediction algorithms. This facility matches the input variables to the database (or worksheet) fields and then performs the scoring on the database (or worksheet).

Analytic Solver Data Science also supports the scoring of Test Data. When Analytic Solver Data Science calculates prediction or classification results, internal values and coefficients are generated and used in the computations. Analytic Solver Data Science saves these values to an additional output sheet, termed Stored Model Sheet, which uses the output sheet name, XX_Stored_N where XX are the initials of the classification or prediction method and N is the number of generated stored sheets. This sheet is used when scoring the test data.

Note: In previous versions of XLMiner, this utility was a separate add-on application named XLMCalc. Starting in XLMiner V12.5, this utility is included free of charge.   Starting in V2014-R2, PsiClassify(), PsiPredict() and PsiForecast functions are available for instantaneous interactive scoring on the worksheet without the need to click the Score icon.

Using Help, Licensing and Product Subsets

Introduction

Analytic Solver Data Science (previously referred to as XLMiner™ and more recently Analytic Solver Data Mining) is a comprehensive data science software package for use on the Web or as an add-in to Excel. Data science is a discovery-driven data analysis technology used for identifying patterns and relationships in data sets. With overwhelming amounts of data now available from transaction systems and external data sources, organizations are presented with increasing opportunities to understand their data and gain insights into it. Data science is still an emerging field, and is a convergence of fields like statistics, machine learning, and artificial intelligence.

Often, there may be more than one approach to a problem. Analytic Solver Data Science is a tool belt to help you get started quickly by offering a variety of methods to analyze your data. It has extensive coverage of statistical and machine learning techniques for classification, prediction, affinity analysis and data exploration and reduction.

This chapter describes the ways Analytic Solver V2018 handles the overall operation, including registration, licensing, use of product subsets, and use of the Startup Screen, online Help and examples.

Working with Licenses in V2025 Q4

A **license** is a grant of rights, from Frontline Systems to you, to use our software in specified ways. Information about a license – for example, its temporary vs. permanent status and its expiration date – is encoded in a **license code**. The same binary files are used for all Analytic Solver products . The product features you see depend on the license code you have.

Frontline License Manager:

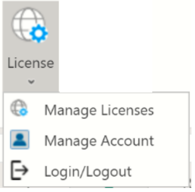
In Analytic Solver V2025 Q4, the Frontline License Manager has replaced the Reprise License Manager – its basic purpose is license control (allowing the software to run, or not). But unlike Reprise, the new Frontline License Manager ties a license to a human user ID / email, not to a hardware ID or "lock code". Starting with a user ID stored locally, it will ask for license rights for a user via a web (REST API) request to Frontline's License Manager, and store this license information locally. When our software is first installed, it will have an embedded user ID *and* license code for a trial or evaluation license, so the software can run for the trial period even without Internet access.

Upgrading from an Early Version of Desktop ASP

Analytic Solver Desktop V2025 Q4 can store license codes locally but typically your license will be stored on Frontline's license server. If you're upgrading from V2017/2018, a *new* V2025 Q4 license code will **not** be required. Simply download and install V2025 Q4 for Desktop Excel and your existing V2017/V2018 license will activate the newer version. Old license codes for V2016 and earlier have no negative effect in V2025 Q4. If they exist in the obsolete Solver.lic file (located at C:\ProgramData\Frontline Systems), they will be ignored. A license code will be issued at the time of purchase.

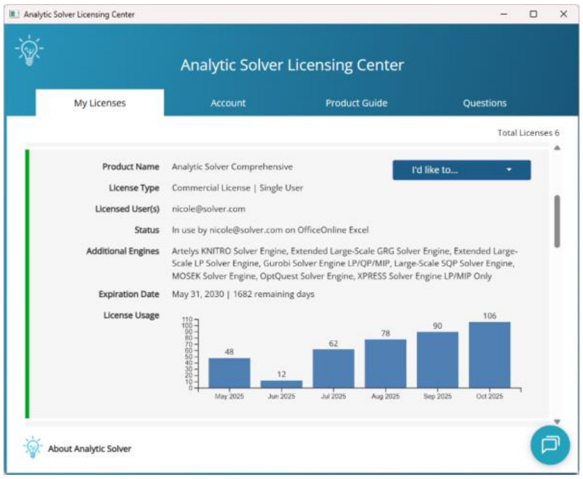
Managing Your License

Click the License button to login or logout of Analytic Solver.



Click License – Manage License/Manage Account to open the Licensing Center.

The "MyLicenses" tab displays your current license and license type, along with the expiration date. Click "I'd like to…" to switch to a different license type, explore upgrade options or chat with a live agent. If you have a trial license, or your license has expired, you can request a quote to renew your current license. The chart tracks the number of times the license has been checked out.



Total Licenses 6

Product Name Analytic Solver Comprehensive

License Type Commercial License | Single User

Licensed User(s) nicole@solver.com

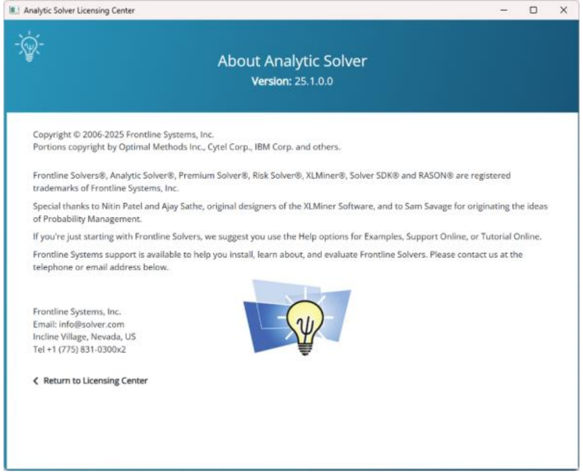
Status In use by nicole@solver.com on OfficeOnline Excel

Additional Engines Artelys KNITRO Solver Engine, Extended Large-Scale GRG Solver Engine, Extended Large-Scale LP Solver Engine, Gurobi Solver Engine LP/QP/MIP, Large-Scale SQP Solver Engine, MOSEK Solver Engine, OptQuest Solver Engine, XPRESS Solver Engine LP/MIP Only

Expiration Date May 31, 2030 | 1682 remaining days

License Usage

Click *About Analytic Solver* to open the following dialog containing information on this release.



About Analytic Solver

Version: 25.1.0.0

Copyright © 2006-2025 Frontline Systems, Inc.
Portions copyright by Optimal Methods Inc., Cytel Corp., IBM Corp. and others.

Frontline Solvers®, Analytic Solver®, Premium Solver®, Risk Solver®, XLMiner®, Solver SDK® and RASON® are registered trademarks of Frontline Systems, Inc.

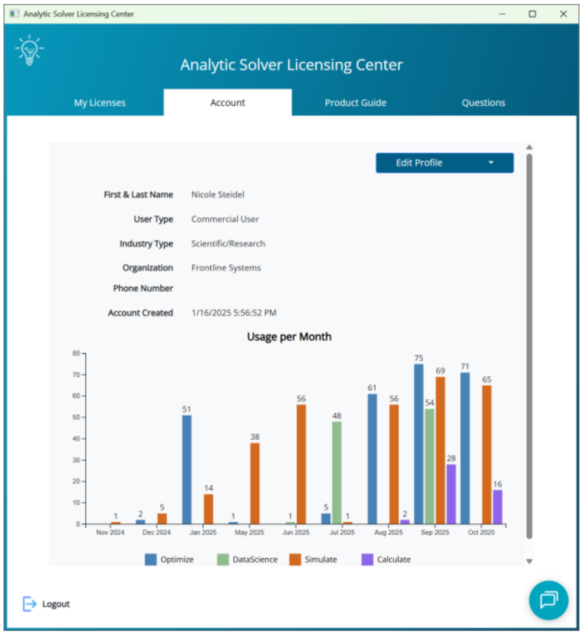
Special thanks to Nitin Patel and Ajay Sathe, original designers of the XLMiner Software, and to Sam Savage for originating the ideas of Probability Management.

If you're just starting with Frontline Solvers, we suggest you use the Help options for Examples, Support Online, or Tutorial Online.

Frontline Systems support is available to help you install, learn about, and evaluate Frontline Solvers. Please contact us at the telephone or email address below.

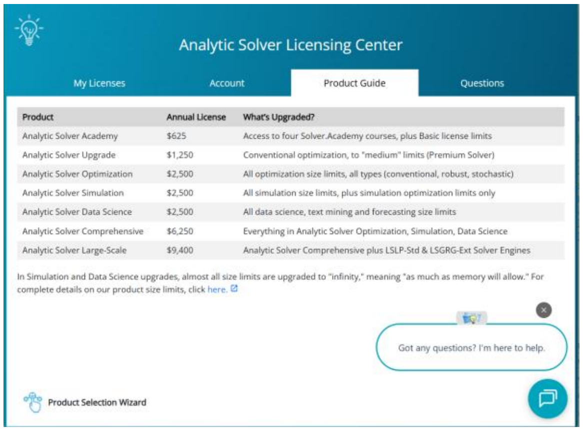
Frontline Systems, Inc.
Email: info@solver.com
Incline Village, Nevada, US
Tel +1 (775) 831-0300x2

Click the Account tab to view your account on www.solver.com. The chart displays the number of optimizations, simulations or data science methods performed. Click Edit Profile to edit the information. Click Live Chat to open a Live Chat window or Log Out to log out of the product.


Analytic Solver Licensing Center
Analytic Solver Licensing Center
My Licenses
Account
Product Guide
Questions
Edit Profile

First & Last Name Nicole Steidel
User Type Commercial User
Industry Type Scientific/Research
Organization Frontline Systems
Phone Number
Account Created 1/16/2025 5:56:52 PM

Click the Product Guide tab to view a list of products and pricing information. Click Product Selection Wizard to open the Product Selection Wizard. See the next section for information on this feature.

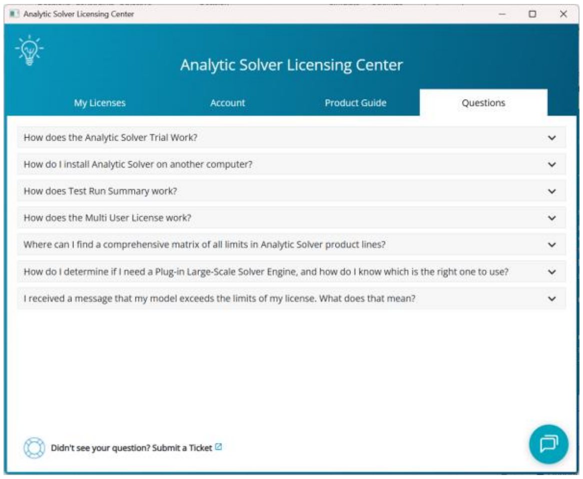


Product	Annual License	What's Upgraded?
Analytic Solver Academy	$625	Access to four Solver.Academy courses, plus Basic license limits
Analytic Solver Upgrade	$1,250	Conventional optimization, to "medium" limits (Premium Solver)
Analytic Solver Optimization	$2,500	All optimization size limits, all types (conventional, robust, stochastic)
Analytic Solver Simulation	$2,500	All simulation size limits, plus simulation optimization limits only
Analytic Solver Data Science	$2,500	All data science, text mining and forecasting size limits
Analytic Solver Comprehensive	$6,250	Everything in Analytic Solver Optimization, Simulation, Data Science
Analytic Solver Large-Scale	$9,400	Analytic Solver Comprehensive plus LSLP-Std & LSGRG-Ext Solver Engines

In Simulation and Data Science upgrades, almost all size limits are upgraded to "infinity," meaning "as much as memory will allow." For complete details on our product size limits, click here.

Got any questions? I'm here to help.

Click the Questions tab to review a list of FAQs, submit a support ticket or start a live chat.


Analytic Solver Licensing Center
Analytic Solver Licensing Center
My Licenses
Account
Product Guide
Questions

How does the Analytic Solver Trial Work?

How do I install Analytic Solver on another computer?

How does Test Run Summary work?

How does the Multi User License work?

Where can I find a comprehensive matrix of all limits in Analytic Solver product lines?

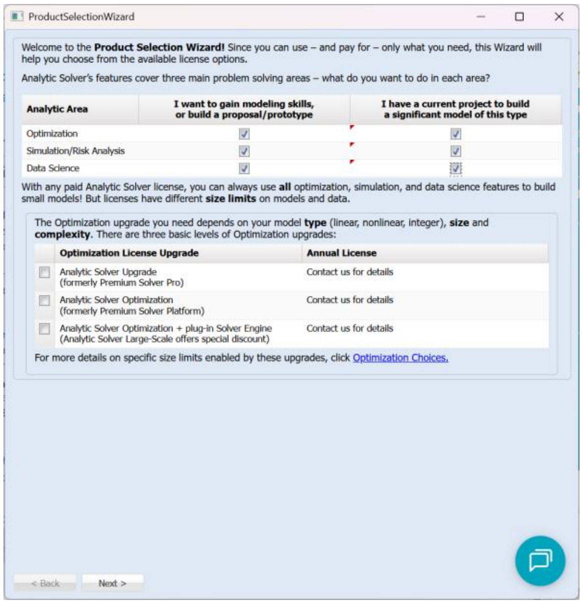
How do I determine if I need a Plug-in Large-Scale Solver Engine, and how do I know which is the right one to use?

I received a message that my model exceeds the limits of my license. What does that mean?

Didn't see your question? Submit a Ticket

Product Selection Wizard

Select **Product Selection Wizard** from the Product Guide tab in the Licensing Center to open a series of dialogs that will help you determine which product will best meet your needs based on your recent pattern of use.



Welcome to the **Product Selection Wizard!** Since you can use – and pay for – only what you need, this Wizard will help you choose from the available license options.

Analytic Solver's features cover three main problem solving areas – what do you want to do in each area?

Analytic Area	I want to gain modeling skills, or build a proposal/prototype	I have a current project to build a significant model of this type
Optimization	☑	☑
Simulation/Risk Analysis	☑	☑
Data Science	☑	☑

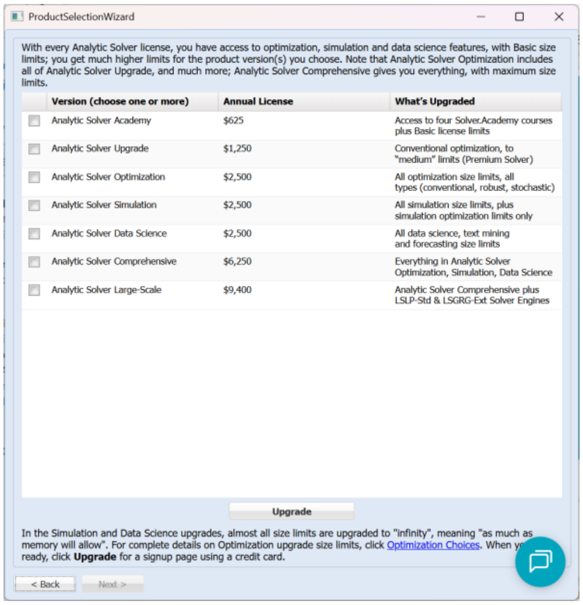
With any paid Analytic Solver license, you can always use **all** optimization, simulation, and data science features to build small models! But licenses have different **size limits** on models and data.

The Optimization upgrade you need depends on your model **type** (linear, nonlinear, integer), **size** and **complexity**. There are three basic levels of Optimization upgrades:

	Optimization License Upgrade	Annual License
□	Analytic Solver Upgrade (formerly Premium Solver Pro)	Contact us for details
□	Analytic Solver Optimization (formerly Premium Solver Platform)	Contact us for details
□	Analytic Solver Optimization + plug-in Solver Engine (Analytic Solver Large-Scale offers special discount)	Contact us for details

For more details on specific size limits enabled by these upgrades, click Optimization Choices.

Select the Product that you'd like to purchase and then click **Next**. Click the *Optimization Choices* link to learn more about Analytic Solver products that can solve optimization models and to find more information on speed, memory, and the use of plug-in Solver Engines.



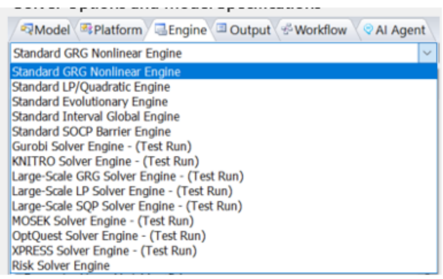
With every Analytic Solver license, you have access to optimization, simulation and data science features, with Basic size limits; you get much higher limits for the product version(s) you choose. Note that Analytic Solver Optimization includes all of Analytic Solver Upgrade, and much more; Analytic Solver Comprehensive gives you everything, with maximum size limits.

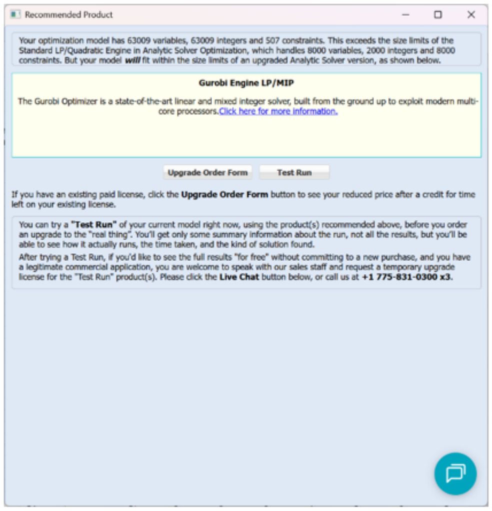
	Version (choose one or more)	Annual License	What's Upgraded
□	Analytic Solver Academy	$625	Access to four Solver.Academy courses plus Basic license limits
□	Analytic Solver Upgrade	$1,250	Conventional optimization, to "medium" limits (Premium Solver)
□	Analytic Solver Optimization	$2,500	All optimization size limits, all types (conventional, robust, stochastic)
□	Analytic Solver Simulation	$2,500	All simulation size limits, plus simulation optimization limits only
□	Analytic Solver Data Science	$2,500	All data science, text mining and forecasting size limits
□	Analytic Solver Comprehensive	$6,250	Everything in Analytic Solver Optimization, Simulation, Data Science
□	Analytic Solver Large-Scale	$9,400	Analytic Solver Comprehensive plus LSLP-Std & LSGRG-Ext Solver Engines

In the Simulation and Data Science upgrades, almost all size limits are upgraded to "infinity", meaning "as much as memory will allow". For complete details on Optimization upgrade size limits, click Optimization Choices. When yo ready, click **Upgrade** for a signup page using a credit card.

On this screen, the Product Selection Wizard will recommend a product or products based on your answers on the previous screens. Click Upgrade to purchase the recommended product. Click the *Optimization Choices* link to learn more about Analytic Solver products that can solve optimization models. If at any time you'd like to chat with a member of our Technical Support staff, click **Live Chat**. Or if you'd like to amend your answers on a previous dialog, click **Back**.

When you run a simulation or optimization model that contains too many decision variables/uncertain variables or constraints/uncertain functions for the selected engine, the Product Wizard will automatically appear and recommend a product that *can* solve your model.


Model
Platform
Engine
Output
Workflow
AI Agent
Standard GRG Nonlinear Engine
Standard GRG Nonlinear Engine
Standard LP/Quadratic Engine
Standard Evolutionary Engine
Standard Interval Global Engine
Standard SOCP Barrier Engine
Gurobi Solver Engine - (Test Run)
KNITRO Solver Engine - (Test Run)
Large-Scale GRG Solver Engine - (Test Run)
Large-Scale LP Solver Engine - (Test Run)
Large-Scale SQP Solver Engine - (Test Run)
MOSEK Solver Engine - (Test Run)
OptQuest Solver Engine - (Test Run)
XPRESS Solver Engine - (Test Run)
Risk Solver Engine



Your optimization model has 63009 variables, 63009 integers and 507 constraints. This exceeds the size limits of the Standard LP/Quadratic Engine in Analytic Solver Optimization, which handles 8000 variables, 2000 integers and 8000 constraints. But your model ***will*** fit within the size limits of an upgraded Analytic Solver version, as shown below.

Gurobi Engine LP/MIP

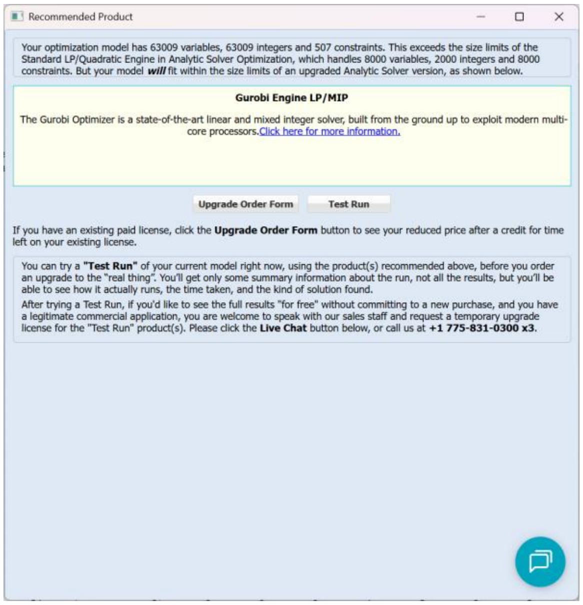
The Gurobi Optimizer is a state-of-the-art linear and mixed integer solver, built from the ground up to exploit modern multi-core processors.Click here for more information.

If you have an existing paid license, click the **Upgrade Order Form** button to see your reduced price after a credit for time left on your existing license.

You can try a **"Test Run"** of your current model right now, using the product(s) recommended above, before you order an upgrade to the "real thing". You'll get only some summary information about the run, not all the results, but you'll be able to see how it actually runs, the time taken, and the kind of solution found.

After trying a Test Run, if you'd like to see the full results "for free" without committing to a new purchase, and you have a legitimate commercial application, you are welcome to speak with our sales staff and request a temporary upgrade license for the "Test Run" product(s). Please click the **Live Chat** button below, or call us at **+1 775-831-0300 x3.**

When you click “Test Run”, the Product Wizard will immediately run the optimization or simulation model using the recommended product.   (Only summary information will be available.) At this point, you can purchase the recommended product(s), or close the dialog.



Your optimization model has 63009 variables, 63009 integers and 507 constraints. This exceeds the size limits of the Standard LP/Quadratic Engine in Analytic Solver Optimization, which handles 8000 variables, 2000 integers and 8000 constraints. But your model ***will*** fit within the size limits of an upgraded Analytic Solver version, as shown below.

Gurobi Engine LP/MIP

The Gurobi Optimizer is a state-of-the-art linear and mixed integer solver, built from the ground up to exploit modern multi-core processors.Click here for more information.

If you have an existing paid license, click the **Upgrade Order Form** button to see your reduced price after a credit for time left on your existing license.

You can try a **"Test Run"** of your current model right now, using the product(s) recommended above, before you order an upgrade to the "real thing". You'll get only some summary information about the run, not all the results, but you'll be able to see how it actually runs, the time taken, and the kind of solution found.

After trying a Test Run, if you'd like to see the full results "for free" without committing to a new purchase, and you have a legitimate commercial application, you are welcome to speak with our sales staff and request a temporary upgrade license for the "Test Run" product(s). Please click the **Live Chat** button below, or call us at **+1 775-831-0300 x3.**

This same behavior will also occur when solving smaller models, if you select a specific external engine, from the Engine drop down menu on the Engine tab of the Solver Task Pane, for which you do not have a license. The Product Wizard will recommend the selected engine, and allow you to solve your model using this engine. Once Solver has finished solving, you will have the option to purchase the product.

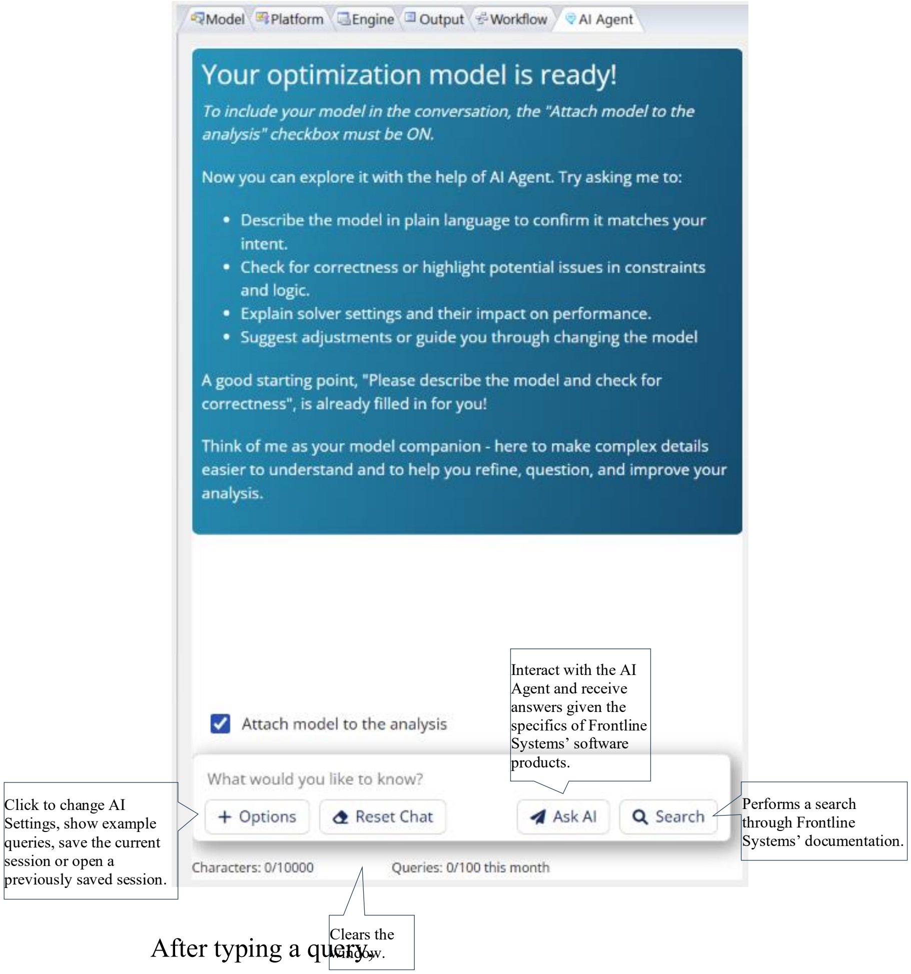
Getting Help

AI Agent

If you are new to Analytic Solver Data Science and/or optimization, simulation, forecasting and data science, don't worry – Frontline's AI technical support assistant, AI Agent, is here to help. Simply click the AI Agent tab on the Analytic Solver Task Pane. AI Agent is designed to provide assistance and support for users of Frontline Solvers' Analytic Solver and Analytic Solver Data Science software. The AI Agent is knowledgeable about the functionality and features of the software, as well as the concepts and processes involved in optimization, simulation and data science/forecasting. Just enter a topic or question such as "What classification algorithms are supported in Analytic Solver Data Science?" and click **Ask AI** to get started.

Example Queries

- How can I schedule 25 trucks with different capacities and shifts to minimize cost while covering all orders?
- If demand ranges 80-120 units, what's the chance we run out of stock with 100 units on hand?
- Can you build a decision table for loan approvals based on credit score, income, and debt?
- How can I classify incoming support tickets into categories automatically?
- Can I estimate the probability a new IT project will exceed its budget?



Your optimization model is ready!

To include your model in the conversation, the "Attach model to the analysis" checkbox must be ON.

Now you can explore it with the help of AI Agent. Try asking me to:

- Describe the model in plain language to confirm it matches your intent.
- Check for correctness or highlight potential issues in constraints and logic.
- Explain solver settings and their impact on performance.
- Suggest adjustments or guide you through changing the model

A good starting point, "Please describe the model and check for correctness", is already filled in for you!

Think of me as your model companion - here to make complex details easier to understand and to help you refine, question, and improve your analysis.

☑ Attach model to the analysis

What would you like to know?

Characters: 0/10000

Queries: 0/100 this month

Interact with the AI Agent and receive answers given the specifics of Frontline Systems' software products.

Click to change AI Settings, show example queries, save the current session or open a previously saved session.

Performs a search through Frontline Systems' documentation.

Clears the window.

After typing a query:

- Click **Submit Query** to interact with the AI Agent and receive answers given the specifics of Frontline Systems' software products. On each new question, the AI Agent will take into account the chat history that is present in the current session. Each extract contains a link to the appropriate page in one of the official documents.

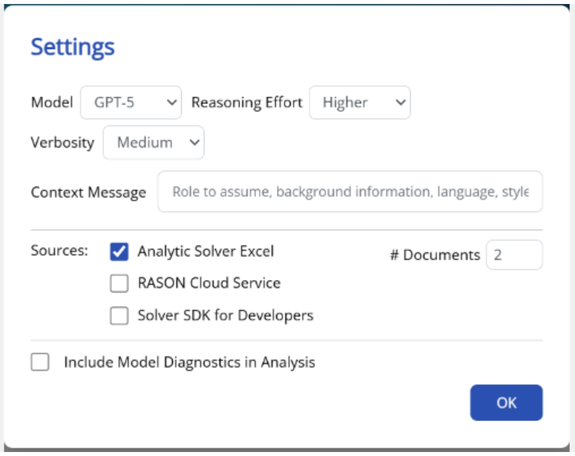
- Click **Search Documentation** (for faster search results) to perform a semantic/relevant search through all Analytic Solver and RASON documentation. The resulting response will be similar to the "Learn More" section as described above.

For more information on AI assistance in Analytic Solver, see the Analytic Solver User Guide.

Note: Frontline's AI Agent uses artificial intelligence which could produce erroneous information about people, places or facts, including misinformation concerning Frontline Systems software products. Frontline Systems bears no responsibility for these inaccuracies.

AI Agent Options

Click Options – Settings to open the AI Agent Settings dialog.



Settings

Model GPT-5 Reasoning Effort Higher

Verbosity Medium

Context Message: Role to assume, background information, language, style

Sources: ☑ Analytic Solver Excel # Documents 2

□ RASON Cloud Service

□ Solver SDK for Developers

□ Include Model Diagnostics in Analysis

Model: Use the down arrow to select either GPT-5 (the default) or GPT-4.1. While GPT-5 uses the latest AI technology, GPT-4.1 is included because of its time tested stability.

Reasoning Effort: In GPT-5's API, users can choose a level for this parameter, **lower** or **higher,** to trade off between **speed** and **depth of reasoning. Higher** means the model will engage in a more extensive internal chain-of-thought or multi-step planning before producing an answer.

Verbosity: Available when GPT-5 is selected, this parameter adjusts how **expansive or concise** the AI Agent's responses are—without needing to rewrite your prompt every time. It supports three levels:

- **low:** short, to-the-point answers with minimal prose
- **medium**: balanced level of detail (this is the default)
- **high**: rich, in-depth responses, ideal for documentation requests.

Context Message: Context Message lets users add their own background instructions that are sent with every AI Agent query. It's included in the prompt (along with retrieved document excerpts) to steer responses toward your specific model, goals, terminology, or tone. If left blank, no extra context is added. Note that the syntax in the results is LaTeX. To turn off this syntax, type "not latex" into the Context Message field and then click OK.

Sources: Select all sources to be used as input to AI Agent.

- Analytic Solver Excel: Sources are Analytic Solver User & Reference Guides and Analytic Solver Data Science User & Reference Guides plus all Excel examples found by clicking Help – Examples in Analytic Solver and Data Science tabs.
- RASON Cloud Service: Sources include RASON User and Reference guides plus all examples found by clicking RASON Examples on Editor tab of **www.RASON.com**.
- Solver SDK for Developers: Sources include Solver SDK User and Reference Guides, XLMiner SDK User Guide, plus a**ll SDK** examples installed to C:\ProgramFiles\Frontline Systems\Analytic Solver SDK\Solver SDK\Examples.

- **# Documents:** Controls how many documentation extracts the Agent retrieves and uses to source its answer. Higher values result in broader coverage (more sources considered) but this can result in slower and sometimes less focused answers. Lower values can result in faster, more focused responses but with narrower coverage. Accepts values [1, 10].

AI in Action

Enter the following prompt into AI Agent and click Ask AI.

“Optimize the inventory levels for a retail store to minimize stockouts and overstocking, while considering demand variability, lead tie, and holding costs.”

The AI Agent will respond with an output similar to the example shown below. That’s all there is to it! With AI Agent, in the latest version of Analytic Solver, you’ll have your first data science model up and running in no time!

```
Below is a practical, model-driven approach you can implement
(and solve) with Frontline’s Analytic Solver to minimize
stockouts and overstocking while accounting for demand
variability, lead time, and holding costs.

1) Assemble inputs
- Demand by SKU: mean per period and variability (e.g.,
daily/weekly mean μ, standard deviation σ). Use Analytic Solver’s
Forecasting to estimate μ and σ from history (e.g., Exponential
Smoothing/ARIMA), or fit distributions for simulation.
- Lead time L: average and variability (deterministic or a
distribution).
- Costs: holding cost h (per unit per period), ordering cost K
(per order), unit cost c, and shortage/stockout penalty p (per
unit short) or a target service level (e.g., cycle service level
CSL or fill rate FR).
- Constraints (optional): storage capacity, capital/budget,
supplier MOQs, case-pack multiples, integer order quantities.

2) Choose a replenishment policy
- Continuous review (s, Q): When inventory position ≤ s, order Q.
- Periodic review (R, S): Review every R periods; order up to S.
Both work with variable demand and lead time; (s, Q) is standard
for steady demand, (R, S) fits fixed review schedules.

3) Start with analytical baselines
- Economic Order Quantity: Q* = sqrt(2 K D / h), where D is the
average demand rate. This balances holding and ordering costs (no
constraints or shortages).
- Lead time demand:
- Mean μL = μ × L (for constant L; for random L use E[L] and
variance propagation).
- Std dev σL ≈ sqrt(L) × σ if demand is i.i.d.; otherwise compute
from your forecast/error model.
- Safety stock for cycle service level CSL: SS = z × σL, where z
= NORM.S.INV(CSL).
- Reorder point: s = μL + SS.
- Periodic review protection period P = L + R; S = μP + z × σP.

4) Extend to cost-based service (optional)
- If you prefer cost minimization over target CSL, use penalty p
to set z via the newsvendor critical fractile: CSL* ≈ p / (p +
hP) for a protection period P (continuous review: use P = L).
```

This gives z, hence SS and s. Then use EOQ for Q (or optimize Q directly).

5) Add real-world constraints (multi-item)
For SKU i:
- Average on-hand ≈ SS_i + Q_i/2 (continuous review).
- Capacity: sum_i space_i × (SS_i + Q_i/2) ≤ SpaceAvailable.
- Budget/capital: sum_i c_i × (SS_i + Q_i/2) ≤ Budget.
- Optional integrality or case-pack: Q_i ∈ multiples, s_i integer.

6) Build the optimization model in Analytic Solver
- Decision variables: For each SKU, choose s_i and Q_i (or R and S_i).
- Objective: Minimize expected total cost = sum_i [holding + ordering + shortage].
- Deterministic approximation:
- Holding ≈ h_i × (SS_i + Q_i/2)
- Ordering ≈ K_i × D_i / Q_i
- Shortage: enforce CSL/FR constraints instead of penalizing, or use cost p_i with an approximation.
- Simulation-based (recommended when variability is material):
- Use PSI distributions for demand and lead time.
- Compute realized service metrics and costs per trial.
- Objective = average simulated total cost; constraints = CSL or FR targets across trials.
- Constraints: Capacity, budget, service level (e.g., cycle service level ≥ target, or fill rate ≥ target), bounds on s_i, Q_i.
- Solver settings:
- If nonlinear/with integers: Use Frontline’s GRG Nonlinear or Evolutionary (for integrality/case packs).
- Enable Multistart to avoid local optima.
- For stochastic models, use Simulation Optimization to search s_i, Q_i that minimize expected cost while meeting service levels.

7) Calibrate service levels and trade-offs
- Higher CSL raises SS and holding cost, reduces stockouts. Use scenario analysis: vary CSL (e.g., 90%, 95%, 98%) and compare cost vs. service.
- Consider reducing lead time variability (supplier changes) as an alternative to high safety stock.

8) Execute and validate
- Run Solver to obtain s_i and Q_i (or R and S_i).
- Validate via simulation: check realized fill rate, backorders, cost distribution; adjust constraints if targets aren’t met.
- Implement with monitoring rules (daily/weekly) and exception alerts.

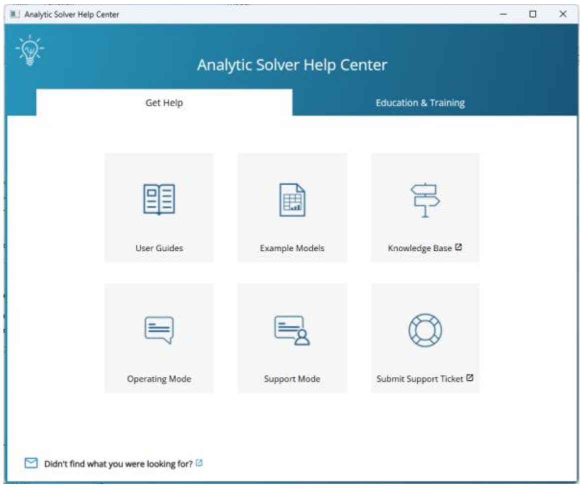
Quick formulas you can apply now (continuous review)
- Q_i = sqrt(2 K_i D_i / h_i)
- σLi = sqrt(L) × σ_i (if demand i.i.d. per period)
- z from CSL target (e.g., 1.28 for 90%, 1.645 for 95%)
- SS_i = z × σLi
- s_i = μ_i × L + SS_i
- Average inventory_i ≈ SS_i + Q_i/2
- Capacity check: sum(space_i × Average inventory_i) ≤ Capacity
- Budget check: sum(c_i × Average inventory_i) ≤ Budget

```
Tips
- Classify SKUs (ABC) to assign differentiated CSL targets and
solver effort (A-items get tighter targets and more frequent
review).
- Impose practical bounds (e.g., 0 ≤ Q_i ≤ 1000; s_i between 0
and a few weeks of demand) to speed convergence.
- If lead time is variable, compute σL using both demand and lead
time variability: σL ≈ sqrt(μ^2 Var(L) + σ^2 E[L]) when
independent.

If you share your demand, lead time, costs, and constraints (or
an Excel layout), I can outline the exact Analytic Solver model
setup, including cell formulas, PSI functions, and Solver
parameter settings.
```

Help Center

The Help Center gives you easy access to video demos, User Guides, online Help, example models, and Website support pages to learn how to use our software tools, and build an effective model.



User Guides

Click the User Guides menu choice to open PDF files of the Analytic Solver Optimization and Simulation User and Reference Guides, Analytic Solver Data Science User or Reference Guides, or our Quick Start Guides.

Example Models

Clicking this menu item will open the Frontline Solvers Example Models Overview dialog with nearly 120 self-guided example models covering a range of model types and business situaltions.

Knowledge Base

Click Knowledge Base to peruse a multitude of online articles related to support and installation issues or to locate articles that will help you to quickly build accurate, efficient optimization, simulation, and data science models.

Operating Mode

Click Operating Mode to switch between three different levels of help. The Excel formulas and functions you use in your model have a huge impact on how fast it runs and how well it solves. If you learn more about this, you can get better results, but if you don't, your results will be limited. Guided Mode can help you learn.

- Guided Mode prompts you step-by-step when solving, with dialogs.
- Auto-Help Mode shows dialogs or Help only when there's a problem or error condition.
- Expert Mode provides only messages in the Task Pane Output tab. (This mode not supported when using a trial license.)

Support Mode

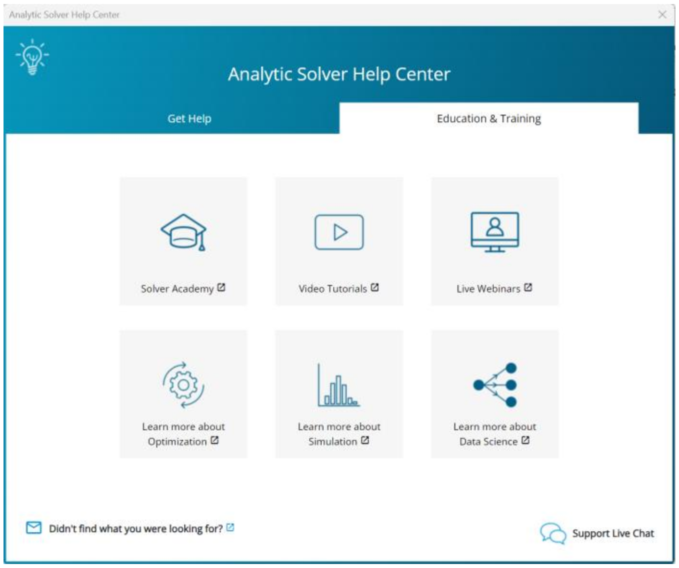
Click Support Mode to switch between three different levels of support. No information (cell contents etc.) from your Excel model is ever reported automatically to Frontline Systems, in any of these Support Modes. Only events in Frontline's software, such as menu selections, Solver Result messages, or error messages are reported.

- **Active Support** automatically reports events, errors and problems to Frontline Support, receives and displays messages to you from Support, and allows you to start a Live Chat with Support while working in Excel (**Recommended**).Auto-Help Mode shows dialogs or Help only when there's a problem or error condition.
- **Standard Support** automatically reports events, errors and problems anonymously (not associated with you) to Frontline Support, but does not provide a means to receive messages or start a Live Chat with Support.
- **Basic Support** provides no automatic connection to Frontline Support. You will have to contact Frontline Support manually via email, website or phone if you need help.

Submit a Support Ticket

If you're having installation, technical, or modeling issues, submit a Support Ticket to open an online support request form. Submit your email address and a short, concise description of the issue that you are experiencing. You'll receive a reply from one of Frontline's highly trained Support Specialists within 24 hours, and generally much sooner.

Our technical support service is designed to supplement your own efforts: Getting you over stumbling blocks, pointing out relevant sections of our User Guides or example models, helping you fix a modeling error, or -- in rare cases -- working around an issue with our software (always at our expense).


Analytic Solver Help Center
Analytic Solver Help Center
Get Help
Education & Training
Solver Academy
Video Tutorials
Live Webinars
Learn more about Optimization
Learn more about Simulation
Learn more about Data Science
Didn't find what you were looking for?
Support Live Chat

Solver Academy

Solver Academy is Frontline Systems' own learning platform. It's the place where business analysts can gain expertise in advanced analytics: forecasting, data science, text mining, mathematical optimization, simulation and risk analysis, and stochastic optimization.

Video Tutorials/Live Webinars

Click Video Tutorials to be directed to Frontline's You Tube Channel. Browse videos on how to create an optimization or simulation model or construct a data science or prediction model using Analytic Solver.

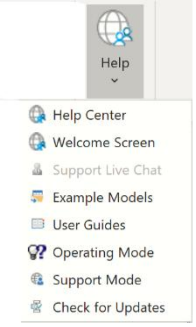
Click Live Webinars to be redirected to www.solver.com to join a live or pre-recorded webinar. Topics include *Using Analytic Solver Data Mining to Gain Insights from your Excel Data*, *Overview of Monte Carlo Simulation Applications*, *Applications of Optimization in Excel*, etc.

Learn more!

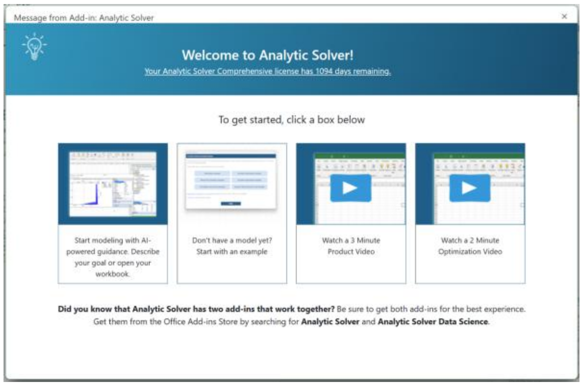
Click any of the three Learn More buttons to learn more about how you can solve large-scale optimization, simulation, and data science models, reduce costs, quantify and mitigate risk, and create forecasting, data science and text science models using Analytic Solver.

Help Menu

Use the Help Menu to gain short cuts to live chat, example models, documentation, set your operating and support mode preferences, and also to open the Welcome Screen and check for software updates.



Use the Welcome Screen to obtain the status of your license or watch a quick video on how to get up and running quickly with Analytic Solver.


Message from Add-in: Analytic Solver

Welcome to Analytic Solver!

Your Analytic Solver Comprehensive license has 1094 days remaining.

To get started, click a box below

Start modeling with AI-powered guidance. Describe your goal or open your workbook.

Don't have a model yet? Start with an example

Watch a 3 Minute Product Video

Watch a 2 Minute Optimization Video

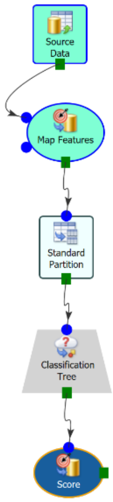
Did you know that Analytic Solver has two add-ins that work together? Be sure to get both add-ins for the best experience. Get them from the Office Add-ins Store by searching for **Analytic Solver** and **Analytic Solver Data Science**.

Creating & Deploying a Workflow

Introduction

The Workflow tab, released in version 2018, allows the combination of all available data science techniques into an all-inclusive workflow, or workflows. Once the workflow is created, either manually or simply by recording your actions, you can initiate the start of the pipeline, or pipelines, by clicking the ▶ button. Note: This is the only tab supported in the Data Science Cloud app.

Analytic Solver Comprehensive allows you to export your existing workflow to Frontline's Rason Cloud Services where it can be deployed to a website without the need to write code! Continue reading to find out how.

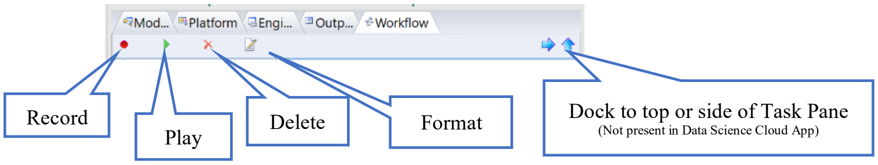


Creating a Workflow

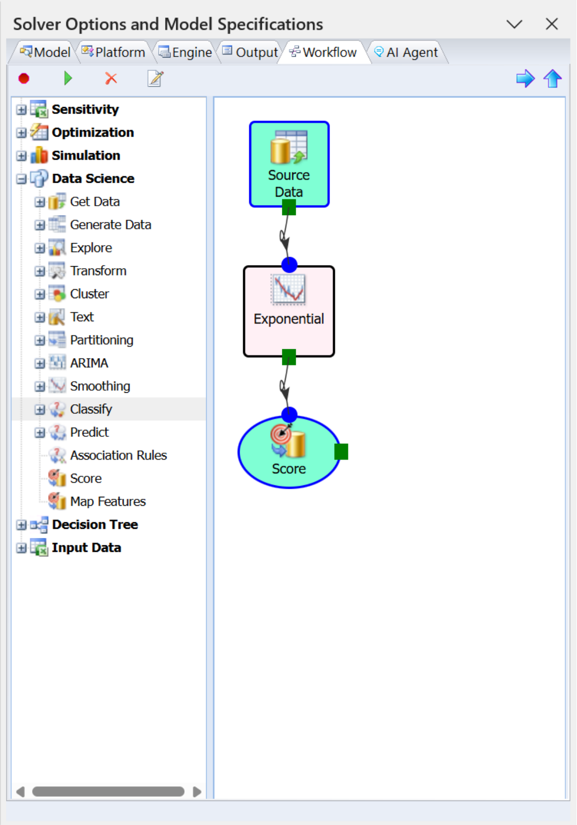
Workflows can be created in two ways: 1. By recording your actions performed on the Data Science ribbon after the Record button is pressed on the Workflow tab or 2. By manually dragging the Data Science nodes from the left of the Workflow tab onto the Workflow window. The next two sections, Recording a Workflow and Manually Creating a Workflow, contain examples to illustrate each method.

Recording a Workflow

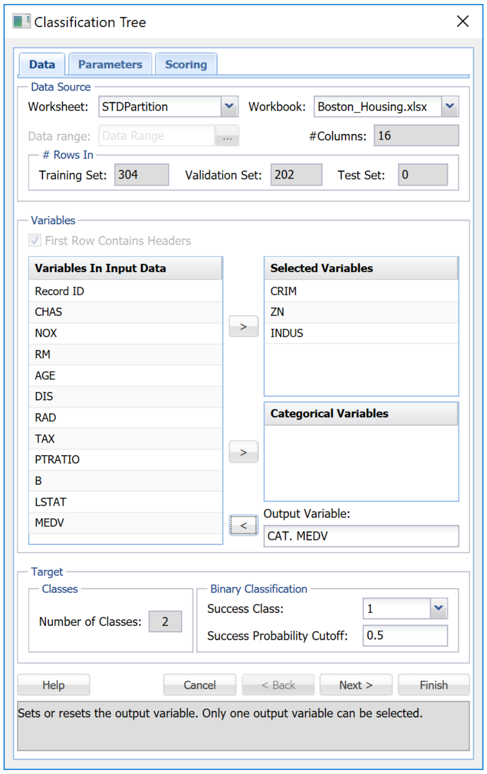
At the top of the Workflow tab, you'll find 6 icons. Each one performs a specific function.


Mod...
Platform
Engi...
Outp...
Workflow
Record
Play
Delete
Format
Dock to top or side of Task Pane
(Not present in Data Science Cloud App)

Press ● to record a new workflow. On the Data Science ribbon, click Help – Example Models – Forecasting/Data Science Examples, then click the Boston_Housing dataset hyperlink to open the Boston Housing example dataset. Click a cell on the Data worksheet, then click ● to start the Workflow recorder. Click Data Science – Partition – Standard Partition to open the Standard Data Partition dialog. While holding down the shift key, select all variables under *Variables In Input Data*, then click > to move them to *Selected Variables*. Leave all options at their defaults, then click OK to run the partition. You'll see several icons added to the Workflow tab.



Click Data Science -- Classify – Classification Tree to open the Classification Tree dialog. Under Data Source, click the down arrow next to Worksheet and select STDPartition. While holding down the SHIFT key, select CRIM, ZN, and INDUS, under *Variables In Input Data*, then click > to move these three variables to *Selected Variables*. Next, select the CAT.MEDV variable, under *Variables In Input Data*, and click > to choose this variable as the *Output Variable*.



Classification Tree

Data | Parameters | Scoring

Data Source

Worksheet: STDPartition
Workbook: Boston_Housing.xlsx

Data range: Data Range
#Columns: 16

Rows In

Training Set: 304
Validation Set: 202
Test Set: 0

Variables

☑ First Row Contains Headers

Variables In Input Data
Record ID
CHAS
NOX
RM
AGE
DIS
RAD
TAX
PTRATIO
B
LSTAT
MEDV

Selected Variables
CRIM
ZN
INDUS

Categorical Variables

Output Variable: CAT. MEDV

Target

Classes

Number of Classes: 2

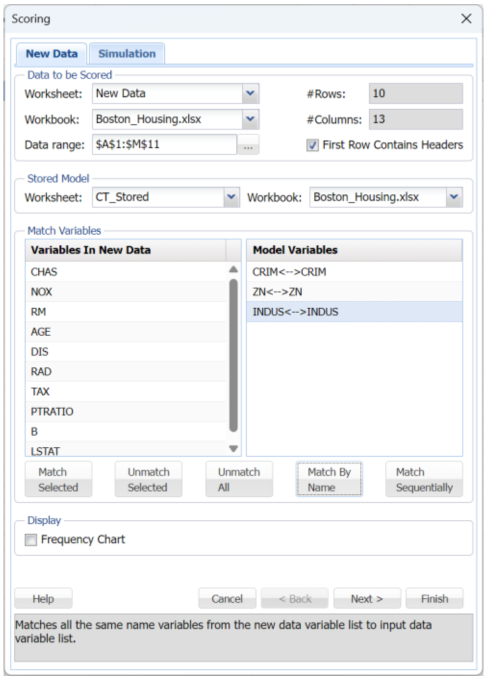
Binary Classification

Success Class: 1
Success Probability Cutoff: 0.5

Sets or resets the output variable. Only one output variable can be selected.

Leave all remaining options at their defaults, then click Finish to run the Classification Tree algorithm.

Finally, click the Score button on the Data Science ribbon to open the Scoring tab. Under *Data to be Scored*, click the down arrow next to Worksheet and select New Data. The *Variables In New Data* field will populate with new variables. Afterwards click *Match By Name* to match the variables listed under *Variables in New Data* with the variables listed under *Model Variables*.



Scoring

New Data | Simulation

Data to be Scored

Worksheet: New Data
#Rows: 10
Workbook: Boston_Housing.xlsx
#Columns: 13
Data range: $A$1:$M$11
☑ First Row Contains Headers

Stored Model

Worksheet: CT_Stored
Workbook: Boston_Housing.xlsx

Match Variables

Variables In New Data
CHAS
NOX
RM
AGE
DIS
RAD
TAX
PTRATIO
B
LSTAT

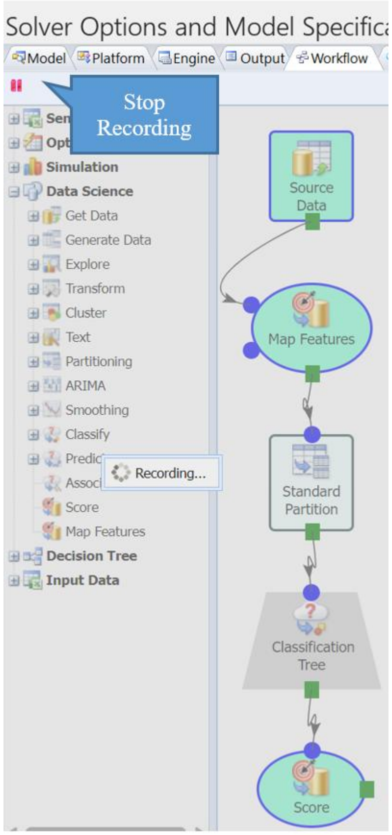
Model Variables
CRIM<-->CRIM
ZN<-->ZN
INDUS<-->INDUS

Display

☐ Frequency Chart

Matches all the same name variables from the new data variable list to input data variable list.

Click *Finish* to score the new data then click the Stop Recording button to stop the Workflow recorder.



The completed workflow is displayed in the window. Use Excel's File – Save to save the workflow to the workbook.

There is no limit to the number of workflows that may appear in the Workflow tab. Each new workflow will be added to the right of the existing workflow(s).

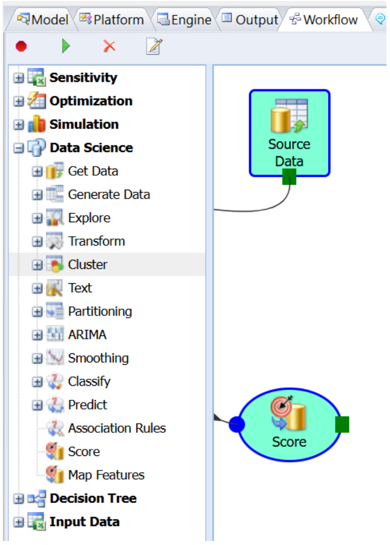
Workflow Tab Options

Press ▶ **to execute the workflow or workflows**. If no nodes in the workflow are selected, all workflows present in the Workflow window will be executed. If a node in the flow is selected, and the Execute Workflow button is pressed, the workflow will be run up to and including the selected node.

Press ✕ **to delete the contents of the Workflow window.** If no nodes in the workflow are selected, all nodes in the Workflow window will be deleted. If a node is selected in the Workflow window, pressing the Delete Workflow icon will delete only the selected node.

Press 📝 **to format the Workflow** into a more readable form. If the Task Pane is docked to the top of the Excel workbook, the Workflow will be formatted horizontally. If the Task Pane is docked to the side of the Excel workbook, the Workflow will be formatted vertically.

Note: It is possible to move a node "behind" the task pane as shown in the screenshot below.



To regain access to the node, click the Format Workflow button.

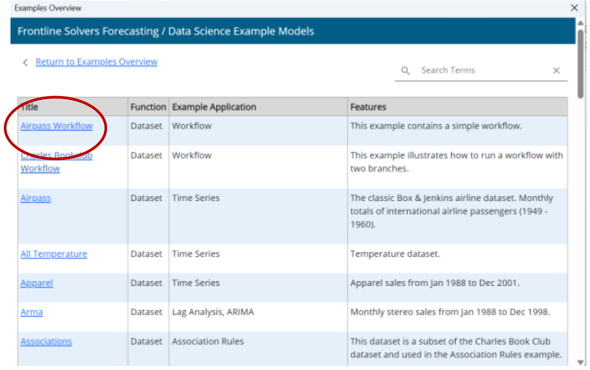
Press → to dock the Solver Task Pane to the right of the workbook. Press ↑ to dock the Solver Task Pane to the top of the workbook.

Deploying Your Workflow

Starting with Analytic Solver Comprehensive V2020, when you click Deploy Model – Cloud Service, your model (or a copy of it) is saved to the Azure cloud. (Note: Analytic Solver Data Science does not support this new feature. Trial licenses of Analytic Solver Comprehensive *do* support this new feature.)

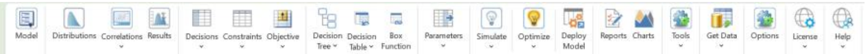
If you use File Save to store your workbook in OneDrive or OneDrive for Business, we use that, which is by far the best option. Afterwards, any web or mobile application can easily run your model, via simple, standard REST API calls, often made from JavaScript on a web page, of from C# of Java code on a Web server.

Click Help – Example Models – Forecasting / Data Science Examples, then click the Airpass Workflow link. The Airpass Workflow Example will open.


Examples Overview
Frontline Solvers Forecasting / Data Science Example Models
Return to Examples Overview
Search Terms

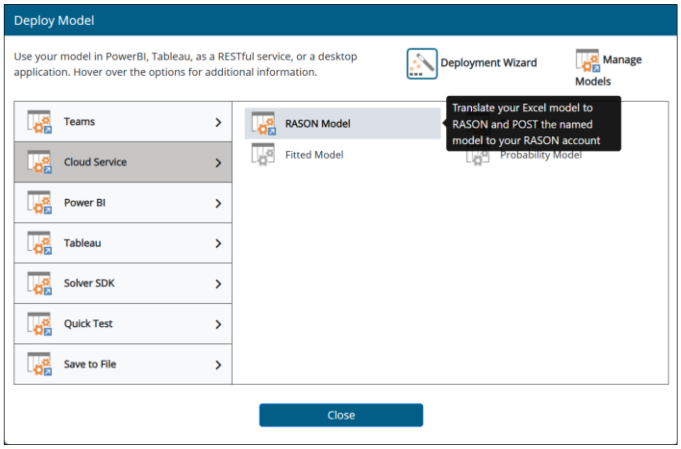
Title	Function	Example Application	Features
Airpass Workflow	Dataset	Workflow	This example contains a simple workflow.
Charles Bookclub Workflow	Dataset	Workflow	This example illustrates how to run a workflow with two branches.
Airpass	Dataset	Time Series	The classic Box & Jenkins airline dataset. Monthly totals of international airline passengers (1949 - 1960).
All Temperature	Dataset	Time Series	Temperature dataset.
Apparel	Dataset	Time Series	Apparel sales from Jan 1988 to Dec 2001.
Arma	Dataset	Lag Analysis, ARIMA	Monthly stereo sales from Jan 1988 to Dec 1998.
Associations	Dataset	Association Rules	This dataset is a subset of the Charles Book Club dataset and used in the Association Rules example.

Click the Analytic Solver tab to move to the Analytic Solver ribbon.



Posting Workflow to RASON Cloud Services

Click Deploy Model– Cloud Service – RASON Model.



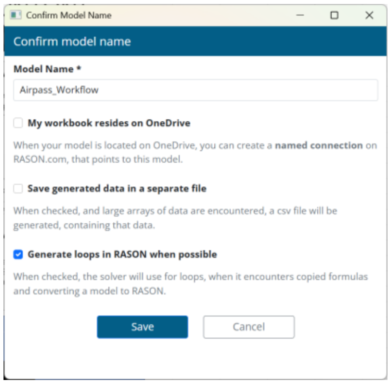
Use your model in PowerBI, Tableau, as a RESTful service, or a desktop application. Hover over the options for additional information.

Translate your Excel model to RASON and POST the named model to your RASON account

In the COM Addin, it's easy for us to check if the user has an optimization, simulation, or flow model, and we can give that proper warning/error message.

Note: *For a multi-stage data science flow, Cloud service – RASON model is the only available Deploy Model options in either Analytic Solver Com Addin or Analytic Solver Cloud app.* Users of the Analytic Solver COM Addin will receive an error if they attempt to use Power BI, Tableau, etc. However, users of the Analytic Solver Cloud app will not.

On the next screen, click Save to accept the default Model Name. (You can also provide a more meaningful name here if wanted. If your workbook resides on your OneDrive account, click the checkbox.)



Model Name *

Airpass_Workflow

☐ My workbook resides on OneDrive

When your model is located on OneDrive, you can create a **named connection** on RASON.com, that points to this model.

☐ Save generated data in a separate file

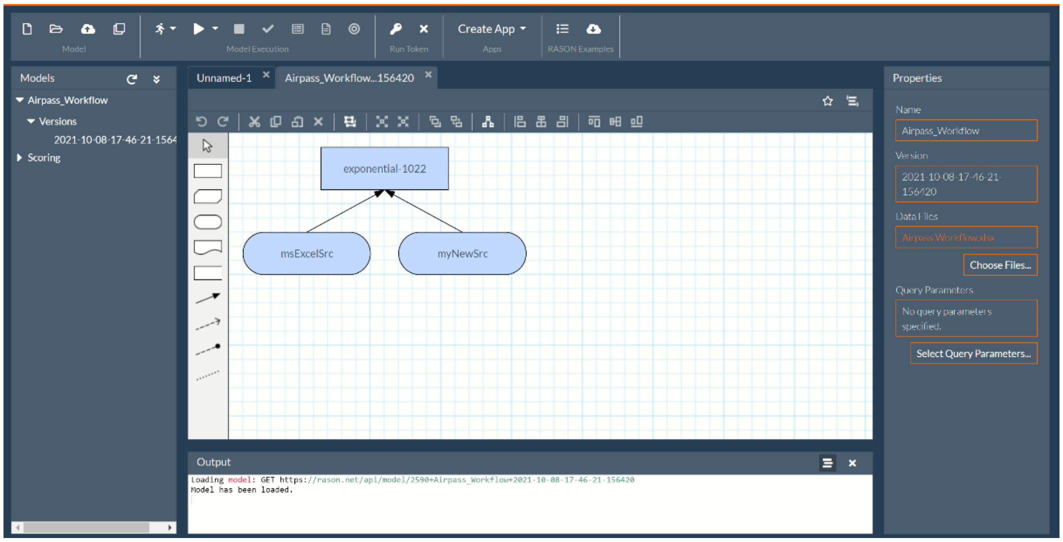
When checked, and large arrays of data are encountered, a csv file will be generated, containing that data.

☑ Generate loops in RASON when possible

When checked, the solver will use for loops, when it encounters copied formulas and converting a model to RASON.

Note that data science models that include a weight variable or a partition variable may not be translated to Rason.

Immediately, a browser opens to the Editor tab on www.Rason.com. From here you can view your workflow translated into Frontline's Rason modeling language. Notice that the Airpass_Workflow is listed under Models on the right and the workflow appears in graphical form in the grid.



Name: Airpass_Workflow

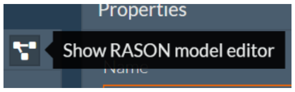
Version: 2021-10-08-17-46-21-156420

Data Files: Airpass Workflow.xlsx

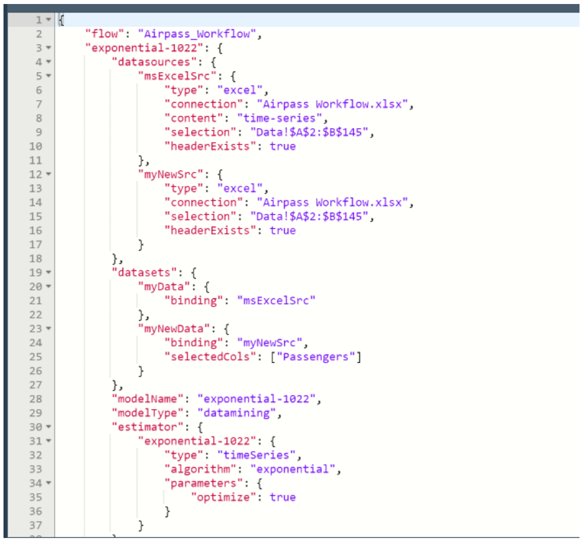
Query Parameters: No query parameters specified.

```
Loading model: GET https://rason.net/api/model/2590+Airpass_Workflow+2021-10-08-17-46-21-156420
Model has been loaded.
```

To see the model translated into Rason simply click the Show RASON model editor icon in the top left of the grid.



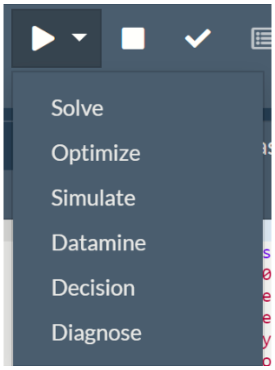
The Rason model editor displays the Airpass Workflow example translated into the Rason modeling language.



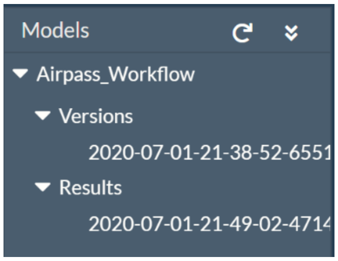
```
{
    "flow": "Airpass_Workflow",
    "exponential-1022": {
        "datasources": {
            "msExcelSrc": {
                "type": "excel",
                "connection": "Airpass Workflow.xlsx",
                "content": "time-series",
                "selection": "Data!$A$2:$B$145",
                "headerExists": true
            },
            "myNewSrc": {
                "type": "excel",
                "connection": "Airpass Workflow.xlsx",
                "selection": "Data!$A$2:$B$145",
                "headerExists": true
            }
        },
        "datasets": {
            "myData": {
                "binding": "msExcelSrc"
            },
            "myNewData": {
                "binding": "myNewSrc",
                "selectedCols": ["Passengers"]
            }
        },
        "modelName": "exponential-1022",
        "modelType": "datamining",
        "estimator": {
            "exponential-1022": {
                "type": "timeSeries",
                "algorithm": "exponential",
                "parameters": {
                    "optimize": true
                }
            }
```

From here you can edit your model directly in Rason or if you'd rather not interact with Rason code at all, simply make the change to the Excel model and then re-deploy the edited workflow. There's no need to learn a new language!

To solve this model on the Editor tab, simply click the down arrow next to the "play" button and click Solve. (Only the Solve endpoint supports the solving of workflows.)

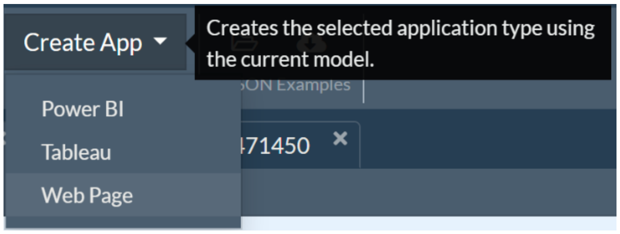


After the workflow is executed, you can find the results listed under Results.

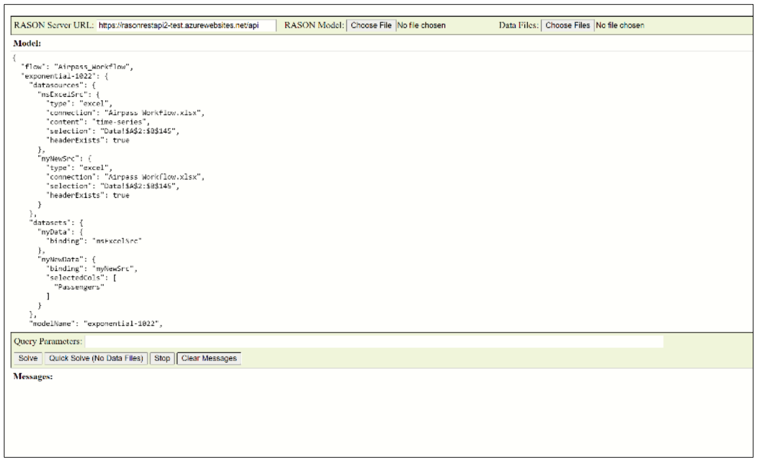


Deploying Your Model

Click Create App – Web Page on the Editor tab ribbon to deploy the Airpass Workflow to a Web page application.


Creates the selected application type using the current model.

The file, RasonScript.html, will be downloaded locally. Double click the file to open the Web application. Notice that the translated workflow appears in the Model window.


RASON Server URL: https://rasonrestapi2-test.azurewebsites.net/api RASON Model: Choose File No file chosen Data Files: Choose Files No file chosen

Model:

```
{
  "flow": "Airpass_Workflow",
  "exponential-1022": {
    "datasources": {
      "msExcelSrc": {
        "type": "excel",
        "connection": "Airpass Workflow.xlsx",
        "content": "time-series",
        "selection": "Data!$A$2:$B$145",
        "headerExists": true
      },
      "myNewSrc": {
        "type": "excel",
        "connection": "Airpass Workflow.xlsx",
        "selection": "Data!$A$2:$B$145",
        "headerExists": true
      }
    },
    "datasets": {
      "myData": {
        "binding": "msExcelSrc"
      },
      "myNewData": {
        "binding": "myNewSrc",
        "selectedCols": [
          "Passengers"
        ]
      }
    },
    "modelName": "exponential-1022",
```

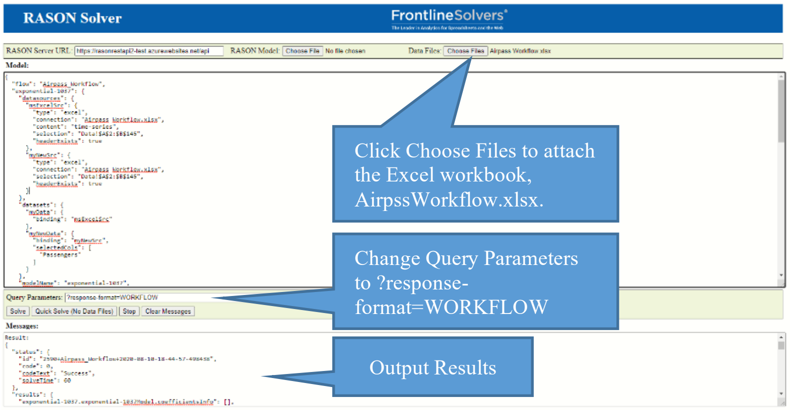
Query Parameters:

Messages:

To solve the workflow:

1. Attach the Excel file, Airpass Workflow.xlsx. Click "Choose Files" at the top of the Web app, then browse to C:\ProgramData\Frontline Systems\Datasets and select Airpass Workflow.xlsx.
2. Since we are solving a workflow, we must first change the query parameter from ?response-format=STANDALONE to response-format=WORKFLOW.
3. Now click the Solve button to solve the workflow through the Web application.

Note: Quick Solve does not support external files so it is unable to solve this translated workflow.


RASON Solver
FrontlineSolvers®
The Leader in Analytics for Spreadsheets and the Web
Choose File
Choose Files
Solve
Quick Solve (No Data Files)
Stop
Clear Messages

RASON Server URL: https://rasonrestapi2-test.azurewebsites.net/api

RASON Model: No file chosen

Data Files: Airpass Workflow.xlsx

Model:

```
{
  "flow": "Airpass_Workflow",
  "exponential-1037": {
    "datasources": {
      "msExcelSrc": {
        "type": "excel",
        "connection": "Airpass Workflow.xlsx",
        "content": "time-series",
        "selection": "Data!$A$2:$B$145",
        "headerExists": true
      },
      "myNewSrc": {
        "type": "excel",
        "connection": "Airpass Workflow.xlsx",
        "selection": "Data!$A$2:$B$145",
        "headerExists": true
      }
    },
    "datasets": {
      "myData": {
        "binding": "msExcelSrc"
      },
      "myNewData": {
        "binding": "myNewSrc",
        "selectedCols": [
          "Passengers"
        ]
      }
    },
    "modelName": "exponential-1037",
```

Query Parameters: ?response-format=WORKFLOW

Messages:

```
Result:
{
  "status": {
    "id": "2590+Airpass_Workflow+2020-08-10-18-44-57-498458",
    "code": 0,
    "codeText": "Success",
    "solveTime": 60
  },
  "results": {
    "exponential-1037.exponential-1037Model.coefficientsInfo": [],
```

Click Choose Files to attach the Excel workbook, AirpssWorkflow.xlsx.

Change Query Parameters to ?response-format=WORKFLOW

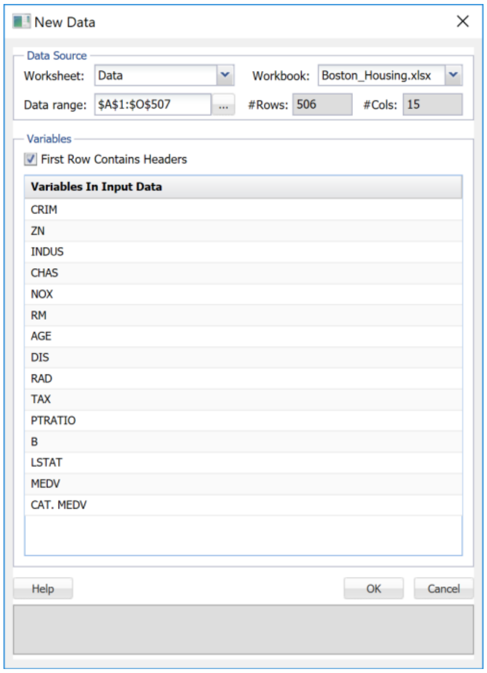
Output Results

Whether you decide to keep your model in Excel or translate it to RASON, the *important thing* is that your workflow (or model) can be executed **whenever or wherever** it's needed – on the factory floor, on a salesperson's laptop or smartphone or in call center custom application. And Rason can get the updated data your model needs directly from operational business systems. This is amazingly easy if you are using Power BI, Power Apps, Power Automate or Dynamics 365.

For more information on this new feature, including information on the Deployment Wizard and how to Manage your deployed models within Excel, see the Analytic Solver User Guide chapter, Deploying Your Model.

Manually Creating a Workflow

It's also possible to manually create a workflow or workflows. Go back to Excel, open 'Boston Housing.xlsx' workbook and navigate to 'Workflow' tab. Click the + in front of Get Data, then drag "New" into the workflow window. The New Data dialog will open. Under Data Source, click the down arrow next to Worksheet and select Data.



Data Source

Worksheet: Data

Workbook: Boston_Housing.xlsx

Data range: $A$1:$O$507

#Rows: 506

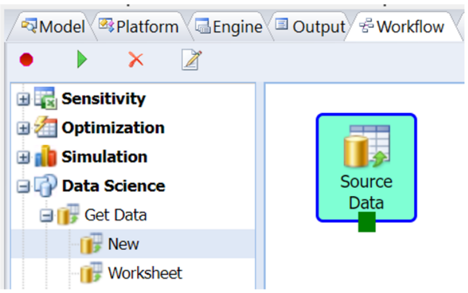
#Cols: 15

Variables

☑ First Row Contains Headers

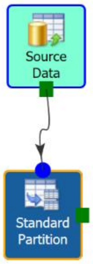
Variables In Input Data
CRIM
ZN
INDUS
CHAS
NOX
RM
AGE
DIS
RAD
TAX
PTRATIO
B
LSTAT
MEDV
CAT. MEDV

Click OK to close the dialog. The Source Data icon will appear in the Workflow window. It's important to note that a workflow does not need to start with a Source Data node. In fact, any node dragged to the workflow window can stand alone and will be executed when the Execute Solver button is pressed.



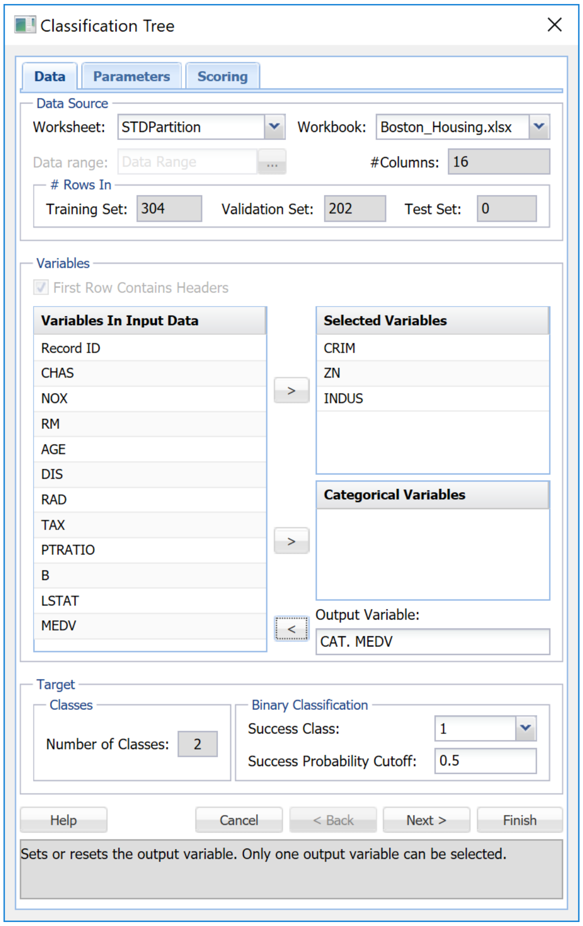
Afterwards, expand Partitioning (by clicking the +), then drag Standard Partition to the Workflow window. When the Standard Partition dialog opens, select all variables under *Variables In Input Data* then click > to move all variables to *Selected Variables*. Leave all options at their defaults and click OK. Immediately, the workflow updates in the Workflow window displaying the connection.

Note: Notice that when creating a workflow manually, Map Features is not required. However, if you'd like to provide a new data source for an existing flow, you must use Map Features to specify the input and output variables to be used in the flow. See below for more information.



Expand Classify, then drag Classification Tree to the window.

- Under Data Source, click the down arrow next to Worksheet and select STDPartition.
- Select the first three variables under *Variables In Input Data* (CRIM, ZN and INDUS) and click > to move them to *Selected Variables*.
- Select CAT.MEDV as the *Output Variable*.



Data Source

Worksheet: STDPartition

Workbook: Boston_Housing.xlsx

Data range: Data Range

#Columns: 16

Rows In

Training Set: 304

Validation Set: 202

Test Set: 0

Variables

☑ First Row Contains Headers

Variables In Input Data
Record ID
CHAS
NOX
RM
AGE
DIS
RAD
TAX
PTRATIO
B
LSTAT
MEDV

Selected Variables
CRIM
ZN
INDUS

Categorical Variables

Output Variable: CAT. MEDV

Target

Classes

Number of Classes: 2

Binary Classification

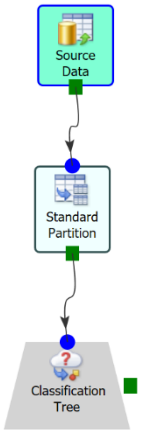
Success Class: 1

Success Probability Cutoff: 0.5

Sets or resets the output variable. Only one output variable can be selected.

Then click Finish. For more information on the Classification Tree feature in Analytic Solver Data Science, see the Data Science Reference Guide.

The workflow will update immediately by connecting the Standard Partition icon to the Classification Tree icon.

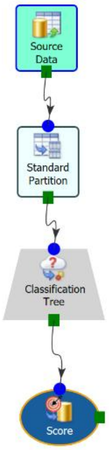


Note: Automatic connections are made based on the current effective state of the workflow and sheets in the workbook. For example, if you were to run this workflow, a new STDPartition1 would be created. As a result, connecting a new classification or regression to the STDPartition sheet would not be allowed.

On the Workflow tab, drag Score into the Workflow window. The Scoring dialog will open. Click the down arrow next to Worksheet under *Data to be Scored* and select *New Data*. Then click the down arrow for *Worksheet* under *Stored Model* and select CT_Stored. Click *Match By Name* to match the variables in the New Data worksheet with the selected Model Variables, then click OK.

Note: When a node is dragged onto the workspace, the node's dialog opens. When Finish or OK is pressed on the open dialog, the node is executed. If this dialog is reopened and either minor or no changes are made, the node will not be executed again when Finish or OK is clicked to close the dialog. However, if major changes are made, and children (additional nodes) are attached to this node, the node will be executed again.

Immediately, the workflow window is updated.



Once all nodes are connected, click ▶ to run the workflow.

Saving a Workflow

A workflow is automatically saved to the workbook when one of three events is executed.

1. When an Analytic Solver Data Science dialog is closed.

2. After the workflow is formatted by pressing the Format Workflow button on the Workflow tab.

3. After the Stop Recording button is pressed on the Workflow tab.

Otherwise, use Excel's File – Save menu item to save the workflow to the workbook.

Note on Sampling/Summarizing Big Data

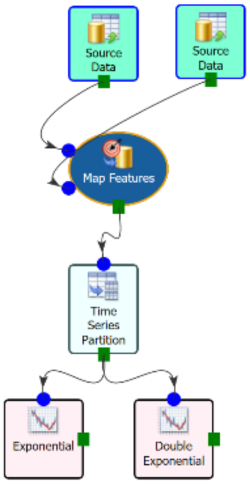
Currently, workflow methods that perform asynchronous activities, such as Sampling and Summarizing Big Data, will first be converted to synchronous methods when included in a workflow. Running big data synchronously (i.e. clicking the RUN button on the Sampling or Summarizing Big Data dialog) is supported without any required internal conversion.

Making/Breaking a Connection

To make a connection between two existing nodes, simply connect the green square on the first node with the blue circle on the second node. To break a connection, simply click the arrow connection in the blue node of the second node and move it back to the green square on the first node.

Running a Workflow with a New Dataset

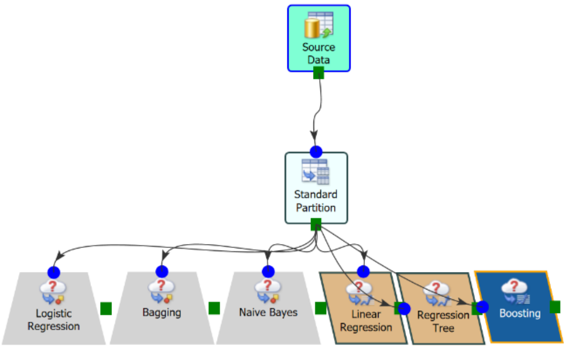
When there is an existing workflow, you may drag and drop a new dataset (a new Get Data node) onto the Workflow window and connect the new Source Data icon to the existing "Map Features" node. The result is two connections, one from each Source Data icon, to the Map Features icon. Connecting to, or double-clicking the "Map Features" node will pop up the existing Score button dialog, enabling column names to be matched. When the workflow is executed, the newer data source will be used.



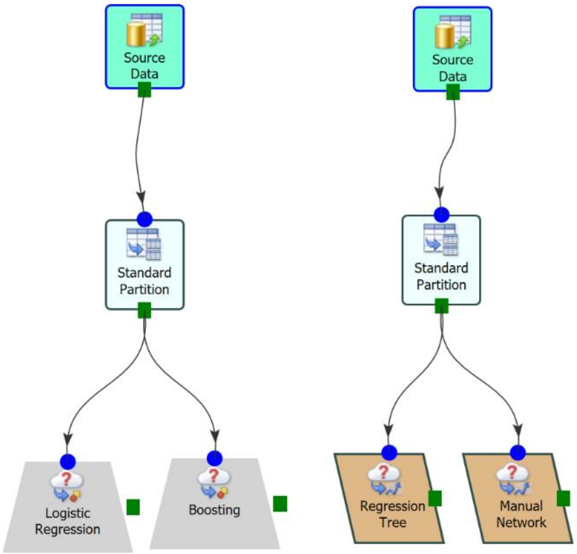
Multiple Workflows

As discussed earlier, there is no limit to the number of workflows appearing in the workflow window. To run several different classification or regression algorithms at the same time, simply drag the desired method into the workflow and connect the method to the flow.

In the example workflow below, three different classification techniques, Logistic Regression, Bagging and Naïve Bayes, and three different regression methods, Linear Regression, Regression Tree and Boosting, use the same standard partitions in the workflow. When this flow is executed, all six models will be created and used to score new data. Note: Although each classification and regression technique uses the same standard partition and data source, each model may contain different selected variables.



Two workflows are present below. Once the Execute Workflow button is pressed, all workflows will be executed at the same time.

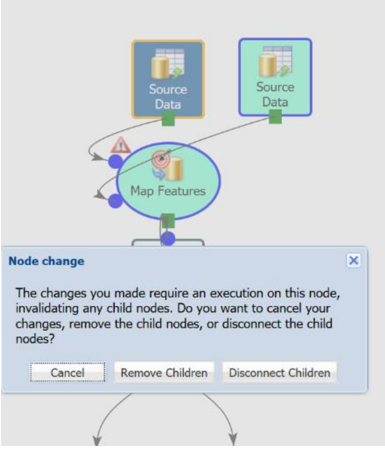


Changing Options Settings

To change the option settings for any of the nodes in the workflow, simply double click the desired node to bring up the task dialog, make the desired option changes, then click OK or Finish.

Important Note: If you make changes to the Selected Variables or the Input Sheet for any existing node, a message with three options will appear asking if you would like to 1. Cancel, 2. Remove Children or 3. Disconnect Children.

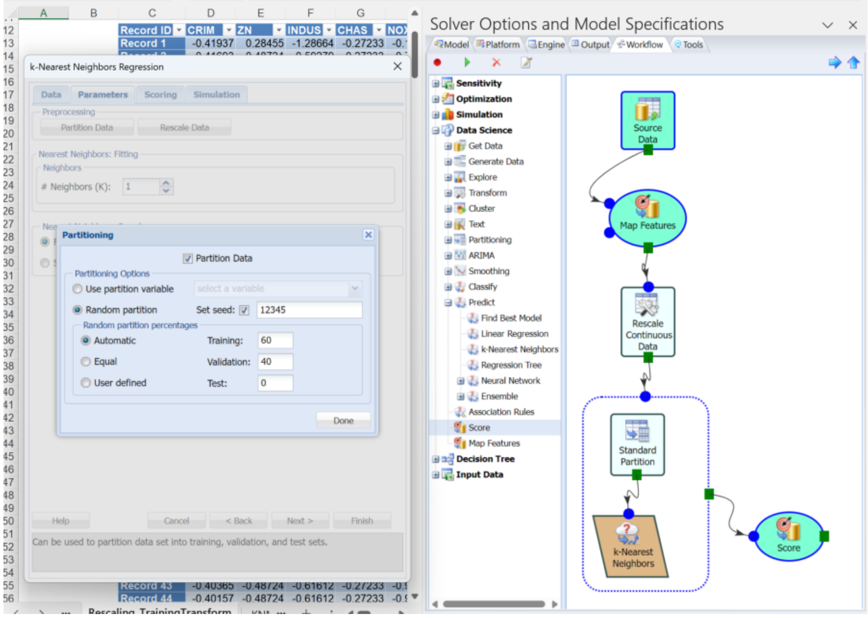
Click Cancel to return to the Workflow tab without any changes being made. Click Remove Children to delete all nodes beneath the selected node or Disconnect Children to disconnect the selected node from the rest of the workflow.



The changes you made require an execution on this node, invalidating any child nodes. Do you want to cancel your changes, remove the child nodes, or disconnect the child nodes?

Workflow Groups

Analytic Solver Data Science allows you to partition, rescale and score a dataset "on-the-fly". This means that instead of performing separate steps for partitioning, rescaling or scoring, users can perform all these actions on the same set of dialogs used to create the classification or regression model. When partitioning, rescaling or scoring "on the fly" during the recording of a workflow, a "group" will be created. This group is treated as one node and is denoted with a dotted line surrounding all nodes in the group. Clicking any of the nodes in the group will bring up the classification or regression method dialogs.



k-Nearest Neighbors Regression

Neighbors (K): 1

Partitioning

☑ Partition Data

○ Use partition variable — select a variable

● Random partition — Set seed: ☑ 12345

● Automatic — Training: 60

○ Equal — Validation: 40

○ User defined — Test: 0

Can be used to partition data set into training, validation, and test sets.

To create a group when creating a workflow manually, simply select the partitioning, rescaling or scoring options once the classification or regression method dialog opens.

To remove or change a group, double click any of the nodes in the group to open the classification or regression method dialog and disable the desired feature or features.

Deploying a Fitted Model to RASON, Power BI or Tableau

Introduction

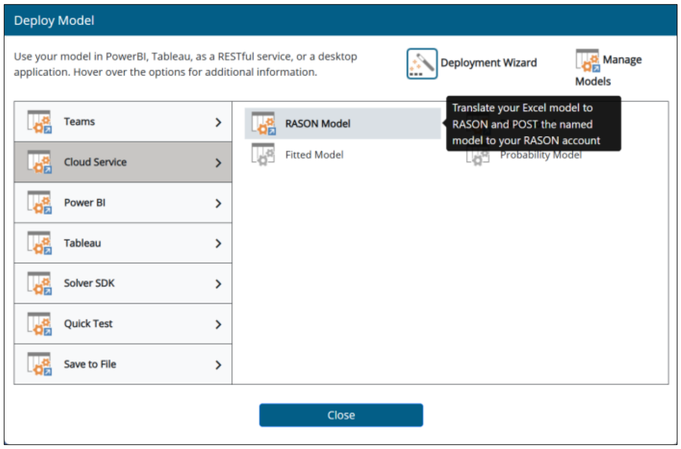
When using Analytic Solver you have the ability to not only create, design and solve your optimization, simulation, simulation optimization, stochastic optimization and data science/forecasting models in Microsoft Excel, but you also have the ability to embed your model into your own application and solve it on the Web or within Power BI or Tableau by calling our RASON Server.

RASON (which stands for Restful Analytic Solver Object Notation) is Frontline Systems' modeling language embedded in JSON and a REST API that makes it easy to create, test and deploy analytic models using optimization, simulation and data science, in web and mobile applications.

Users of Frontline's Excel Solvers will find that 1. it's exceedingly easy to translate Excel models into RASON models 2. that your knowledge of Excel formulas and functions is immediately usable and 3. RASON models can be more flexibly "bound" to data from a variety of sources.

When you click the RASON Deployment icon located on the Analytic Solver ribbon, you'll notice that your optimization, simulation or forecasting/data science model can be deployed to the Cloud, Power BI, Tableau, or your own application on the Web.

Note: In order to use the RASON Deployment functionality, users of Analytic Solver Data Science must also have a license for Analytic Solver Upgrade, Decision, Optimization, Simulation or Comprehensive. For more information on fitted models, please see the Scoring chapter within this guide.



Deploy Model

Use your model in PowerBI, Tableau, as a RESTful service, or a desktop application. Hover over the options for additional information.

Translate your Excel model to RASON and POST the named model to your RASON account

Web App Developers will be able to immediately find how exceptionally easy it is to embed RASON models as JSON and solve them using Frontline's RASON server, which exposes a simple REST API that's scalable to handle very large, compute-intensive analytic models. Months of work, that would have previously been required, have been reduced to a single command button click!

RASON Deployment Menu

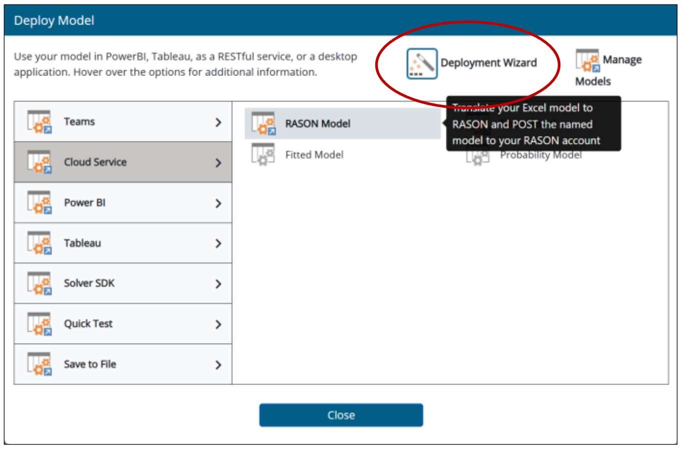
The purpose of RASON Deployment is to enable your model to be used wherever/whenever it is needed – on the factory floor, on a salesperson's laptop or smart phone or in a call center custom application. On your own, without web development or IT help, you can get your model working as a cloud service. Once you've shown that it works, your developers or IT will find they can everything they'd like – customize code, connect to operational databases, apply security and governance best practices.

Rason makes your model accessible 24 x 7 via simply REST API and JSON responses – the common standard for web and mobile apps. Even better, it makes your model results available via OData – the standard widely supported by Microsoft apps including Power Platform. And Rason has built-in facilities to get the updated data your model needs directly from operational business systems. This is amazingly easy if you are using Power BI, Power Apps, Power Automate, or Dynamics 365.

Rason contains the entire Excel formula language as a subset, including virtually all of Excel's built-in functions. That makes translation possible and it allows you to see your Excel formulas embedded in Rason's syntax. To work well in web and mobile apps, Rason is embedded in JSON (JavaScript Object Notation) which allows it to work directly in Power BI, Tableau or a page on any website.

With RASON, you can create more flexible models, by working with Rason arrays and tables instead of fixed-size cell ranges. When products, regions, time periods and other "dimensions" change, your model can automatically adjust and keep running – without any extra work on your part.

When you first click the RASON Deployment icon on the Analytic Solver ribbon, the new Rason Deployment dialog appears.



Deploy Model

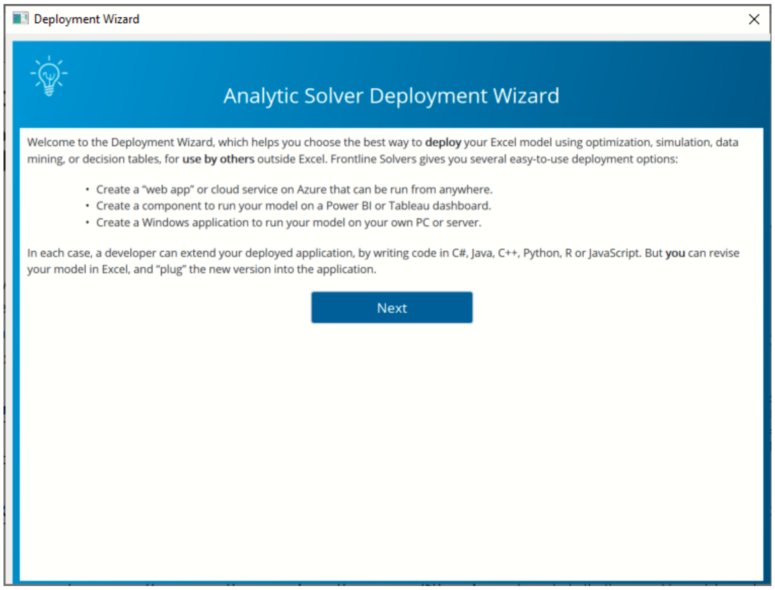
Use your model in PowerBI, Tableau, as a RESTful service, or a desktop application. Hover over the options for additional information.

Translate your Excel model to RASON and POST the named model to your RASON account

Follow the steps below to convert an optimization, simulation or data science model into a RASON model and solve it in the Cloud, Tableau, Power BI, with Solver SDK or on an automatically created web page.

Deployment Wizard

If you just aren't sure where to start or what you need, use the handy deployment wizard to help guide you through the RASON Deployment menu options.

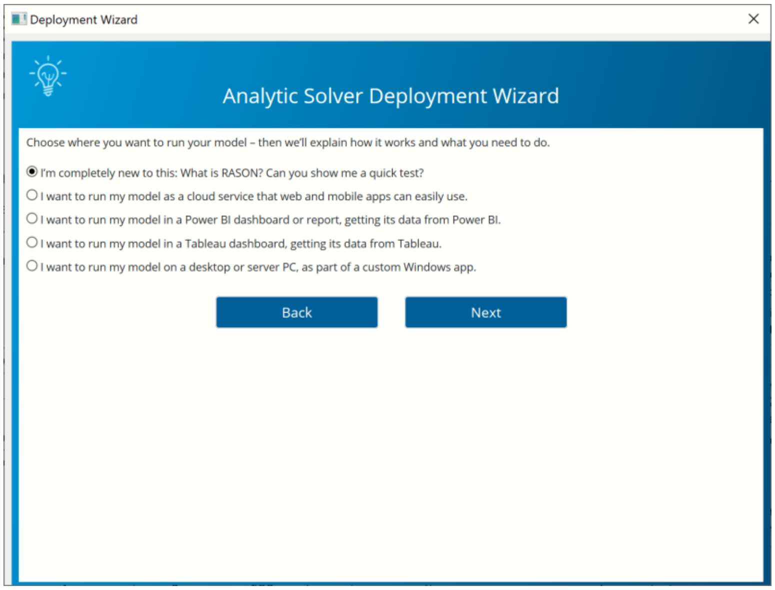


Welcome to the Deployment Wizard, which helps you choose the best way to **deploy** your Excel model using optimization, simulation, data mining, or decision tables, for **use by others** outside Excel. Frontline Solvers gives you several easy-to-use deployment options:

- Create a "web app" or cloud service on Azure that can be run from anywhere.
- Create a component to run your model on a Power BI or Tableau dashboard.
- Create a Windows application to run your model on your own PC or server.

In each case, a developer can extend your deployed application, by writing code in C#, Java, C++, Python, R or JavaScript. But **you** can revise your model in Excel, and "plug" the new version into the application.

When you click "Next" you'll be guided to select the best way to deploy your Excel model. In each case, you or a developer can extend your deployed application by writing C#, JAVA, C++, Python, R, JavaScript, or you can revise your model in Excel and "plug" the new version into the application.



Choose where you want to run your model – then we'll explain how it works and what you need to do.

● I'm completely new to this: What is RASON? Can you show me a quick test?
○ I want to run my model as a cloud service that web and mobile apps can easily use.
○ I want to run my model in a Power BI dashboard or report, getting its data from Power BI.
○ I want to run my model in a Tableau dashboard, getting its data from Tableau.
○ I want to run my model on a desktop or server PC, as part of a custom Windows app.

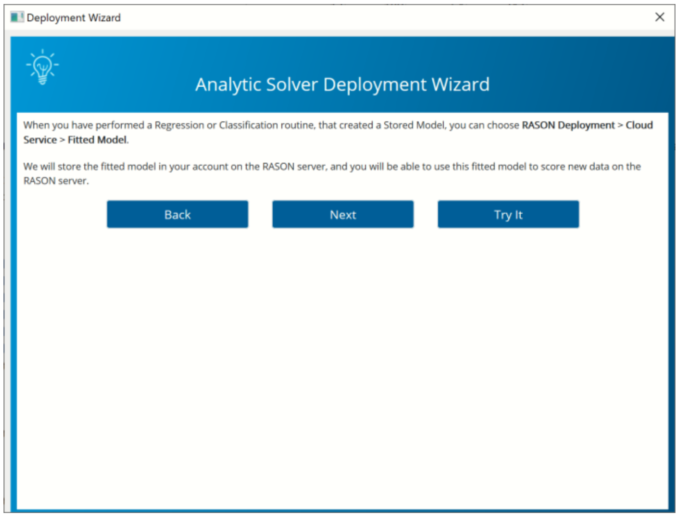
RASON Deployment Menu

Clicking the second option, "I want to run my model as a cloud service…" gives you four options, RASON Model, Excel Model, Fitted Model and Probability Model.

- RASON Model translates your optimization, simulation, data science and decision table model into Rason and POSTs the model to your RASON account.
- Excel Model POSTs the Excel workbook model to the RASON account.
- Fitted Model POSTs the data science or forecasting fitted model saved to an Excel workbook to the RASON Server.
- Probability Model allows the deploying and sharing of probability models, following the open Probability Management 3.0 standard. See the Analytic Solver User Guide for more information.

From here you can maintain and solve your model and get the results on www.RASON.com.

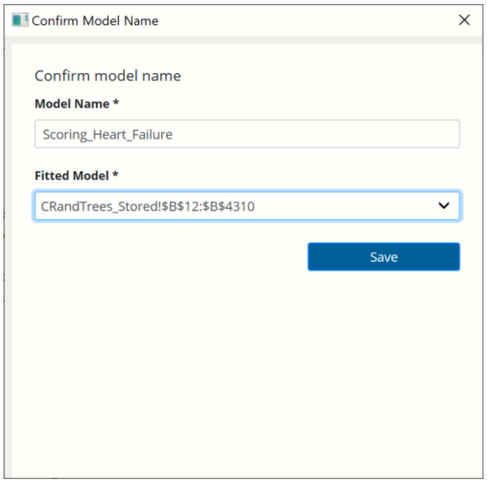
Clicking "Next" a few times brings you to the Fitted Model deployment page.



When you have performed a Regression or Classification routine, that created a Stored Model, you can choose **RASON Deployment > Cloud Service > Fitted Model**.

We will store the fitted model in your account on the RASON server, and you will be able to use this fitted model to score new data on the RASON server.

Click "Try It" to deploy your data science/forecasting fitted model to a Web application. If a fitted model is contained in your workbook, the following dialog will appear. Here you can give the RASON model a name and also select the range of the fitted model.



Confirm model name

Model Name *

Scoring_Heart_Failure

Fitted Model *

CRandTrees_Stored!$B$12:$B$4310

Clicking the third option, "I want to run my model in a Power BI dashboard…", gives you the ability to turn your Excel-based optimization, simulation or data science fitted model into a **Microsoft Power BI Custom Visual**. First, you simply select rows or columns of data to serve as changeable parameters, next, you select **RASON Deployment – Power BI – Managed Model/Embedded Model**, and thirdly, you save the file created by Analytic Solver. Afterwards, you click the Load Custom Visual icon in Power BI and select the file you just saved. What you get isn't just a chart – it's your *full optimization, simulation or data science fitted model*, ready to accept Power BI data**, run on demand** on the web, and display visual results in Power BI! You simply need to drag and drop appropriate Power BI datasets into the "well" of inputs to match your model parameters. This is possible because Analytic Solver translates your Excel model into **RASON**® then "wraps" a JavaScript-based Custom Visual around the RASON model. For more information on this feature, see the next chapter "Creating Power BI Custom Visuals".

Clicking the fourth option, "I want to run my model in a Tableau dashboard…", allows you to turn your Excel-based optimization, simulation or data science fitted model into a **Tableau Dashboard Extension**. You simply select rows or columns of data to serve as changeable parameters, then choose **RASON Deployment – Tableau – Managed Model/Embedded Model**, and save the file created by Analytic Solver. In Tableau, you'll see the newly created file under **Extensions** on the left side of the dashboard, where you can drag it onto your dashboard. You'll be prompted to match the parameters your model needs, with data in Tableau. Much like with Power BI, what you get isn't just a chart – it's your *full optimization, simulation or data science fitted model*, ready to accept Tableau data**, run on demand** (using our **RASON** server), and display visual results in Tableau! Note: This feature works (only) with Tableau version 2018.2 or later. For more information on this feature, see the chapter "Creating Custom Extensions in Tableau".

Continue reading for a step-by-step example that illustrates how to use each of these three RASON Deployment functionality.

Conversion Exceptions

There are several types of models that Frontline Excel Solvers are not able to convert to RASON models, i.e. Excel models containing errors such as #NUM, #VALUE, etc. For all unsupported features, the conversion generator returns an appropriate message.

Note: When converting a simulation model into the RASON modeling language note that only statistics with explicit numeric arguments such as =PsiPercentile(cell, 0.50) or =PsiPercentile(cell, A1) where A1 = 0.05 are supported. Expressions passed to arguments such as =PsiPercentile(cell, A1+A2), where A1 = .1 and A2 = .3 are not supported.

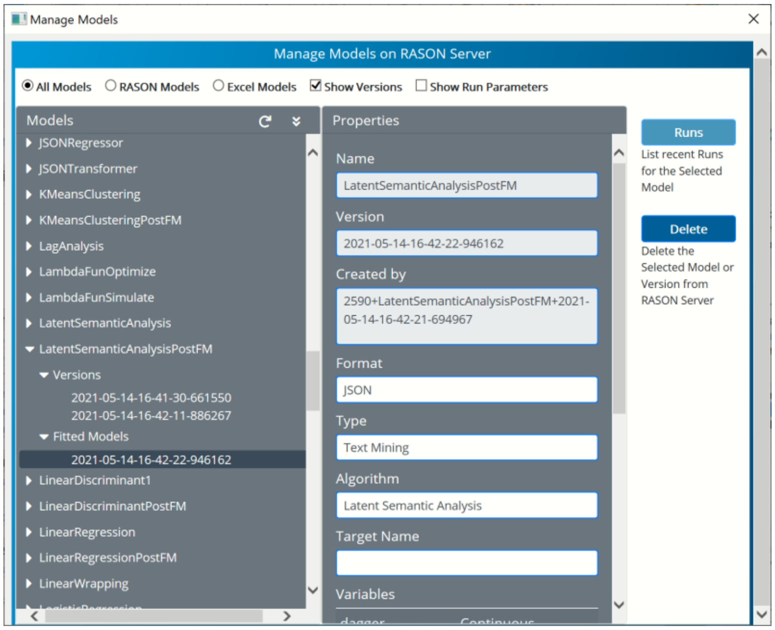
Manage Models

Click Manage Models (top right) to display your models and model versions from your account on www.rason.com.

- Selecting a model name in the left column displays the model properties in the right.
- Select all models to see all your models on your Rason account, Rason as well as Excel models.
- Click Show Versions to see all versions of each named model.
- Click Show Run Parameters to display query parameters.

- With a model name selected, clicking the Runs button displays a list of the most recent model instances in the right column.
- Clicking Delete deletes the selected named model (all versions) or a specific version.

The screenshot below displays a fitted model in the Manage Model dialog.



Manage Models

Manage Models on RASON Server

● All Models ○ RASON Models ○ Excel Models ☑ Show Versions □ Show Run Parameters

Models

- JSONRegressor
- JSONTransformer
- KMeansClustering
- KMeansClusteringPostFM
- LagAnalysis
- LambdaFunOptimize
- LambdaFunSimulate
- LatentSemanticAnalysis
- LatentSemanticAnalysisPostFM
  - Versions
    - 2021-05-14-16-41-30-661550
    - 2021-05-14-16-42-11-886267
  - Fitted Models
    - 2021-05-14-16-42-22-946162
- LinearDiscriminant1
- LinearDiscriminantPostFM
- LinearRegression
- LinearRegressionPostFM
- LinearWrapping

Properties

Name: LatentSemanticAnalysisPostFM

Version: 2021-05-14-16-42-22-946162

Created by: 2590+LatentSemanticAnalysisPostFM+2021-05-14-16-42-21-694967

Format: JSON

Type: Text Mining

Algorithm: Latent Semantic Analysis

Target Name:

Variables

Runs: List recent Runs for the Selected Model

Delete: Delete the Selected Model or Version from RASON Server

Cloud Service

Clicking the first option on the RASON Deployment dialog, Cloud Service, gives you four options, RASON Model, Excel Model, Fitted Model and Probability Model.

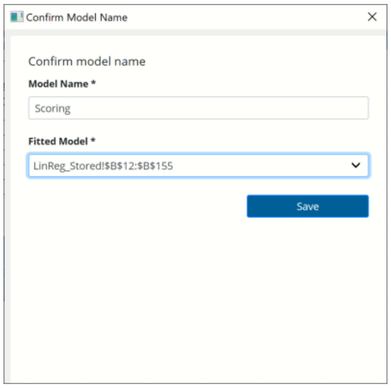
- *RASON Model* translates the Excel model into Rason and POSTs the model to your RASON account. In Analytic Solver Data Science, this option is applicable only for workflows. Please see the previous chapter for more information.
- *Excel Model* POSTs the Excel workbook model to the RASON account. This option is not applicable for data science/forecasting models.
- *Fitted Model* POSTs the fitted model contained within your Excel workbook to your RASON account.
- *Probablity Model* - This option is not supported for data science/forecasting models.

Cloud Service Fitted Model

The majority of Analytic Solver Data Science routines generate fitted, or stored, models which are used to score (classify, predict or forecast) new data points. Selecting Cloud Service – Fitted Model POSTs the fitted model to your account on the RASON server allowing users to utilize their fitted models to score new data on the RASON Server. Continue reading to follow a step-by-step example illustrating this feature.

On the Analytic Solver ribbon, click Help – Example Models – Forecasting / Data Science Examples. Scroll down and click the Scoring example link to open Scoring.xlsx. This workbook contains a stored data science model, LinReg_Stored.

Click RASON Deployment – Cloud Service – Fitted Model. The Confirm Model Name dialog opens.



Confirm model name

Model Name *

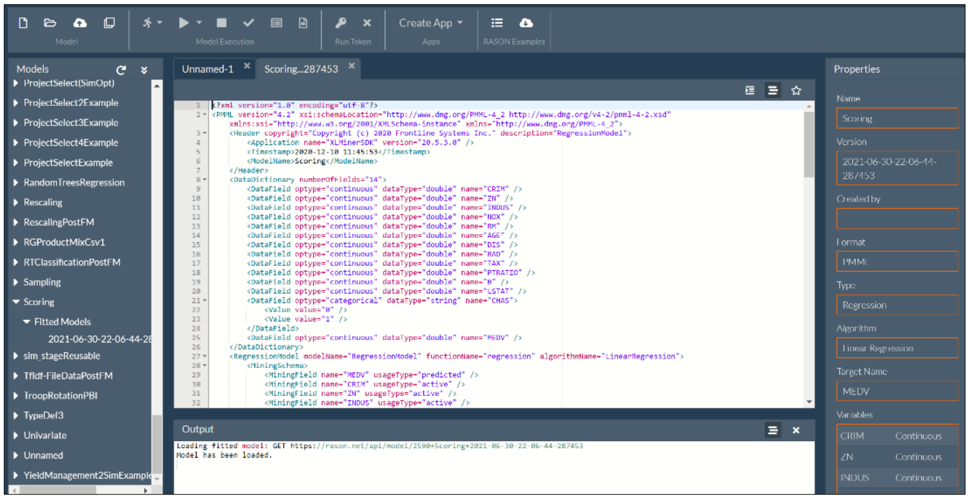
Scoring

Fitted Model *

LinReg_Stored!$B$12:$B$155

- Enter a Model name. This example will use the default, Scoring, but any name will work.
- Click the down arrow beneath Fitted Model to select the fitted model contained within the workbook, LinReg_Stored!$B$12:$B$155.
- Click Save to POST the fitted model to your RASON account.

From here you can maintain the fitted model through RASON Decision Services on RASON.com.



```
<?xml version="1.0" encoding="utf-8"?>
<PMML version="4.2" xsi:schemaLocation="http://www.dmg.org/PMML-4_2 http://www.dmg.org/v4-2/pmml-4-2.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.dmg.org/PMML-4_2">
    <Header copyright="Copyright (c) 2020 Frontline Systems Inc." description="RegressionModel">
        <Application name="XLMinerSDK" version="20.5.3.0" />
        <Timestamp>2020-12-10 11:45:53</Timestamp>
        <ModelName>Scoring</ModelName>
    </Header>
    <DataDictionary numberOfFields="14">
        <DataField optype="continuous" dataType="double" name="CRIM" />
        <DataField optype="continuous" dataType="double" name="ZN" />
        <DataField optype="continuous" dataType="double" name="INDUS" />
        <DataField optype="continuous" dataType="double" name="NOX" />
        <DataField optype="continuous" dataType="double" name="RM" />
        <DataField optype="continuous" dataType="double" name="AGE" />
        <DataField optype="continuous" dataType="double" name="DIS" />
        <DataField optype="continuous" dataType="double" name="RAD" />
        <DataField optype="continuous" dataType="double" name="TAX" />
        <DataField optype="continuous" dataType="double" name="PTRATIO" />
        <DataField optype="continuous" dataType="double" name="B" />
        <DataField optype="continuous" dataType="double" name="LSTAT" />
        <DataField optype="categorical" dataType="string" name="CHAS">
            <Value value="0" />
            <Value value="1" />
        </DataField>
        <DataField optype="continuous" dataType="double" name="MEDV" />
    </DataDictionary>
    <RegressionModel modelName="RegressionModel" functionName="regression" algorithmName="LinearRegression">
        <MiningSchema>
            <MiningField name="MEDV" usageType="predicted" />
            <MiningField name="CRIM" usageType="active" />
            <MiningField name="ZN" usageType="active" />
            <MiningField name="INDUS" usageType="active" />
```

```
Loading fitted model: GET https://rason.net/api/model/2590+Scoring+2021-06-30-22-06-44-287453
Model has been loaded.
```

Name: Scoring

Version: 2021-06-30-22-06-44-287453

Created by:

Format: PMML

Type: Regression

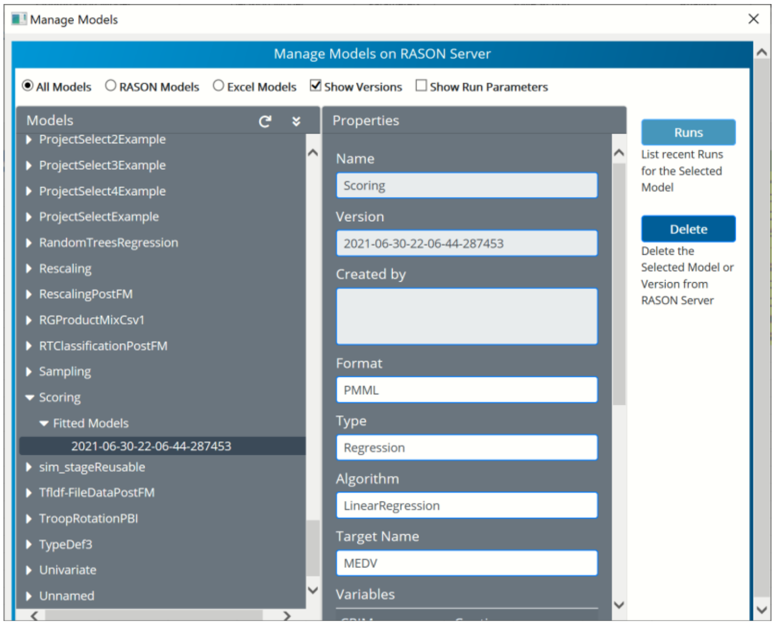
Algorithm: Linear Regression

Target Name: MEDV

Variables

CRIM	Continuous
ZN	Continuous
INDUS	Continuous

The fitted model will also be listed in Manage Models.



Manage Models

Manage Models on RASON Server

● All Models ○ RASON Models ○ Excel Models ☑ Show Versions □ Show Run Parameters

Models

- ProjectSelect2Example
- ProjectSelect3Example
- ProjectSelect4Example
- ProjectSelectExample
- RandomTreesRegression
- Rescaling
- RescalingPostFM
- RGProductMixCsv1
- RTClassificationPostFM
- Sampling
- Scoring
  - Fitted Models
    - 2021-06-30-22-06-44-287453
- sim_stageReusable
- TfIdf-FileDataPostFM
- TroopRotationPBI
- TypeDef3
- Univariate
- Unnamed

Properties

Name: Scoring

Version: 2021-06-30-22-06-44-287453

Created by:

Format: PMML

Type: Regression

Algorithm: LinearRegression

Target Name: MEDV

Variables

Runs: List recent Runs for the Selected Model

Delete: Delete the Selected Model or Version from RASON Server

Power BI

Clicking the second option on the RASON Deployment dialog, Power BI, gives you three options, Managed Model, Embedded Model and Fitted Model. *Fitted Model* POSTs the data science or forecasting fitted model contained within your Excel workbook to your RASON account. The first two options, Managed Model and Embedded Model, do not support data science/forecasting models.

Power BI Fitted Model

Users can choose to deploy and share data science and machine learning models, trained in Analytic Solver or RASON, to the Azure cloud, and use them directly for classification and prediction (without needing auxiliary "code" in R or Python, RASON or Excel). (In the past, Analytic Solver Data Science models could only be scored from within Excel, either online or desktop.) This example illustrates how one could use a fitted model created in Analytic Solver Data Science to score data residing in Power BI, Microsoft's data analytics platform.

Open the Scoring_Heart_Failure.xlsx example workbook by clicking Help – Example Models – Forecasting / Data Science Examples on the Analytic Solver ribbon. This example uses the Heart Failure Clinical Records Dataset[2], which contains thirteen variables describing 299 patients experiencing heart failure. The journal article referenced here discusses how the authors analyzed the dataset to first rank the features (variables) by significance and then used the Random Trees machine learning algorithm to fit a model to the dataset to be used to classify patients at risk of death due to heart failure.

[2] Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020). (link)

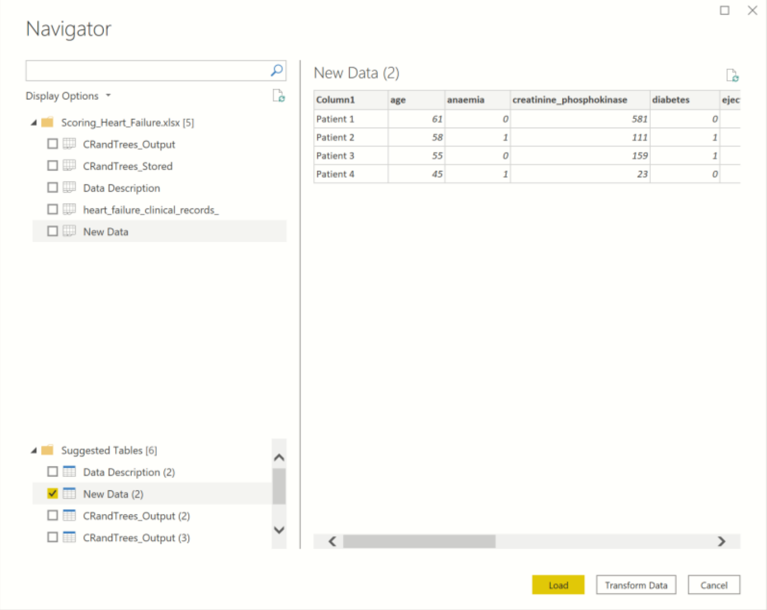
DEATH_EVENT is the output variable in the Random Trees Classification model. This variable can either be a 1 or 0 . A patient classified as a 1 is predicted to perish due to complications of heart failure.

See the Feature Selection chapter within the Data Science User Guide for an example that attempts to emulate the results of that study. For now, this example dataset will illustrate how to score new data contained within Power BI using the Random Trees Classification fitted model saved in the workbook on the CRandTrees_Stored worksheet.

Click the New Data tab to open a list of new patient records which require classification using the stored model, CRandTrees_Stored. It is imperative that patients with a classification of "1" for the output variable, DEATH_EVENT, receive medical mitigation in order to reduce the chance of death from heart failure.

Open Power BI and import the New Data table. (Open Power BI and click Get Data – Excel and select the New Data table, then click Load.)

Note: This dataset includes four rows of new data but only 1 row is required for scoring.

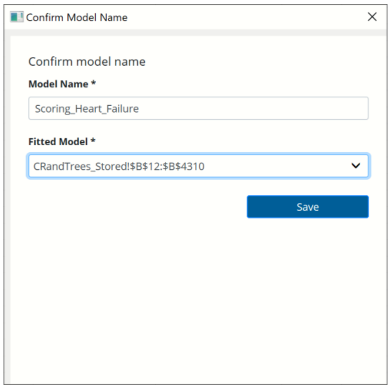


- Scoring_Heart_Failure.xlsx [5]
  - ☐ CRandTrees_Output
  - ☐ CRandTrees_Stored
  - ☐ Data Description
  - ☐ heart_failure_clinical_records_
  - ☐ New Data
- Suggested Tables [6]
  - ☐ Data Description (2)
  - ☑ New Data (2)
  - ☐ CRandTrees_Output (2)
  - ☐ CRandTrees_Output (3)

New Data (2)

Column1	age	anaemia	creatinine_phosphokinase	diabetes	ejec
Patient 1	61	0	581	0	
Patient 2	58	1	111	1	
Patient 3	55	0	159	1	
Patient 4	45	1	23	0	

Click back to the CRandTrees_Stored worksheet in Excel and then click RASON Deployment – Power BI – Fitted model to POST the fitted model, CRAndTrees_Stored, to the RASON Server. This fitted model will be used to score the data that was just imported into Power BI.



Confirm Model Name

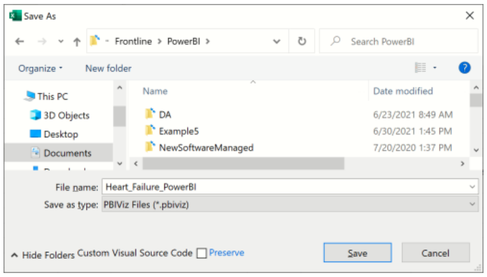
Confirm model name

Model Name * Scoring_Heart_Failure

Fitted Model * CRandTrees_Stored!$B$12:$B$4310

Click the down arrow beneath Fitted Model and select the range CRandTrees_Stored!$B$12:$B$B4310. This is the range where the fitted model is saved on the CRandTrees worksheet.

Enter a file name for the Power BI custom visual file that will be created and saved. This example uses Heart_Failure_PowerBI



Save As

Frontline > PowerBI >

Name	Date modified
DA	6/23/2021 8:49 AM
Example5	6/30/2021 1:45 PM
NewSoftwareManaged	7/20/2020 1:37 PM

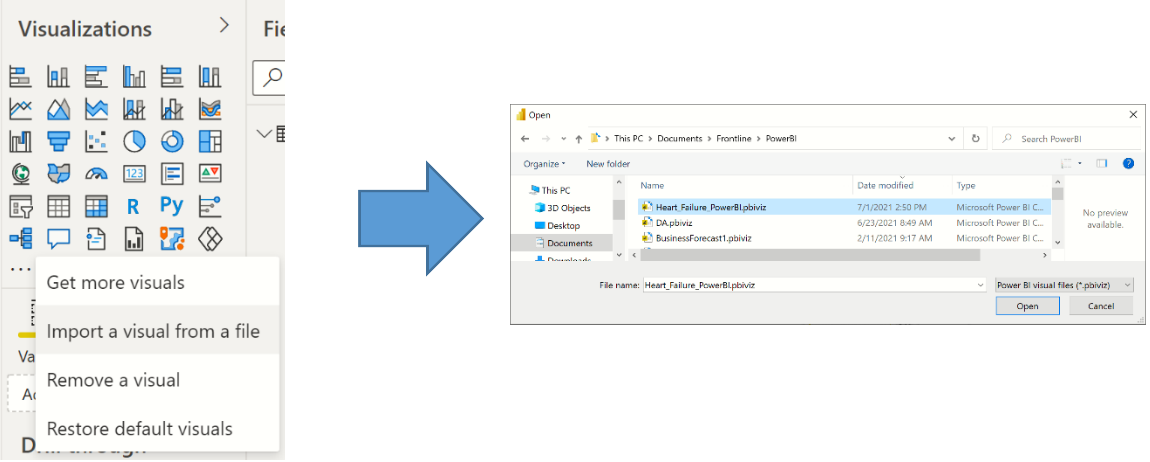
File name: Heart_Failure_PowerBI

Save as type: PBIViz Files (*.pbiviz)

Custom Visual Source Code ☐ Preserve

Click Save to save the Power BI Custom Visual. A command prompt window will appear on your screen and, after several moments, disappear. It is during this time that your Excel model is being translated into the **RASON**® modeling language, and your Power BI custom visual is created.

Click back to Power BI and import the Power BI custom visual by clicking the three dots beneath Visualizations, then browse to open Heart_Failure_PowerBI.pbiviz.



Open

This PC > Documents > Frontline > PowerBI

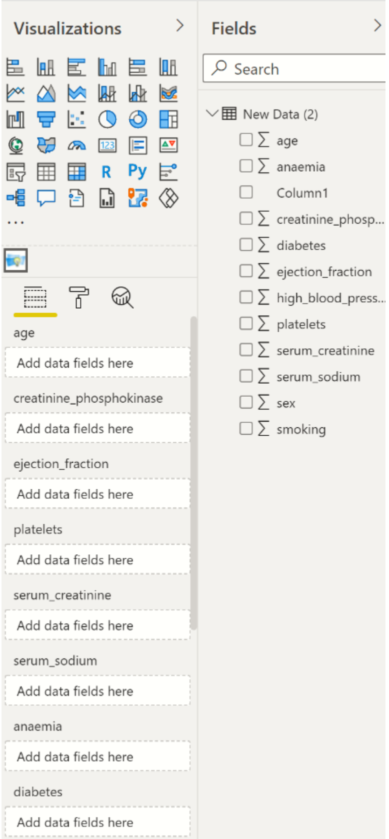
Name	Date modified	Type
Heart_Failure_PowerBI.pbiviz	7/1/2021 2:50 PM	Microsoft Power BI C...
DA.pbiviz	6/23/2021 8:49 AM	Microsoft Power BI C...
BusinessForecast1.pbiviz	2/11/2021 9:17 AM	Microsoft Power BI C...

File name: Heart_Failure_PowerBI.pbiviz

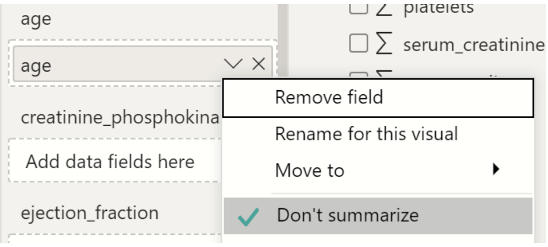
Power BI visual files (*.pbiviz)

Click the newly created Heart_Failure_PowerBI custom visual to display the custom visual data wells.

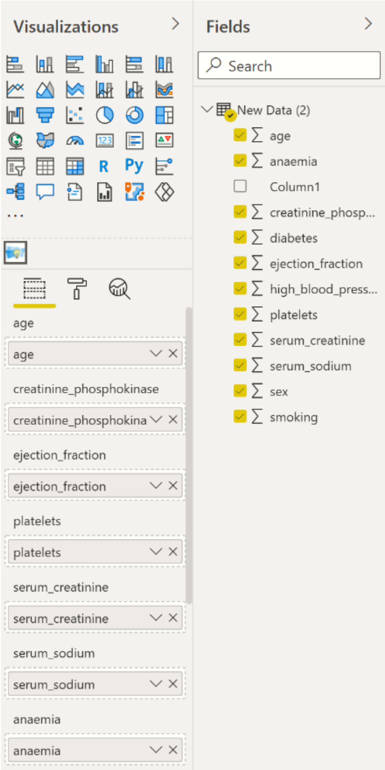
Expand New Data on the right to display the fields contained within the data table.



Match the data into the appropriate data well by dragging the New Data columns to the custom visual data wells, i.e. drag "age" to "age, "anaemia" to "anaemia", etc. Then click the down arrow next to each and select "Don't summarize".



Power BI should look similar to the screenshot below after all fields are matched with the appropriate data wells.



New Data (2)

- ☑ ∑ age
- ☑ ∑ anaemia
- ☐ Column1
- ☑ ∑ creatinine_phosp...
- ☑ ∑ diabetes
- ☑ ∑ ejection_fraction
- ☑ ∑ high_blood_press...
- ☑ ∑ platelets
- ☑ ∑ serum_creatinine
- ☑ ∑ serum_sodium
- ☑ ∑ sex
- ☑ ∑ smoking

age: age

creatinine_phosphokinase: creatinine_phosphokina

ejection_fraction: ejection_fraction

platelets: platelets

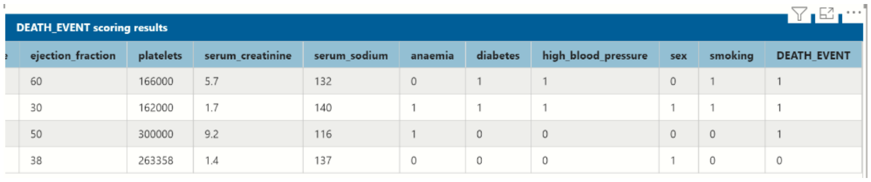
serum_creatinine: serum_creatinine

serum_sodium: serum_sodium

anaemia: anaemia

Once the last data column is match to the last data well, the custom visual will score the data and display the results in the Power BI dashboard, as shown in the screenshot below.

Output results contain the data to be scored plus the column containing the classification of the output variable, DEATH_EVENT.


DEATH_EVENT scoring results

ejection_fraction	platelets	serum_creatinine	serum_sodium	anaemia	diabetes	high_blood_pressure	sex	smoking	DEATH_EVENT
60	166000	5.7	132	0	1	1	0	1	1
30	162000	1.7	140	1	1	1	1	1	1
50	300000	9.2	116	1	0	0	0	0	1
38	263358	1.4	137	0	0	0	1	0	0

The scoring results indicate that the first three, out of the four patients, require some sort of medical mitigation in order to decrease the risk of death from heart failure.

Note that the order that Power BI scores the data may be slightly different than the order that Analytic Solver Data Science scores the data.

Tableau

Clicking the third option on the RASON Deployment dialog, Tableau, gives you three options, Managed Model, Embedded Model and Fitted Model. F*itted Model* POSTs the data science or forecasting fitted model contained within your Excel workbook to your RASON account. The first two options, Managed Model and Embedded Model, do not support data science models.

Tableau Fitted Model

As introduced in the previous section, What's New in Analytic Solver V2021.5, users can now deploy and share data science and machine learning models, trained in Analytic Solver or RASON, to the Azure cloud, and use them directly for classification and prediction (without needing auxiliary "code" in R or Python, RASON or Excel). (In the past, Analytic Solver Data Science models could only be scored from within Excel, either online or desktop.) This example illustrates how one could use a fitted model created in Analytic Solver Data Science to score data residing in Tableau, a visualization platform. For more information on Tableau, see the chapter Creating Custom Extensions in Tableau that appears later in this guide.

Open the Scoring_Heart_Failure.xlsx example workbook by clicking Help – Example Models – Forecasting / Data Science Examples on the Analytic Solver ribbon. This example uses the Heart Failure Clinical Records Dataset[3], which contains thirteen variables describing 299 patients experiencing heart failure. The journal article referenced here discusses how the authors analyzed the dataset to first rank the features (variables) by significance and then used the Random Trees machine learning algorithm to fit a model to the dataset to be used to classify patients at risk of death due to heart failure.

DEATH_EVENT is the output variable in the Random Trees Classification model. This variable can either be a 1 or 0. A patient classified as a 1 is predicted to perish due to complications of heart failure.

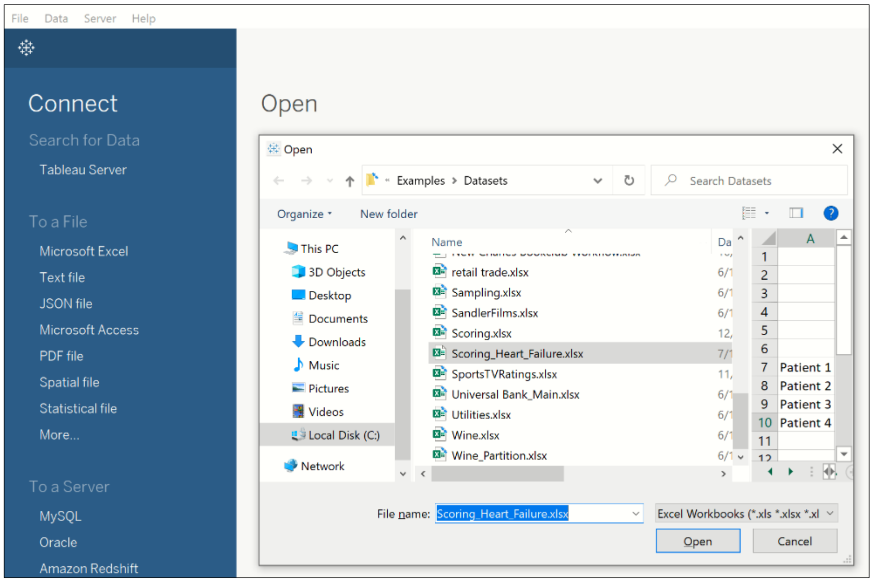
See the Feature Selection chapter within the Analytic Solver Data Science User Guide for an example that attempts to emulate the results of that study. For now, this example dataset will illustrate how to score new data contained within Tableau, using the Random Trees Classification fitted model saved in the workbook.

Click the New Data tab to open a list of new patient records which require classification using the stored model, CRandTrees_Stored. This classification is required in order to determine which patients require medical mitigation to reduce the chance of death from heart failure.

Open Tableau and import the New Data table.

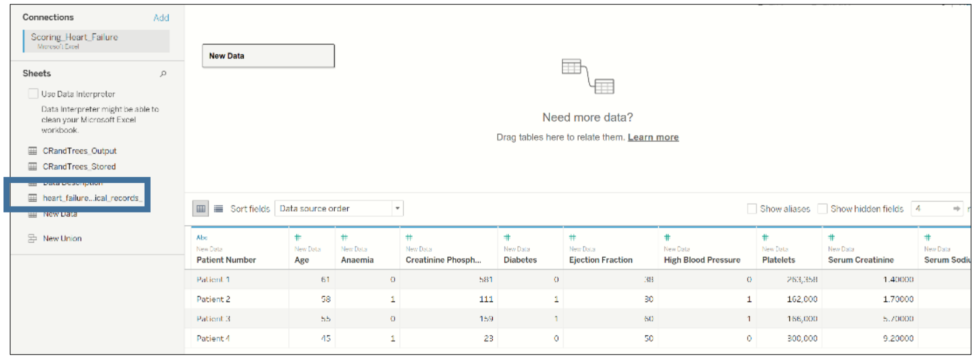
Open Tableau and click Connect To a File: Microsoft Excel. Browse to the location of the Scoring_Heart_Failure.xlsx file typically C:\ProgramData\Frontline Systems\Datasets, then click Open.

[3] Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020). (link)



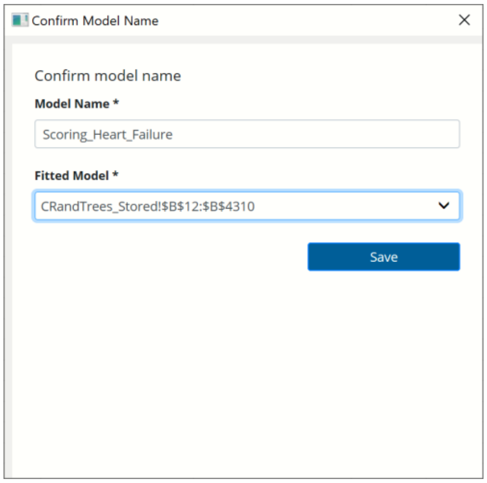
Click the New Data, on the bottom left, to open the table in Tableau. This table contains a list of new patient records which require classification using the stored model, CRandTrees_Stored. It is imperative that patients with a classification of "1" for the output variable, DEATH_EVENT, receive medical mitigation in order to reduce the chance of death from heart failure.

Note: This dataset includes four rows of new data but only 1 row is required for scoring.



Patient Number	Age	Anaemia	Creatinine Phosph...	Diabetes	Ejection Fraction	High Blood Pressure	Platelets	Serum Creatinine	Serum Sodiu
Patient 1	61	0	581	0	38	0	263,358	1.40000	
Patient 2	58	1	111	1	30	1	162,000	1.70000	
Patient 3	55	0	159	1	50	1	166,000	5.70000	
Patient 4	45	1	23	0	50	0	300,000	9.20000	

Click back to the CRandTrees_Stored worksheet in Excel and then click RASON Deployment – Tableau– Fitted model to POST the fitted model, CRAndTrees_Stored, to the RASON Server. This fitted model will be used to score the data that was just imported into Tableau



Confirm Model Name

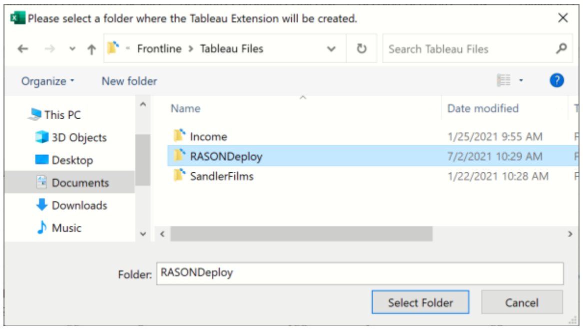
Confirm model name

Model Name *: Scoring_Heart_Failure

Fitted Model *: CRandTrees_Stored!$B$12:$B$4310

Click the down arrow beneath Fitted Model and select the range CRandTrees_Stored!$B$12:$B$B4310. This is the range where the fitted model is saved on the CRandTrees worksheet.

Select a folder to save the Tableau custom extension. This example uses the RASONDeploy folder.

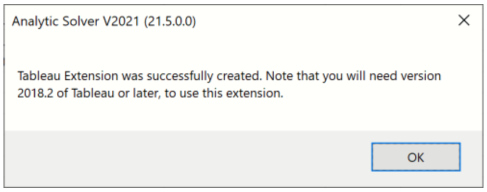


Please select a folder where the Tableau Extension will be created.

Name	Date modified
Income	1/25/2021 9:55 AM
RASONDeploy	7/2/2021 10:29 AM
SandlerFilms	1/22/2021 10:28 AM

Folder: RASONDeploy

Click Select Folder. It is during this time that your Excel model is being translated into the **RASON**® modeling language, and your Tableau custom extension is created. After a few moments, a dialog will appear indicating the Tableau extension has been created successfully.

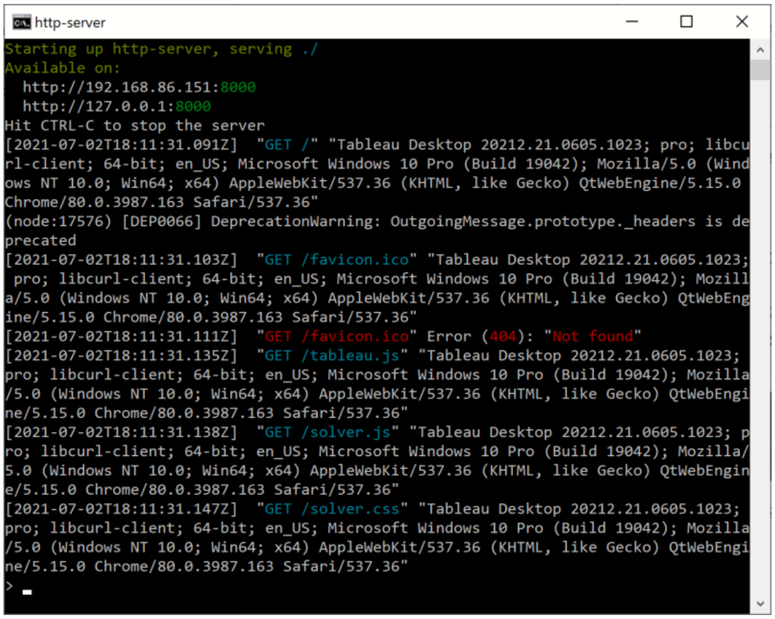


Analytic Solver V2021 (21.5.0.0)

Tableau Extension was successfully created. Note that you will need version 2018.2 of Tableau or later, to use this extension.

Starting up the Server

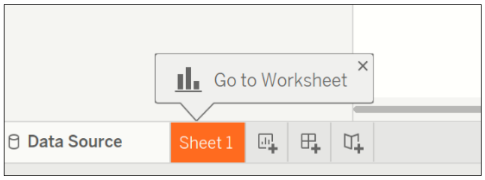
Since Tableau extensions are simply web pages, we will first need to start up a web server to serve our content. For this example, we will serve up the webpage to the default location. To do so, open a command prompt, navigate to the root of the extensions repository and run “http-server -p 8000”.



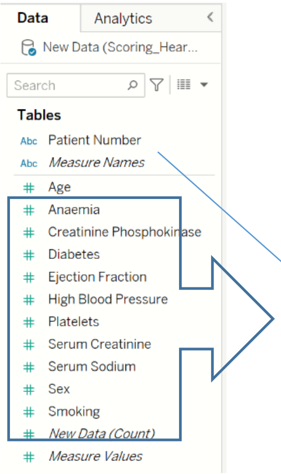
```
Starting up http-server, serving ./
Available on:
  http://192.168.86.151:8000
  http://127.0.0.1:8000
Hit CTRL-C to stop the server
[2021-07-02T18:11:31.091Z]  "GET /" "Tableau Desktop 20212.21.0605.1023; pro; libcu
rl-client; 64-bit; en_US; Microsoft Windows 10 Pro (Build 19042); Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) QtWebEngine/5.15.0
Chrome/80.0.3987.163 Safari/537.36"
(node:17576) [DEP0066] DeprecationWarning: OutgoingMessage.prototype._headers is de
precated
[2021-07-02T18:11:31.103Z]  "GET /favicon.ico" "Tableau Desktop 20212.21.0605.1023;
 pro; libcurl-client; 64-bit; en_US; Microsoft Windows 10 Pro (Build 19042); Mozill
a/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) QtWebEng
ine/5.15.0 Chrome/80.0.3987.163 Safari/537.36"
[2021-07-02T18:11:31.111Z]  "GET /favicon.ico" Error (404): "Not found"
[2021-07-02T18:11:31.135Z]  "GET /tableau.js" "Tableau Desktop 20212.21.0605.1023;
pro; libcurl-client; 64-bit; en_US; Microsoft Windows 10 Pro (Build 19042); Mozilla
/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) QtWebEngi
ne/5.15.0 Chrome/80.0.3987.163 Safari/537.36"
[2021-07-02T18:11:31.138Z]  "GET /solver.js" "Tableau Desktop 20212.21.0605.1023; p
ro; libcurl-client; 64-bit; en_US; Microsoft Windows 10 Pro (Build 19042); Mozilla/
5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) QtWebEngin
e/5.15.0 Chrome/80.0.3987.163 Safari/537.36"
[2021-07-02T18:11:31.147Z]  "GET /solver.css" "Tableau Desktop 20212.21.0605.1023;
pro; libcurl-client; 64-bit; en_US; Microsoft Windows 10 Pro (Build 19042); Mozilla
/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) QtWebEngi
ne/5.15.0 Chrome/80.0.3987.163 Safari/537.36"
> _
```

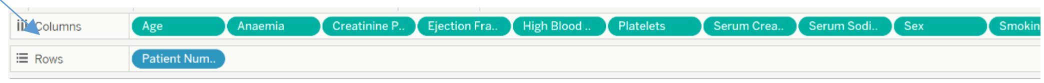
This command starts up a simple http server listening on port 8000.

Click back to Tableau and click Sheet 1.

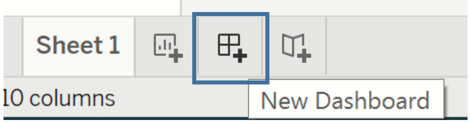


Drag Patient Number to Rows and then drag the 11 features (or variables) to Columns.

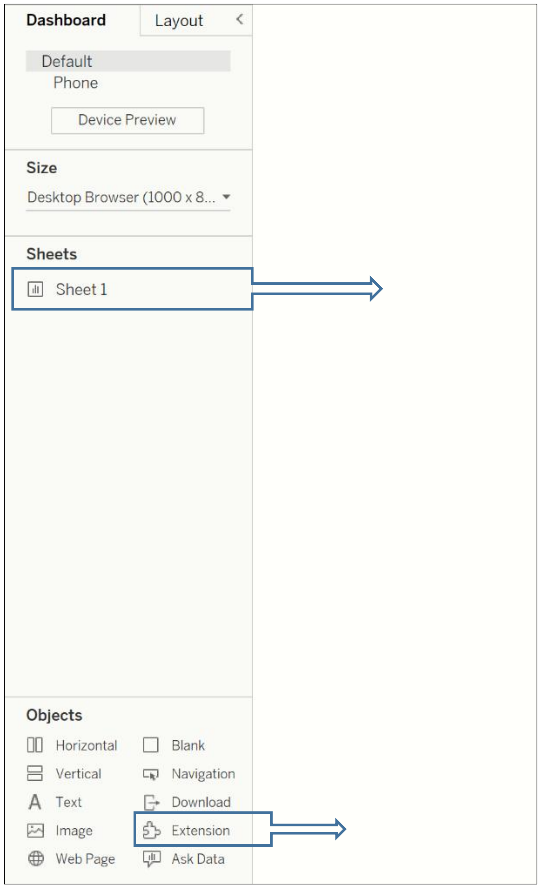




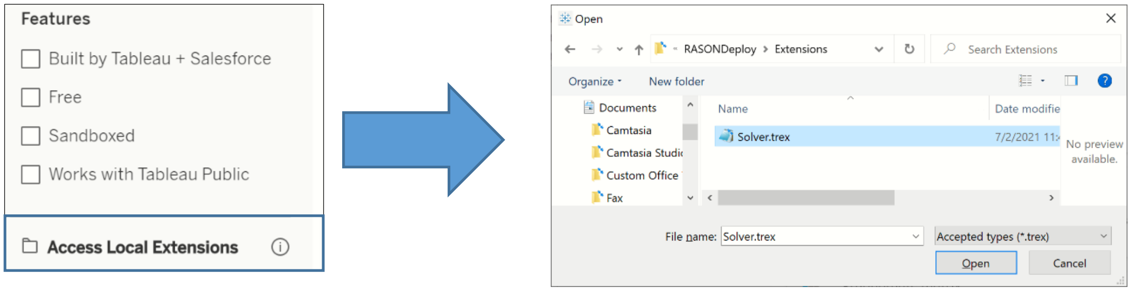
Then click New Dashboard to open a new Tableau dashboard.



Once opened, drag Sheet 1, under Sheets, and Extension, under Objects to the dashboard.

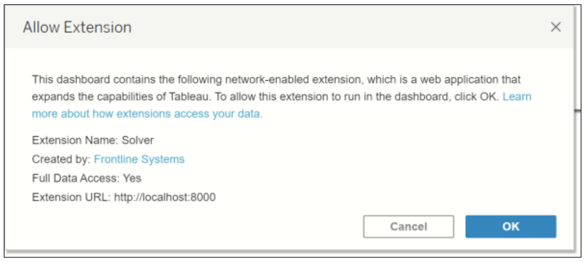


The "Add an Extension" dialog opens. On this dialog, click **Access Local Extensions** on the bottom left and browse to the location of the Tableau .trex file.



File name: Solver.trex

Click Open to open the Tableau extension. Then click OK to allow the extension to run the Tableau dashboard.



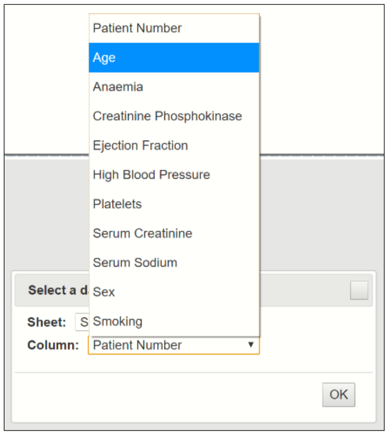
Allow Extension

This dashboard contains the following network-enabled extension, which is a web application that expands the capabilities of Tableau. To allow this extension to run in the dashboard, click OK. Learn more about how extensions access your data.

Extension Name: Solver
Created by: Frontline Systems
Full Data Access: Yes
Extension URL: http://localhost:8000

A series of dialogs will appear which allow the dimensions from the Tableau worksheet to be matched with the dimensions in the Solver custom Tableau extension (Solver.trex).

- Match the data source "age" dimension with the worksheet "age" dimension.

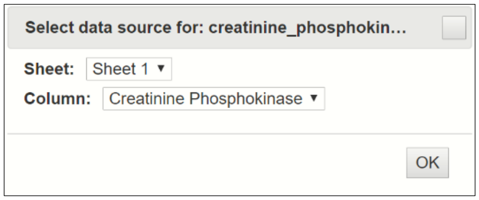


Select a d

Sheet: S

Column: Patient Number

- Match the data source "creatinine_phosphokinase" dimension with the worksheet "Creatinine Phosphokinase" dimension.

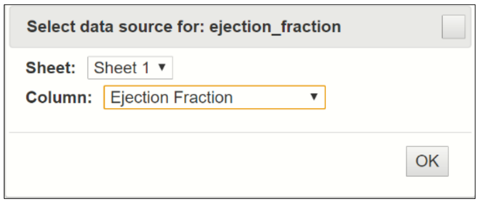


Select data source for: creatinine_phosphokin...

Sheet: Sheet 1

Column: Creatinine Phosphokinase

- Match the data source "ejection_fraction" dimension with the worksheet "Ejection Fraction" dimension.

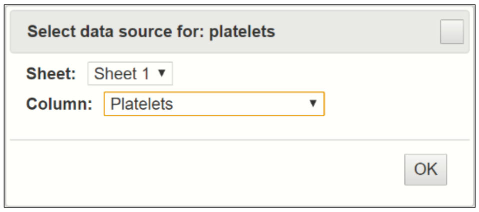


Select data source for: ejection_fraction

Sheet: Sheet 1

Column: Ejection Fraction

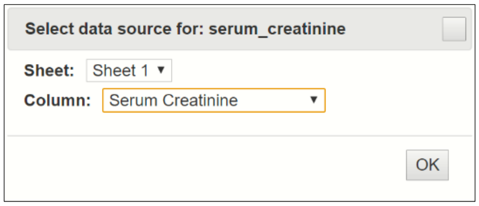
- Match the data source "Platelets" dimension with the worksheet "Platelets" dimension.


Select data source for: platelets

Sheet: Sheet 1

Column: Platelets

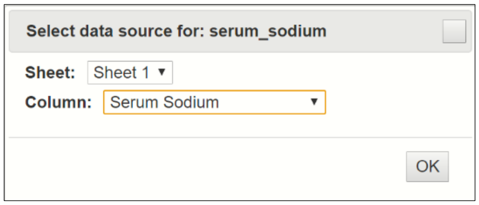
- Match the data source "serum_creatinine" dimension with the worksheet "Serum Creatinine" dimension.


Select data source for: serum_creatinine

Sheet: Sheet 1

Column: Serum Creatinine

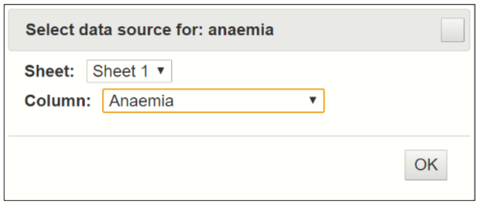
- Match the data source "serum_sodium" dimension with the worksheet "Serum Sodium" dimension.


Select data source for: serum_sodium

Sheet: Sheet 1

Column: Serum Sodium

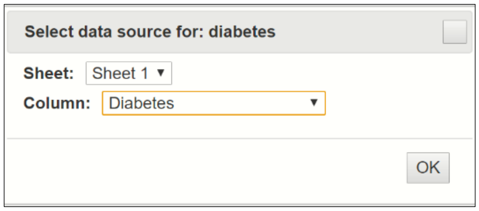
- Match the data source "anaemia" dimension with the worksheet "Anaemia" dimension.


Select data source for: anaemia

Sheet: Sheet 1

Column: Anaemia

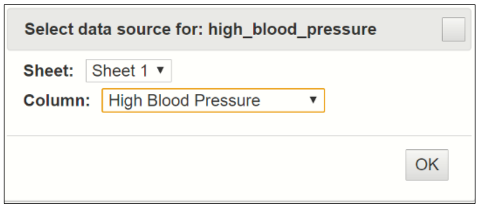
- Match the data source "diabetes" dimension with the worksheet "Diabetes" dimension.


Select data source for: diabetes

Sheet: Sheet 1

Column: Diabetes

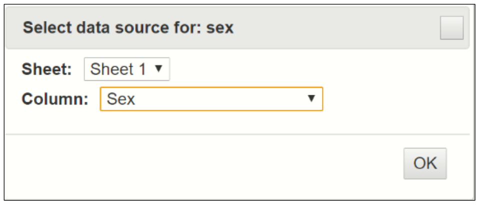
- Match the data source "high_blood_pressure" dimension with the worksheet "High Blood Pressure" dimension.


Select data source for: high_blood_pressure

Sheet: Sheet 1

Column: High Blood Pressure

- Match the data source "sex" dimension with the worksheet "Sex" dimension.

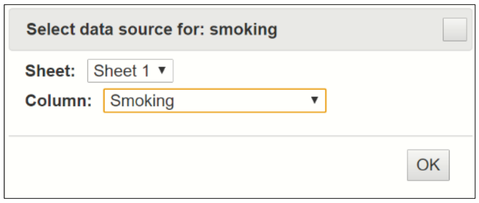


Select data source for: sex

Sheet: Sheet 1

Column: Sex

- Match the data source "smoking" dimension with the worksheet "Smoking" dimension.

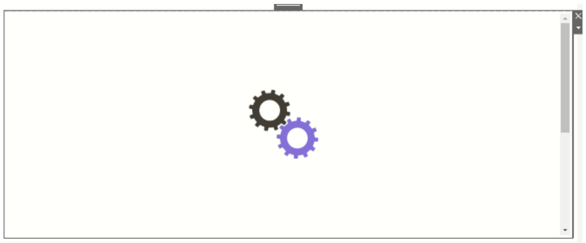


Select data source for: smoking

Sheet: Sheet 1

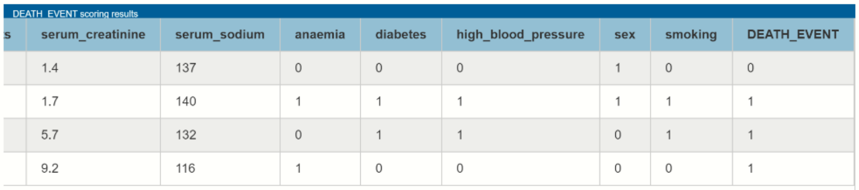
Column: Smoking

Once the last dimension is matched, Tableau immediately begins generating the custom extension.



The screenshot below displays the completed custom extension.

Output results contain the data to be scored plus the column containing the classification of the output variable, DEATH_EVENT.



DEATH EVENT scoring results

s	serum_creatinine	serum_sodium	anaemia	diabetes	high_blood_pressure	sex	smoking	DEATH_EVENT
	1.4	137	0	0	0	1	0	0
	1.7	140	1	1	1	1	1	1
	5.7	132	0	1	1	0	1	1
	9.2	116	1	0	0	0	0	1

The scoring results indicate that the first three, out of the four patients, require some sort of medical mitigation in order to decrease the risk of death from heart failure.

Bringing Big Data into Excel Using Apache Spark

Introduction

Large amounts of data are being generated and collected continuously from a multitude of sources every minute of every day. From your toothbrush to your vehicle GPS to Twitter/Facebook/Google/Yahoo, data is everywhere. Being able to make decisions based on this information requires the ability to extract trends and patterns2 that can be buried deeply within the numbers.

Generally these large datasets contain millions of records (rows) requiring multiple gigabytes or terabytes of storage space across multiple hard drives in an external compute cluster. Analytic Solver Data Science enables users to 'pull' sampled and summarized data into Excel from compute clusters running Apache Spark, the open-source software widely embraced by Big Data vendors and users.

Sampling and Summarizing Big Data

This example illustrates how to use the Big Data Sample/Summarization feature using Data stored across an Apache Spark compute cluster where the Frontline Systems access server is installed. By drawing a *representative sample* of Big Data from all the nodes in the cluster, Excel users can easily train data science and text mining models directly on their desktops.

In this example, we will use the Airline dataset. The data used in this example consists of flight arrival and departure information for all commercial flights within the USA dating from October 1987 to April 2008. This data was obtained from 29 commercial airlines and 3,376 airports and consists of 3.2 million cancelled flights and 25 million flights at least 15 minutes late. This is a large dataset with nearly 120 million records requiring 1.6 GB of storage space when compressed and 12 GB of storage space when uncompressed. Data was obtained from the Research and Innovative Technology Administration (RITA) which coordinates the U.S. Department of Transportation research programs. Note: Southwest (WN), American Airlines (AA), United Airlines (UA), US Airways (US), Continental Airlines (CO), Delta Airlines (DL), Northwest Airlines (NW) and Alaska Airlines (AS) are the only airlines where data is available for all 20 years. Recall the annual revenue from the domestic airline industry is $157 billion. This public dataset was obtained from here. Navigate to this webpage to explore details about this dataset. For supplemental data including the location of each airport, plane type and meteorological data pertaining to each flight, click here.

The information contained in this large dataset could allow us to answer or understand the following questions or issues:

- What are the airports most prone to departure delays? What airports tend to have the most arrival delays?

- What are the times of day and days of week that are most susceptible to departure/arrival delay?
- How can we understand flight patterns as they respond to well-known events? (i.e., examining the data before and after September 2011)
- How many miles per year does each plane by carrier fly?
- When is the best time of day/day of week/time of year to fly to minimize delays?
- How does the number of people flying between different locations change over time?
- How well does weather predict plane delays?
- Can you detect cascading failures as delays in one airport creates delays in others? Are there critical links in the system?
- Understanding flight patterns between the pair of cities that you fly between most often, or all flights to and from a major airport like Chicago (ORD)
- Average arrival delay in minutes by flight or by year?
- How many flights were cancelled, at least 15 minutes late, etc?
- How many flights were less than 50 miles?

Connecting to an Apache Spark Cluster

The first step in connecting Analytic Solver Data Science to your organization's own Apache Spark cluster is to contact Frontline Systems Sales and Technical Support at 775-831-0300. After the server-side software package is installed, the proper entries for the cluster options can be entered as shown in the example below.

Storage Sources and Data Formats

Analytic Solver Data Science can process data from Hadoop Distributed File System (HDFS), local file systems that are visible to Spark cluster, and Amazon S3. Performance is best with HDFS, and it is recommended that you load data from a local file system or Amazon S3 into HDFS. If the local file system is used, the data must be accessible at the same path on all Spark workers, either via a network path, or because it was copied to the same location on all workers.

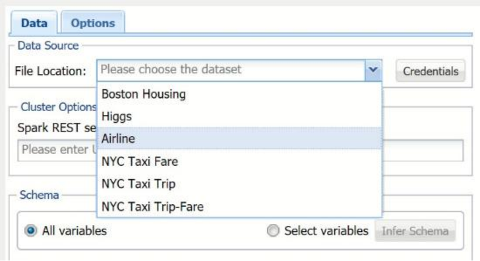
At present, Analytic Solver Data Science can process data in Apache Parquet and CSV (delimited text) formats. Performance is far better with Parquet, which stores data in a compressed, columnar representation; it is highly recommended that you convert CSV data to Parquet before you seek to sample or summarize the data.

Sampling from a Large Dataset

If using desktop Analytic Solver Data Science, click **Get Data – Big Data – Sample** to open the *Sample Big Data* dialog. Enter the location of the file for *File Location* and the URL for the Spark Server for *Spark REST server URL*. This example uses the Airline dataset (described above) installed on a Frontline operated Apache Spark cluster. If your dataset is located on Amazon S3, click Credentials to enter your Access and Secret Keys.

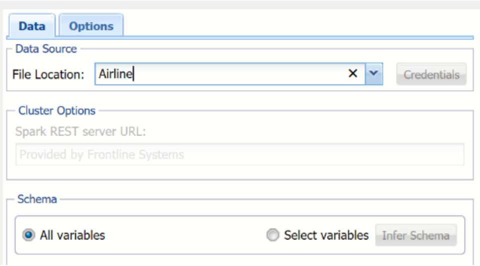
If using AnalyticSolver,com, click the down arrow to open the *File Location* drop down menu and select **Airline**.

Note: Frontline Systems no longer offers Big Data datasets, including the Airline dataset, as a free service. However, the instructions below are applicable to any connected Spark cluster.



Keep *All variables* selected to include all variables in the dataset in the sample.

To only include specific variables, you would choose *Select variables*, then click *Infer Schema*. All variables contained in the dataset will be reported under *Variables*. Use the >/< and >>/<< buttons to select variables to be included in the sample.


File Location: Airline

Spark REST server URL: Provided by Frontline Systems

● All variables ○ Select variables

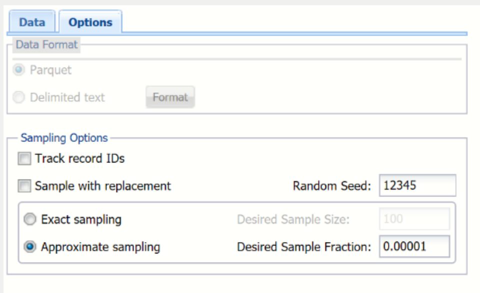
Click the **Options** tab.

Select **Approximate Sampling**. When this option is selected, the size of the resultant sample will be determined by the value entered for *Desired Sample Fraction*. Approximate sampling is much faster than Exact Sampling. Usually, the resultant fraction is very close to the Desired Sample Fraction so this option should be preferred over exact sampling as often as possible. Even if the resultant sample slightly deviates from the desired size, this would be easy to correct in Excel.

Enter **0.00001** for *Desired Sample Fraction*. This option controls the fraction of the total number of records in the full dataset that is expected to be included in the generated sample. Since our dataset contains about 120 million records, our sample will contain approximately 1,200 records. If *Sampling with Replacement* is selected, the value for *Desired Sample Fraction* is the expected number of times each record can be chosen and must be greater than 0. If Sampling **without** replacement (i.e. Sampling with Replacement is not selected), the Desired Sample Fraction becomes the probability that each element is chosen and, as a result, *Desired Sample Fraction* must be between 0 and 1.

Keep *Random Seed* at the default of **12345**. This value initializes the random number generator. This option allows you to generate reproducible samples. *Track record IDs* and *Sample with replacement* should remain unchecked.

Please see the Analytic Solver Data Science Reference guide for a complete description of each option included on this dialog.



Data Format

○ Parquet

○ Delimited text   Format

Sampling Options

□ Track record IDs

□ Sample with replacement   Random Seed: 12345

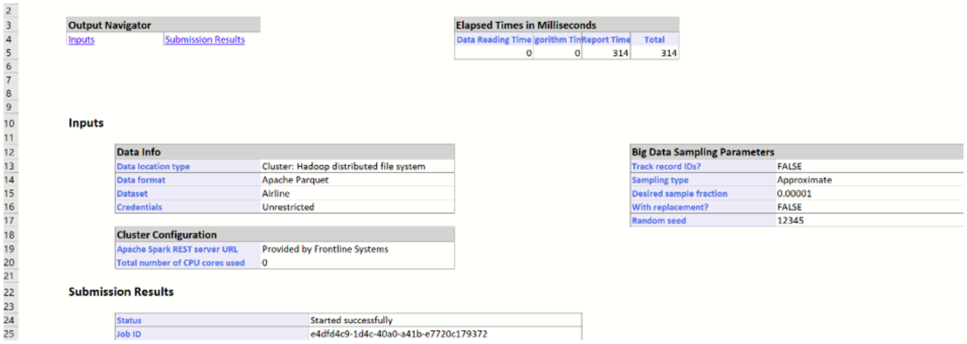
○ Exact sampling   Desired Sample Size: 100

● Approximate sampling   Desired Sample Fraction: 0.00001

Clicking *Submit* sends a request for sampling to the compute cluster but does not wait for completion. The result is output containing the Job ID and basic information about the submitted job so multiple submissions may be identified. This information can be used at any time later for querying the status of the job and generating reports based on the results of the completed job.

Clicking *Run* sends a request for sampling to the compute cluster and waits for the results. Once the job is completed and results are returned to the Analytic Solver client, a report is inserted into the Model tab on the Data Science task pane under Reports – Text Science containing the sampling results.

Click **Submit**.   Results will be inserted into the Data Science Task Pane under Results – Sampling – Run 1. Open BD_Sampling.



Output Navigator

Inputs	Submission Results

Elapsed Times in Milliseconds

Data Reading Time	gorithm Tin	Report Time	Total
0	0	314	314

Inputs

Data Info	
Data location type	Cluster: Hadoop distributed file system
Data format	Apache Parquet
Dataset	Airline
Credentials	Unrestricted

Cluster Configuration	
Apache Spark REST server URL	Provided by Frontline Systems
Total number of CPU cores used	0

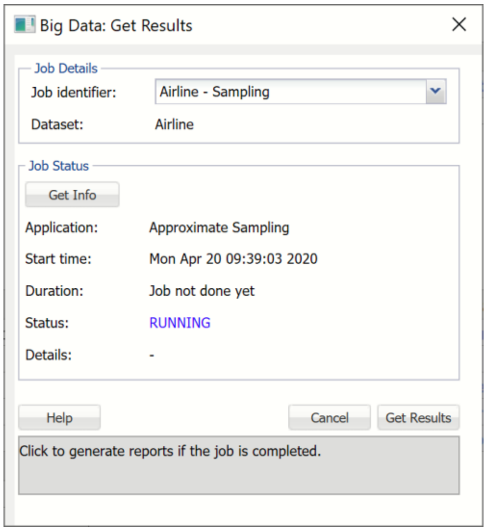
Big Data Sampling Parameters	
Track record IDs?	FALSE
Sampling type	Approximate
Desired sample fraction	0.00001
With replacement?	FALSE
Random seed	12345

Submission Results

Status	Started successfully
Job ID	e4dfd4c9-1d4c-40a0-a41b-e7720c179372

This report displays the details about the chosen dataset, selected options for sampling and the job identifier required for identifying the submission on the cluster.

Click **Get Data – Big Data – Get Results** on the Data Science ribbon to open the *Big Data: Get Results* dialog. Click the down arrow to the right of *Job identifier* and select the previously submitted job. Click **Get Info** to obtain the status of the Job from the cluster.



Big Data: Get Results

Job Details

Job identifier: Airline - Sampling

Dataset: Airline

Job Status

Application: Approximate Sampling

Start time: Mon Apr 20 09:39:03 2020

Duration: Job not done yet

Status: RUNNING

Details: -

Click to generate reports if the job is completed.

Application is the type of the submitted job. This submission corresponds to the Approximate Sampling job that was submitted earlier.

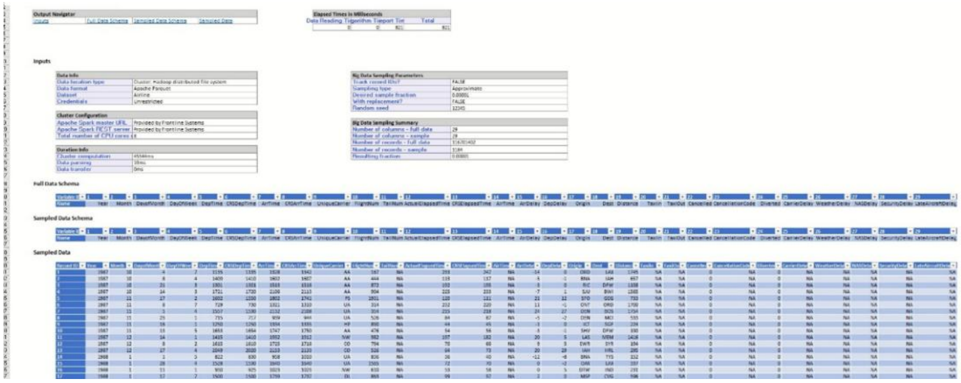
Start Time displays the date and time when the job was submitted. Start Time will always be displayed in the user's Local Time.

Duration shows the elapsed time since job submission if the job is still RUNNING and the cluster total compute time if the job is FINISHED. This job has not yet finished. Once the job is finished, the report will be inserted into the current workbook.

Status is the current state of the job: FINISHED, FAILED, ERRORED or RUNNING. FINISHED indicates that the job has been completed and results are available for retrieval. FAILED or ERRORED indicates that the job has not completed due to an internal cluster failure. When this occurs, *Details* will contain a message indicating the reason.

If Status is FINISHED, you may click the *Get Results* button to obtain the results from the cluster and populate the report as shown below. Note: It is not required to click *Get Info* before *Get Results*. If *Get Results* is clicked, the status of the job will be checked and if the status is FINISHED, the results will be pulled from the cluster and the report will be created. Otherwise, Status will be updated with the appropriate message to reflect the status: FAILED, ERRORED, or RUNNING.

Click **Get Results**. *BD_Sampling1* will be inserted into the Task Pane under Results – Sampling – Run 2. Double click to open.



Output Navigator			
Inputs	Full Data Schema	Sampled Data Schema	Sampled Data

Elapsed Times in Milliseconds			
Data Reading	Algorithm	Report Tim	Total
0	0	821	821

Inputs

Data Info	
Data location type	Cluster: Hadoop distributed file system
Data format	Apache Parquet
Dataset	Airline
Credentials	Unrestricted

Cluster Configuration	
Apache Spark master URL	Provided by Frontline Systems
Apache Spark REST server	Provided by Frontline Systems
Total number of CPU cores	8

Duration Info	
Cluster computation	[illegible]
Data parsing	[illegible]
Data transfer	0ms

Big Data Sampling Parameters	
Track record IDs?	FALSE
Sampling type	Approximate
Desired sample fraction	0.0001
With replacement?	FALSE
Random seed	12345

Big Data Sampling Summary	
Number of columns - full data	29
Number of columns - sample	29
Number of records - full data	[illegible]
Number of records - sample	[illegible]
Resulting fraction	[illegible]

Full Data Schema

Sampled Data Schema

Sampled Data

The *Inputs* section displays the information about the dataset, cluster configuration, details on the running time, the options chosen during setup, and a summary of the data including the size and dimensionality of the full and sampled datasets. Since Approximate Sample was selected, we can expect the resulting fraction to be slightly different from the desired fraction. In our example, the resulting fraction, approximately 0.00001014 (1,184/116,701,402) is very close to the requested fraction (0.00001)). (Recall *Approximate Sample* was selected and *Desired Sample Fraction* was entered on the *Options* tab during setup.)

Scroll down to see the full and sampled data schemas. Since we choose to include all variables in the sample, the set of columns in the full and sampled datasets is the same.

Further down we see the Sampled Data, which includes 1,184 records, as indicated by the *Number of records – sample* field under Sampled Data Summary.

Now that the representative data sample has been drawn and is available in the output, all of the methods and features included in Analytic Solver Data Science are literally at your fingertips. We could choose to explore our sampled data by creating visualizations using the Chart Wizard, transform the data using Analytic Solver Data Science's data transformation utilities, build classification/prediction models to forecast arrival and departure times, predict airport delays, estimate total flight times and perform any other analytic tasks that can address numerous challenges that Big Data, and the Airline dataset in particular, present to the data scientists and analysts.

Summarizing a Large Dataset

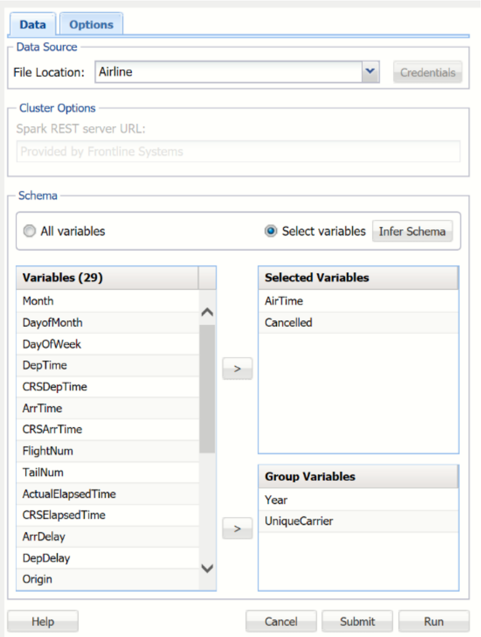
The Big Data Summarization feature in Analytic Solver Data Science translates the lightning-fast cluster computing capabilities from the state of the art Big Data engine, Apache Spark, to the simple and easy to use, "point & click" interface within Excel. This very powerful yet intuitive tool is useful for rapid extraction of key metrics contained in data, which can be immediately used by data analysts and decision makers. The new Summarization feature in Analytic Solver Data Science provides similar functionality as standard SQL engines, but for the data, volume and complexity which extends far beyond your desktop or laptop computer. This tool is a great assistant for composing reports, constructing informative visualizations, building prescriptive and predictive models that can drive the directions of consequent analysis.

Now we will illustrate how to utilize this easy-to-use yet powerful tool by summarizing the Airport dataset and using the information obtained to answer the following three questions:

- What carrier has the most domestic flights by year?
- Who are the most reliable airlines?
- Who are the least reliable airlines?

Click **Get Data – Big Data – Summarize** on the Data Science Ribbon. This time we will select a subset of variables for summarization along with grouping variables, so (after entering the File location, Spark REST server URL and file Credentials, if needed) choose **Select variables** and click **Infer Schema**. Afterwards transfer **ArrTime**, and **Cancelled** to the *Selected Variables* grid and **Year** and **UniqueCarrier** to the *Group Variables* grid. Group Variables are variables from the dataset that are treated as key variables for aggregation. In this example, the variables will be grouped so that all records with the same Year and UniqueCarrier are included in the same group, and then all aggregate functions for each group will be calculated.

Note: If *All variables* is selected, the result is a simple aggregation of all variables across the entire dataset which can be used to quickly obtain overall statistics.



Data | Options

Data Source

File Location: Airline | Credentials

Cluster Options

Spark REST server URL:

Provided by Frontline Systems

Schema

○ All variables ● Select variables | Infer Schema

Variables (29)
Month
DayofMonth
DayOfWeek
DepTime
CRSDepTime
ArrTime
CRSArrTime
FlightNum
TailNum
ActualElapsedTime
CRSElapsedTime
ArrDelay
DepDelay
Origin

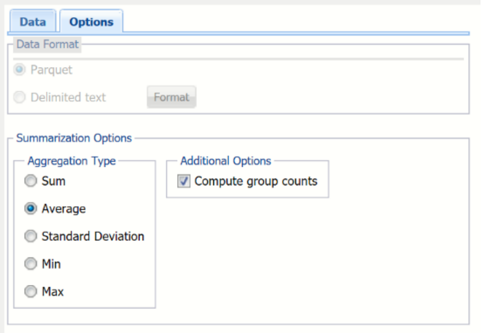
Selected Variables
AirTime
Cancelled

Group Variables
Year
UniqueCarrier

Click the **Options** tab and select **Average** for *Aggregation Type* and **Compute Group Counts**.

Aggregation Type provides 5 statistics that can be inferred from the dataset: sum, average, standard deviation, minimum and maximum.

The option *Compute group counts* is enabled when 1 or more Grouping Variables is selected. When this option is selected, the number of records belonging to each group is computed and reported.



Data | Options

Data Format

● Parquet

○ Delimited text | Format

Summarization Options

Aggregation Type

○ Sum

● Average

○ Standard Deviation

○ Min

○ Max

Additional Options

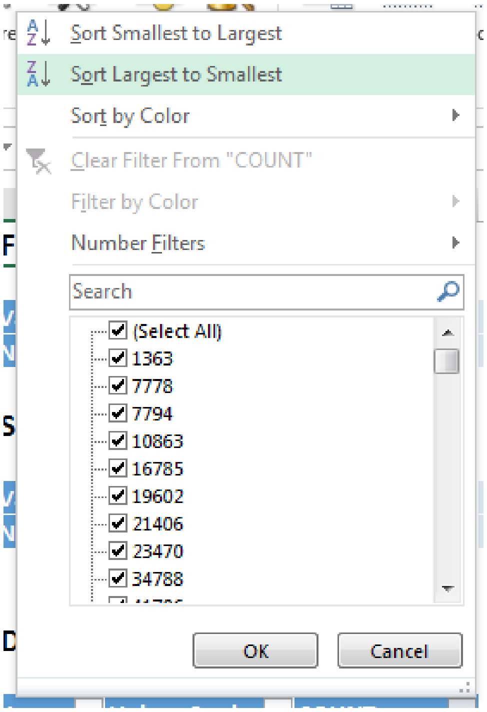
☑ Compute group counts

Click **Run** to send a request for a summarization job to the cluster and wait for the results. Once the job is completed, *BD_Summarization* will be inserted into the current workbook and the Data Science task pane under *Results – Sampling – Run 2*.

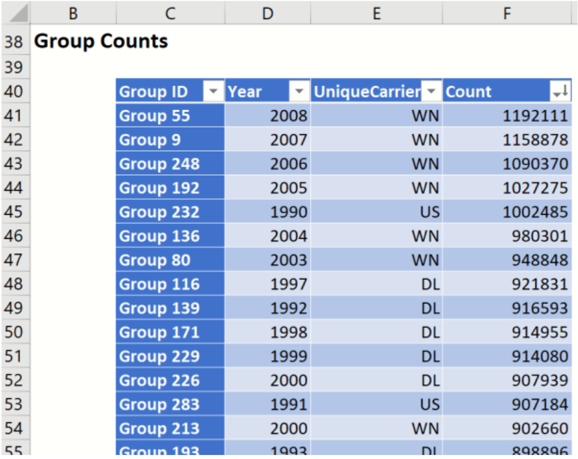
Once again the *Inputs* section recaps the dataset details, the cluster configuration, the time taken to complete the job, the options selected during setup and the number of columns and records in both the full dataset and the summarized data. *Full Data Schema* displays all variables in the dataset while *Summarized Data Schema* displays only the variables that were selected during setup.

Scroll down to *Group Counts* to examine the number of records belonging to each Year and UniqueCarrier. In this example, there were 405,598 US Airways flights in 2003 and 684,961 Southwest flights in 1995.

Click the down arrow next to **Count** and select **Sort Largest to Smallest** to answer our first question, "What carrier has the most domestic flights by year?"



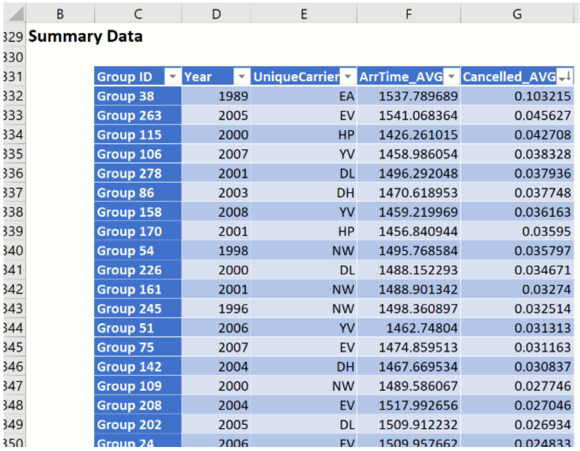
Once, the table is sorted on the **Count** column, we see that Southwest (WN) holds the largest market share in domestic flights for years 2005 - 2008.



38 Group Counts

Group ID	Year	UniqueCarrier	Count
Group 55	2008	WN	1192111
Group 9	2007	WN	1158878
Group 248	2006	WN	1090370
Group 192	2005	WN	1027275
Group 232	1990	US	1002485
Group 136	2004	WN	980301
Group 80	2003	WN	948848
Group 116	1997	DL	921831
Group 139	1992	DL	916593
Group 171	1998	DL	914955
Group 229	1999	DL	914080
Group 226	2000	DL	907939
Group 283	1991	US	907184
Group 213	2000	WN	902660
Group 193	1993	DL	898896

Scroll down to Summary Data to find some evidence (that can be further verified) of the most and least "reliable" airlines. Click the down arrow next to *Cancelled_AVG* and sort from largest to smallest. The airline with the largest average percentage of cancelled flights is Eastern Airlines (EA) with a little over 10% of their flights cancelled on average in 1989. ExpressJet Airlines (EV) and America West (HP) round out the top three spots with 4.5% and 4.3% respectively.

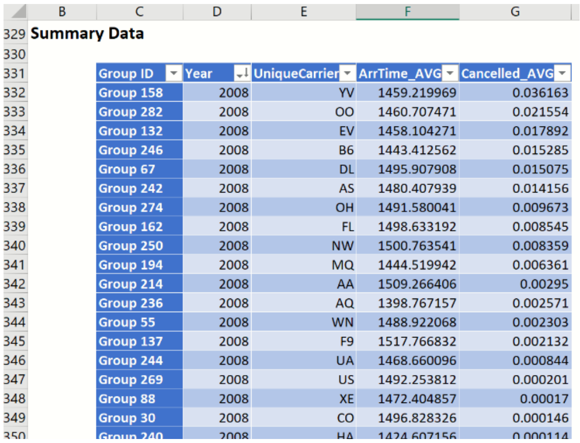


329 Summary Data

330

	Group ID	Year	UniqueCarrier	ArrTime_AVG	Cancelled_AVG
332	Group 38	1989	EA	1537.789689	0.103215
333	Group 263	2005	EV	1541.068364	0.045627
334	Group 115	2000	HP	1426.261015	0.042708
335	Group 106	2007	YV	1458.986054	0.038328
336	Group 278	2001	DL	1496.292048	0.037936
337	Group 86	2003	DH	1470.618953	0.037748
338	Group 158	2008	YV	1459.219969	0.036163
339	Group 170	2001	HP	1456.840944	0.03595
340	Group 54	1998	NW	1495.768584	0.035797
341	Group 226	2000	DL	1488.152293	0.034671
342	Group 161	2001	NW	1488.901342	0.03274
343	Group 245	1996	NW	1498.360897	0.032514
344	Group 51	2006	YV	1462.74804	0.031313
345	Group 75	2007	EV	1474.859513	0.031163
346	Group 142	2004	DH	1467.669534	0.030837
347	Group 109	2000	NW	1489.586067	0.027746
348	Group 208	2004	EV	1517.992656	0.027046
349	Group 202	2005	DL	1509.912232	0.026934
350	Group 24	2006	EV	1509.957662	0.024833

Click the down arrow next to Year and sort from largest to smallest. Now we see that Mesa Airlines held the largest on average cancellation percentage in 2008 (.036%). The second least reliable airline in 2008 goes to SkyWest Airlines (OO) with an average cancellation percentage of 2.2% and the third least reliable airline in 2008 is "awarded" to ExpressJet Airlines (EV) with an on average cancellation percentage of 1.8%.

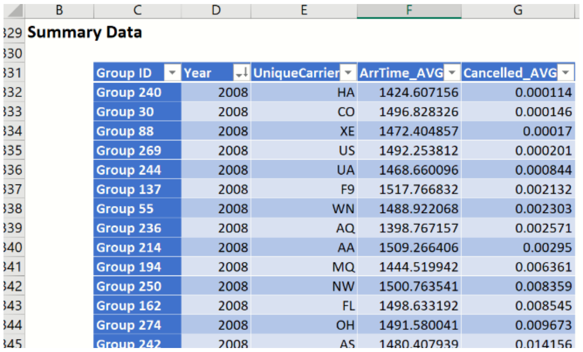


329 Summary Data

330

	Group ID	Year	UniqueCarrier	ArrTime_AVG	Cancelled_AVG
332	Group 158	2008	YV	1459.219969	0.036163
333	Group 282	2008	OO	1460.707471	0.021554
334	Group 132	2008	EV	1458.104271	0.017892
335	Group 246	2008	B6	1443.412562	0.015285
336	Group 67	2008	DL	1495.907908	0.015075
337	Group 242	2008	AS	1480.407939	0.014156
338	Group 274	2008	OH	1491.580041	0.009673
339	Group 162	2008	FL	1498.633192	0.008545
340	Group 250	2008	NW	1500.763541	0.008359
341	Group 194	2008	MQ	1444.519942	0.006361
342	Group 214	2008	AA	1509.266406	0.00295
343	Group 236	2008	AQ	1398.767157	0.002571
344	Group 55	2008	WN	1488.922068	0.002303
345	Group 137	2008	F9	1517.766832	0.002132
346	Group 244	2008	UA	1468.660096	0.000844
347	Group 269	2008	US	1492.253812	0.000201
348	Group 88	2008	XE	1472.404857	0.00017
349	Group 30	2008	CO	1496.828326	0.000146
350	Group 240	2008	HA	1424.607156	0.000114

Click the down array next to *Cancelled_AVG* again, but this time sort from *Smallest to Largest*. Then sort by *Year* from *Largest to Smallest*. The table is updated to display the airlines with the smallest on average flight cancellation percentage in 2008: 1st place – Hawaiian Airlines, 2nd place – Continental Airlines and 3rd place – ExpressJet Airlines (1).



Summary Data

Group ID	Year	UniqueCarrier	ArrTime_AVG	Cancelled_AVG
Group 240	2008	HA	1424.607156	0.000114
Group 30	2008	CO	1496.828326	0.000146
Group 88	2008	XE	1472.404857	0.00017
Group 269	2008	US	1492.253812	0.000201
Group 244	2008	UA	1468.660096	0.000844
Group 137	2008	F9	1517.766832	0.002132
Group 55	2008	WN	1488.922068	0.002303
Group 236	2008	AQ	1398.767157	0.002571
Group 214	2008	AA	1509.266406	0.00295
Group 194	2008	MQ	1444.519942	0.006361
Group 250	2008	NW	1500.763541	0.008359
Group 162	2008	FL	1498.633192	0.008545
Group 274	2008	OH	1491.580041	0.009673
Group 242	2008	AS	1480.407939	0.014156

Using the same steps illustrated here, we could find answers to many other questions, for example:

- What are the yearly flight volume per carrier?
- Which times of day and days of week are most susceptible to departure/arrival delays?
- How many miles per year does each plane by carrier fly?

Concluding Remarks

The ability to sample and summarize large datasets is one that will become more and more important as technology progresses and more and more data is captured. Analytic Solver Data Science's Big Data feature allows users to import these large datasets into Excel allowing business analysts and data scientists the power to build predictive and prescriptive analytic models in their spreadsheets, without the need for complex programming skills. Using Analytic Solver Data Science's Big Data feature, we could easily answer the questions posed in the introduction and more.

Fitting a model using Feature Selection

What is Feature Selection?

Analytic Solver Data Science's Feature Selection tool gives users the ability to rank and select the most relevant variables for inclusion in a classification or prediction model. In many cases the most accurate models, or the models with the lowest misclassification or residual errors, have benefited from better feature selection, using a combination of human insights and automated methods. Analytic Solver Data Science provides a facility to compute all of the following metrics, described in the literature, to give users information on what features should be included, or excluded, from their models.

- **Correlation-based**
  - Pearson product-moment correlation
  - Spearman rank correlation
  - Kendall concordance
- **Statistical/probabilistic independence metrics**
  - Welch's statistic
  - F statistic
  - Chi-square statistic
- **Information-theoretic metrics**
  - Mutual Information (Information Gain)
  - Gain Ratio
- **Other**
  - Cramer's V
  - Fisher score
  - Gini index

Only some of these metrics can be used in any given application, depending on the characteristics of the input variables (features) and the type of problem. In a supervised setting, if we classify data science problems as follows:

- $\mathbb{R}^n \rightarrow \mathbb{R}$: real-valued features, prediction (regression) problem
- $\mathbb{R}^n \rightarrow \{0, 1\}$: real-valued features, binary classification problem
- $\mathbb{R}^n \rightarrow \{1..C\}$: real-valued features, multi-class classification problem
- $\{1..C\}^n \rightarrow \mathbb{R}^n$: nominal categorical features, prediction (regression) problem
- $\{1..C\}^n \rightarrow \{0, 1\}$: nominal categorical features, binary classification problem
- $\{1..C\}^n \rightarrow \{1..C\}$: nominal categorical features, multi-class classification problem

then we can describe the applicability of the Feature Selection metrics by the following table:

	R-R	R-{0,1}	R-{1..C}	{1..C}-R	{1..C}-{0,1}	{1..C}-{1..C}
Pearson	**N**					
Spearman	**N**					
Kendall	**N**					
Welch's	D	**N**				
F-Test	D	**N**	**N**			
Chi-squared	D	D	D	D	**N**	**N**
Mutual Info	D	D	D	D	**N**	**N**
Gain Ratio	D	D	D	D	**N**	**N**
Fisher	D	**N**	**N**			
Gini	D	**N**	**N**			

"N" means that metrics can be applied naturally, and “D” means that features and/or the outcome variable must be discretized before applying the particular filter.

As a result, depending on the variables (features) selected and the type of problem chosen in the first dialog, various metrics will be available or disabled in the second dialog.

Feature Selection Example

The goal of this example is three-fold: 1. To use Feature Selection as a tool for exploring relationships between features and the outcome variable, 2. Reducing the dimensionality based on the Feature Selection results and 3. Evaluating the performance of a supervised learning algorithm (a classification algorithm) for different feature subsets.

This example uses the Heart Failure Clinical Records Dataset[4], which contains thirteen variables describing 299 patients experiencing heart failure. The journal article referenced here discusses how the authors analyzed the dataset to first rank the features (variables) by significance and then used the Random Trees machine learning algorithm to fit a model to the dataset. This example attempts to emulate their results.

A description of each variable contained in the dataset appears in the table below.

VARIABLE	DESCRIPTION
AGE	Age of patient

[4] Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020). (link)

ANAEMIA	Decrease of red blood cells or hemoglobin (boolean)
CREATINE_PHOSPHOKINASE	Level of the CPK enzyme in the blood (mcg/L)
DIABETES	If the patient has diabetes (boolean)
EJECTION_FRACTION	Percentage of blood leaving the heart at each contraction (percentage)
HIGH_BLOOD_PRESSURE	If the patient has hypertension (boolean)
PLATELETS	Platelets in the blood (kiloplatelets/mL)
SERUM_CREATININE	Level of serum creatinine in the blood (mg/dL)
SERUM_SODIUM	Level of serum sodium in the blood (mEq/L)
SEX	Woman (0) or man (1)
SMOKING	If the patient smokes or not (boolean)
TIME	Follow-up period (days)
DEATH_EVENT	If the patient deceased during the follow-up period (boolean)

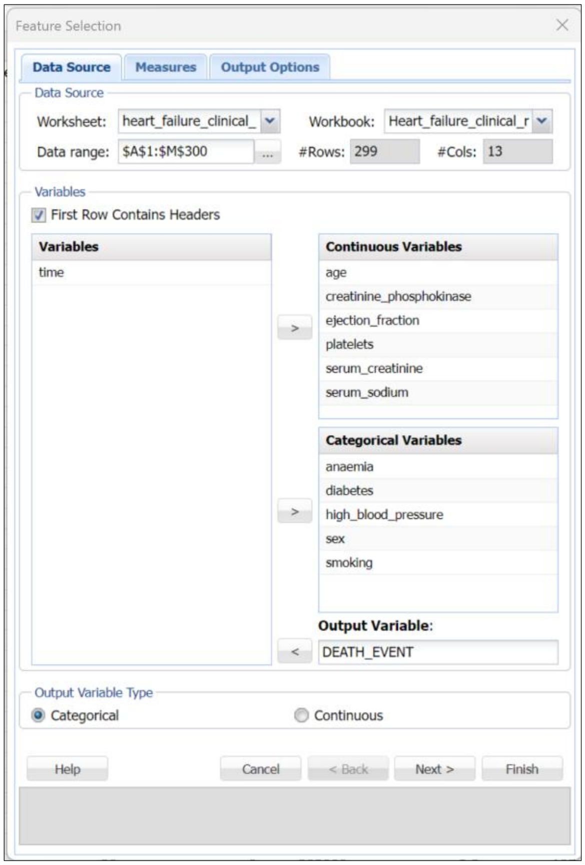
To open the example dataset, click **Help – Example Models – Forecasting/Data Science Examples** – **Heart Failure Clinical Records**.

Select a cell within the data (say A2), then click **Explore – Feature Selection** to bring up the first dialog.

- Select all the following variables as Continuous Variables: age, creatinine_phosphokinase, ejection_fraction, platelets, serum_creatinine and serum_sodium.
- Select the following variables as Categorical Variables: anaemia, diabetes, high_blood_pressure, sex and smoking.
- Select DEATH_EVENT as the Output Variable
- Confirm that Categorical is selected for Output Variable Type. This setting denotes that the Output Variable is a categorical variable. If the number of unique values in the *Output variable* is greater than 10, then *Continuous* will be selected by default. However, at any time the User may override the default choice based on his or her own knowledge of the variable.

This analysis omits the time variable. The Feature Selection dialog should look similar to the screenshot below.

Figure 1: Feature Selection Data Source dialog



Data Source

Worksheet: heart_failure_clinical_

Workbook: Heart_failure_clinical_r

Data range: $A$1:$M$300

#Rows: 299

#Cols: 13

Variables

☑ First Row Contains Headers

Variables: time

Continuous Variables: age, creatinine_phosphokinase, ejection_fraction, platelets, serum_creatinine, serum_sodium

Categorical Variables: anaemia, diabetes, high_blood_pressure, sex, smoking

Output Variable: DEATH_EVENT

Output Variable Type

● Categorical ○ Continuous

Click the **Measures** tab or click **Next** to open the Measures dialog.

Since we have continuous variables, *Discretize predictors* is enabled. When this option is selected, Analytic Solver Data Science will transform continuous variables into discrete, categorical data in order to be able to calculate statistics, as shown in the table in the Introduction to this chapter.

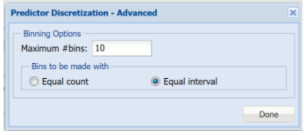
This dataset contains both continuous (or real-valued) features and categorical features which puts this dataset into the following category.

$\mathbb{R}^n \rightarrow \{0, 1\}$: real-valued features, binary classification problem

As a result, if interested in evaluating the relevance of features according to the Chi-Squared Test or measures available in the Information Theory group (Mutual Information and Gain ratio), the variables must first be discretized.

Select **Discretize predictors**, then click **Advanced**. Leave the defaults of **10** for *Maximum # bins* and **Equal Interval** for *Bins to be made with*. Analytic Solver Data Science will create 10 bins and will assign records to the bins based on if the variable's value falls in the interval of the bin. This will be performed for each of the Continuous Variables.

Figure 2: Predictor Discretization - Advanced



Predictor Discretization - Advanced

Binning Options

Maximum #bins: 10

Bins to be made with

○ Equal count ● Equal interval

Note: Discretize output variable is disabled because our output variable, DEATH_EVENT, is already a categorical nominal variable. If we had no Continuous Variables and all Categorical Variables, *Discretize predictors* would be disabled.

Select **Chi-squared** and **Cramer's V** under *Chi-Squared Test*.   The Chi-squared test statistic is used to assess the statistical independence of two events. When applied to Feature Selection, it is used as a test of independence to assess whether the assigned class is independent of a particular variable. The minimum value for this statistic is 0. The higher the Chi-Squared statistic, the more independent the variable.

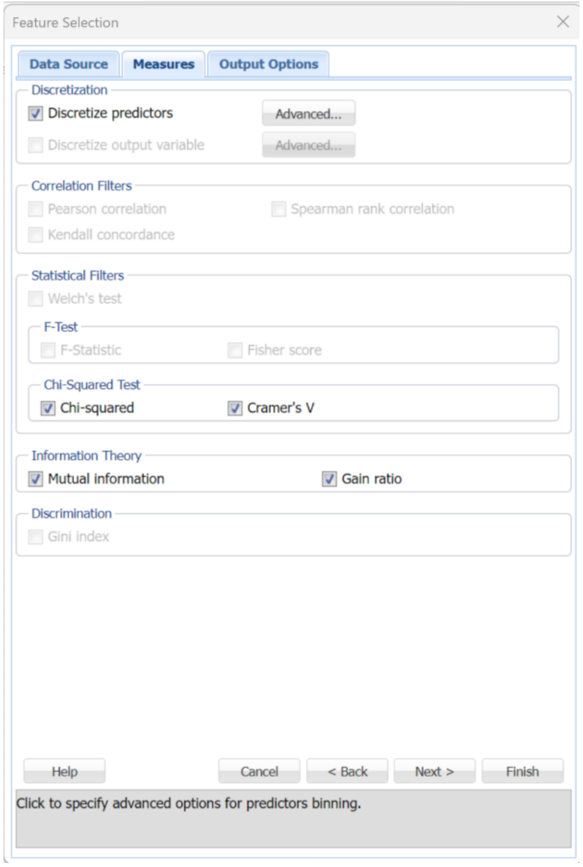
Cramer's V is a variation of the Chi-Squared statistic that also measures the association between two discrete nominal variables. This statistic ranges from 0 to 1 with 0 indicating no association between the two variables and 1 indicating complete association (the two variables are equal).

Select *Mutual information* and *Gain ratio* within the *Information Theory* frame. Mutual information is the degree of a variables' mutual dependence or the amount of uncertainty in variable 1 that can be reduced by incorporating knowledge about variable 2. Mutual Information is non-negative and is equal to zero if the two variables are statistically independent. Also, it is always less than the entropy (amount of information contained) in each individual variable.

The *Gain Ratio*, ranging from 0 and 1, is defined as the mutual information (or information gain) normalized by the feature entropy. This normalization helps address the problem of overemphasizing features with many values but the normalization results in an overestimate of the relevance of features with low entropy. It is a good practice to consider both mutual information and gain ratio for deciding on feature rankings. The larger the gain ratio, the larger the evidence for the feature to be relevant in a classification model.

For more information on the remaining options on this dialog, see the Using Feature Selection found within the Data Science Reference Guide.

Figure 3: Feature Selection Measures dialog



Discretization

☑ Discretize predictors — Advanced...

☐ Discretize output variable — Advanced...

Correlation Filters

☐ Pearson correlation ☐ Spearman rank correlation

☐ Kendall concordance

Statistical Filters

☐ Welch's test

F-Test

☐ F-Statistic ☐ Fisher score

Chi-Squared Test

☑ Chi-squared ☑ Cramer's V

Information Theory

☑ Mutual information ☑ Gain ratio

Discrimination

☐ Gini index

Click to specify advanced options for predictors binning.

Click the **Output Options** tab or click **Next** to open the Output Options dialog. *Table of all produced measures* is selected by default. When this option is selected, Analytic Solver Data Science will produce a report containing all measures selected on the Measures tab.

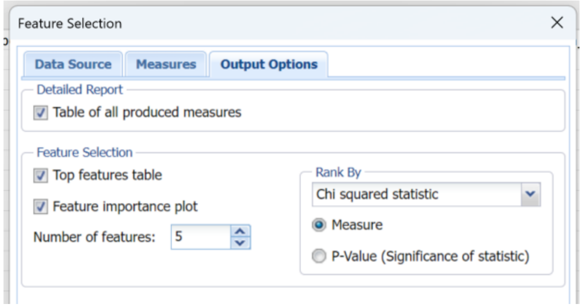
Top Features table is selected by default. This option produces a report containing only top variables as indicated by the Number of features edit box.

Select **Feature importance plot**. This option produces a graphical representation of variable importance based on the measure selected in the *Rank By* drop down menu.

Enter **5** for *Number of features*. Analytic Solver Data Science will display the top 5 most important or most relevant features (variables) as ranked by the statistic displayed in the *Rank By* drop down menu.

Keep *Chi squared statistic* selected for the *Rank By* option. Analytic Solver Data Science will display all measures and rank them by the statistic chosen in this drop down menu.

Figure 4: Feature Selection Output Options dialog



Feature Selection

Detailed Report

☑ Table of all produced measures

Feature Selection

☑ Top features table

☑ Feature importance plot

Number of features: 5

Rank By

Chi squared statistic

● Measure

○ P-Value (Significance of statistic)

Click **Finish**.

Two worksheets are inserted to the right of the heart_failure_clinical_records worksheet: FS_Output and FS_Top_Features.

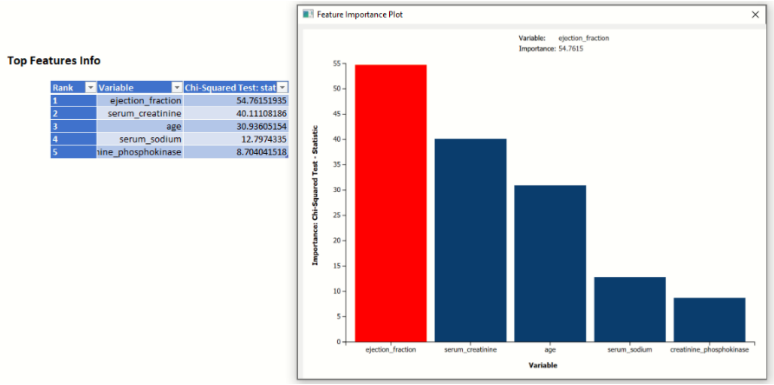
Click the FS_Top_Features tab.

In the Data Science Cloud app, click the Charts icon on the Ribbon to open the Charts dialog, then select **FS_Top_Features** for *Worksheet* and **Feature Importance Chart** for *Chart*.

The Feature Importance Plot ranks the variables by most important or relevant according to the selected measure. In this example, we see that the ejection_fraction, serum_creatinine, age, serum_sodium and creatinine_phosphokinase are the top five most important or relevant variables according to the Chi-Squared statistic. It's beneficial to examine the Feature Selection Importance Plot in order to quickly identify the largest drops or "elbows" in feature relevancy (importance) and select the optimal number of variables for a given classification or regression model.

Note: We could have limited the number of variables displayed on the plot to a specified number of variables (or features) by selecting Number of features and then specifying the number of desired variables. This is useful when the number of input variables is large or we are particularly interested in a specific number of highly – ranked features.

Figure 5: Top Features Plot and Table



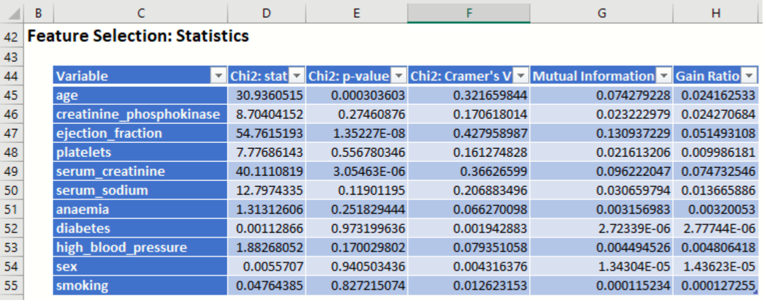
Top Features Info

Rank	Variable	Chi-Squared Test: stat
1	ejection_fraction	54.76151935
2	serum_creatinine	40.11108186
3	age	30.93605154
4	serum_sodium	12.7974335
5	nine_phosphokinase	8.704041518

Run your mouse over each bar in the graph to see the Variable name and Importance factor, in this case Chi-Square, in the top of the dialog.

Click the **X** in the upper right hand corner to close the dialog, then click *FS_Output* tab to open the Feature Selection report.

Figure 6: Feature Selection: Statistics Table



Feature Selection: Statistics

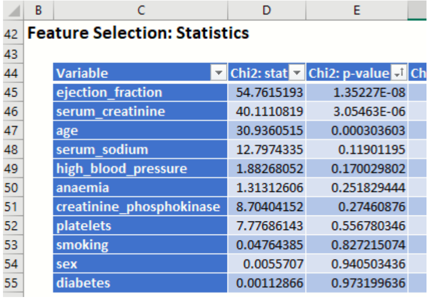
Variable	Chi2: stat	Chi2: p-value	Chi2: Cramer's V	Mutual Information	Gain Ratio
age	30.9360515	0.000303603	0.321659844	0.074279228	0.024162533
creatinine_phosphokinase	8.70404152	0.27460876	0.170618014	0.023222979	0.024270684
ejection_fraction	54.7615193	1.35227E-08	0.427958987	0.130937229	0.051493108
platelets	7.77686143	0.556780346	0.161274828	0.021613206	0.009986181
serum_creatinine	40.1110819	3.05463E-06	0.36626599	0.096222047	0.074732546
serum_sodium	12.7974335	0.11901195	0.206883496	0.030659794	0.013665886
anaemia	1.31312606	0.251829444	0.066270098	0.003156983	0.00320053
diabetes	0.00112866	0.973199636	0.001942883	2.72339E-06	2.77744E-06
high_blood_pressure	1.88268052	0.170029802	0.079351058	0.004494526	0.004806418
sex	0.0055707	0.940503436	0.004316376	1.34304E-05	1.43623E-05
smoking	0.04764385	0.827215074	0.012623153	0.000115234	0.000127255

The Detailed Feature Selection Report displays each computed metric selected on the Measures tab: Chi-squared statistic, Chi-squared P-Value, Cramer's V, Mutual Information, and Gain Ratio.

Chi2: Statistic and p-value

Click the down arrow next to Chi2: p-value to sort the table according to this statistic going from smallest p-value to largest.

Figure 7: Statistics sorted by Chi2:p-value



Feature Selection: Statistics

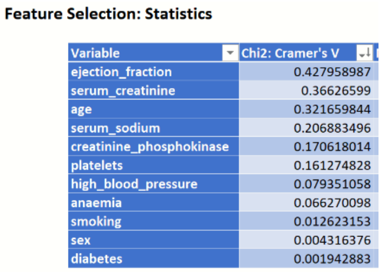
Variable	Chi2: stat	Chi2: p-value
ejection_fraction	54.7615193	1.35227E-08
serum_creatinine	40.1110819	3.05463E-06
age	30.9360515	0.000303603
serum_sodium	12.7974335	0.11901195
high_blood_pressure	1.88268052	0.170029802
anaemia	1.31312606	0.251829444
creatinine_phosphokinase	8.70404152	0.27460876
platelets	7.77686143	0.556780346
smoking	0.04764385	0.827215074
sex	0.0055707	0.940503436
diabetes	0.00112866	0.973199636

According to the Chi-squared test, ejection_fraction, serum_creatinine and age are the 3 most relevant variables for predicting the outcome of a patient in heart failure.

Chi2: Cramer's V

Recall that the Cramer's V statistic ranges from 0 to 1 with 0 indicating no association between the two variables and 1 indicating complete association (the two variables are equal). Sort the Cramer's V statistic from largest to smallest.

Figure 8: Chi2: Cramer's V Statistic



Feature Selection: Statistics

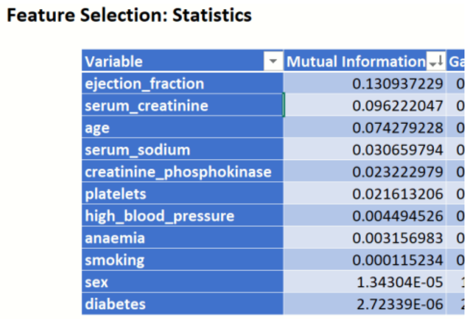
Variable	Chi2: Cramer's V
ejection_fraction	0.427958987
serum_creatinine	0.36626599
age	0.321659844
serum_sodium	0.206883496
creatinine_phosphokinase	0.170618014
platelets	0.161274828
high_blood_pressure	0.079351058
anaemia	0.066270098
smoking	0.012623153
sex	0.004316376
diabetes	0.001942883

Again, this statistic ranks the same four variables, ejection_fraction, serum_creatinine, age and serum_sodium, as the Chi^2 statistic.

Mutual Information

Sort the Mutual Information column by largest to smallest value. This statistic measures how much information the presence/absence of a term contributes to making the correct classification decision.[5] The closer the value to 1, the more contribution the feature provides.

Figure 9: Mutual Information Statistic



Feature Selection: Statistics

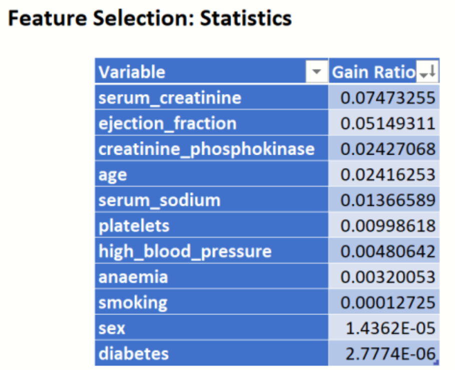
Variable	Mutual Information	Ga
ejection_fraction	0.130937229	0
serum_creatinine	0.096222047	0
age	0.074279228	0
serum_sodium	0.030659794	0
creatinine_phosphokinase	0.023222979	0
platelets	0.021613206	0
high_blood_pressure	0.004494526	0
anaemia	0.003156983	0
smoking	0.000115234	0
sex	1.34304E-05	1
diabetes	2.72339E-06	2

When compared to the Chi2 and Cramer's V statistic, the top four most significant variables calculated for Mututal Information are the same: ejection_fraction, serum_creatinine, age, and serum_sodium.

Gain Ratio

Finally, sort the Gain Ratio from largest to smallest. (Recall that the larger the gain ratio value, the larger the evidence for the feature to be relevant in the classification model.)

Figure 10: Gain Ratio



Feature Selection: Statistics

Variable	Gain Ratio
serum_creatinine	0.07473255
ejection_fraction	0.05149311
creatinine_phosphokinase	0.02427068
age	0.02416253
serum_sodium	0.01366589
platelets	0.00998618
high_blood_pressure	0.00480642
anaemia	0.00320053
smoking	0.00012725
sex	1.4362E-05
diabetes	2.7774E-06

While this statistic's rankings differ from the first 4 statistic's rankings, ejection_fraction, age and serum_creatinine are still ranked in the top four positions.

The Feature Selection tool has allowed us to quickly explore and learn about our data. We now have a pretty good idea of which variables are the most relevant or most important to our classification or prediction model, how our variables relate to each other and to the output variable, and which data attributes would be worth extra time and money in future data collection. Interestingly, for this example, most of our ranking statistics have agreed (mostly) on the most important or relevant features with strong evidence. We computed and

[5] https://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html

examined various metrics and statistics and for some (where p-values can be computed) we’ve seen a statistical evidence that the test of interest succeeded with definitive conclusion. In this example, we’ve observed that several variable (or features) were consistently ranked in the top 3-4 most important variables by most of the measures produced by Analytic Solver Data Science’s Feature Selection tool. However, this will not always be the case. On some datasets you will find that the ranking statistics and metrics compete on rankings. In cases such as these, further analysis may be required.

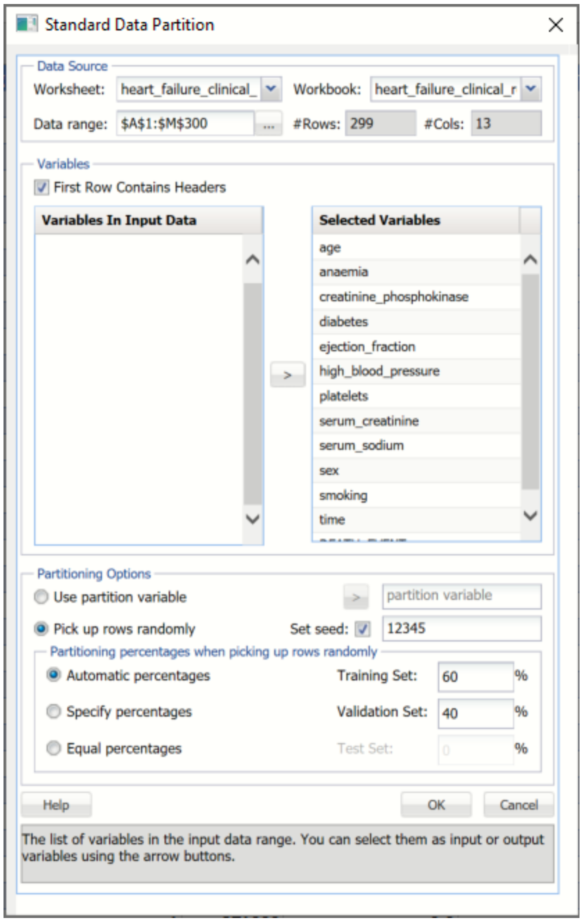
Fitting the Model

To further emulate the results of the journal article discussed in the Feature Selection Example, the Random Trees Ensemble Classification Methods will be used to investigate if a machine learning algorithm can predict a patient's survival using the top two or three ranked features as found by the Feature Selection tool.

First, click **Partition – Standard Partition** to partition the dataset into Training, Validation and Test Sets using the default percentages of 60% allocated to the Training Set and 40% allocate to the Validation Set.

For more information on the remaining options on this dialog, see the Standard Partitioning chapter within the Data Science Reference Guide.

Figure 11: Standard Data Partition dialog



Standard Data Partition

Data Source

Worksheet: heart_failure_clinical_ Workbook: heart_failure_clinical_r

Data range: $A$1:$M$300 #Rows: 299 #Cols: 13

Variables

☑ First Row Contains Headers

Variables In Input Data	Selected Variables
	age
	anaemia
	creatinine_phosphokinase
	diabetes
	ejection_fraction
	high_blood_pressure
	platelets
	serum_creatinine
	serum_sodium
	sex
	smoking
	time

Partitioning Options

○ Use partition variable partition variable

● Pick up rows randomly Set seed: ☑ 12345

Partitioning percentages when picking up rows randomly

● Automatic percentages Training Set: 60 %

○ Specify percentages Validation Set: 40 %

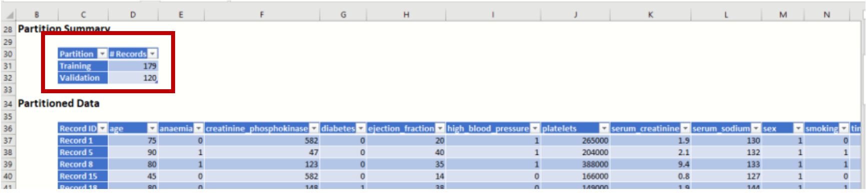
○ Equal percentages Test Set: 0 %

The list of variables in the input data range. You can select them as input or output variables using the arrow buttons.

Then click **OK** to create the two partitions. A new worksheet *STDPartition* is inserted to the right of the dataset. The number of records allocated to the

Training partition is 179 and the number of records allocated to the Validation partition is 120.

Figure 12: Standard Data Partitioning results



Partition Summary

Partition	# Records
Training	179
Validation	120

Partitioned Data

Record ID	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking
Record 1	75	0	582	0	20	1	265000	1.9	130	1	0
Record 5	90	1	47	0	40	1	204000	2.1	132	1	1
Record 8	80	1	123	0	35	1	388000	9.4	133	1	1
Record 15	45	0	582	0	14	0	166000	0.8	127	1	0

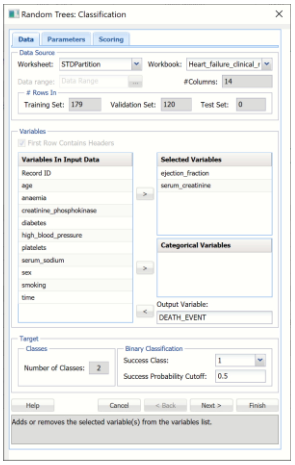
The first time that the model is fit, only two features (ejection_fraction and serum_creatinine) will be utilized.

Click **Classify – Ensemble – Random Trees** to open the *Random Trees: Classification* dialog.

- Select the two Variables from *Variables In Input Data* (ejection_fraction and serum_creatinine) and click the right pointing arrow to the left of *Selected Variables* to add these two variables to the model.   Then take similar steps to select DEATH_EVENT as the *Output Variable*.
- Leave Success Class as "1" and Success Probability Cutoff at 0.5 under Binary Classification.

The Random Trees: Classification dialog should be similar to the one pictured in the Figure 10 below.

Figure 13: Random Trees: Classification dialog with Selection Variables (serum_creatinine and ejection_fraction) and Output Variables (DEATH_EVENT) selected.



Random Trees: Classification

Data Source
Worksheet: STDPartition
Workbook: Heart_failure_clinical_r
Data range: Data Range
#Columns: 14

Rows In
Training Set: 179
Validation Set: 120
Test Set: 0

Variables
☑ First Row Contains Headers

Variables In Input Data: Record ID, age, anaemia, creatinine_phosphokinase, diabetes, high_blood_pressure, platelets, serum_sodium, sex, smoking, time

Selected Variables: ejection_fraction, serum_creatinine

Categorical Variables:

Output Variable: DEATH_EVENT

Target
Classes
Number of Classes: 2
Binary Classification
Success Class: 1
Success Probability Cutoff: 0.5

Adds or removes the selected variable(s) from the variables list.

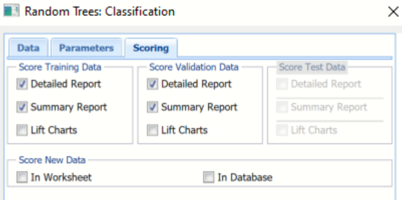
Click the **Scoring tab** to advance to the *Random Trees: Classification Scoring* tab.

For more information on Random Trees parameters, see the Random Trees Classification Options section below.

Summary Report is selected by default. Select **Detailed Report** for both *Score Training Data* and *Score Validation Data* and then click **Finish**.

For more information on the remaining options on any of the Random Trees: Classification dialogs, see the Random Trees: Classification Options section within the Data Science Reference Guide.

Figure 14: Random Trees: Classification dialog with output choices selected



Score Training Data
☑ Detailed Report
☑ Summary Report
□ Lift Charts

Score Validation Data
☑ Detailed Report
☑ Summary Report
□ Lift Charts

Score Test Data
□ Detailed Report
□ Summary Report
□ Lift Charts

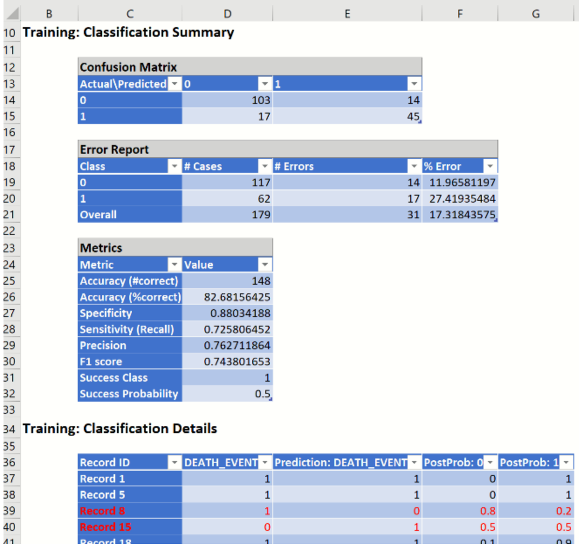
Score New Data
□ In Worksheet
□ In Database

Four worksheets are inserted to the right of the STDPartition tab: *CRandTrees_Output*, *CRandTrees_TrainingScore*, *CRandTrees_ValidationScore* and *CRandTrees_Stored*.

- CRandTrees_Output reports the input data, output data, and parameter settings.
- CRandTrees_TrainingScore reports the confusion matrix, calculated metrics and the actual classification by row for the training partition.
- CRandTrees_ValidationScore reports the confusion matrix, calculated metrics and the actual classification by row for the validation partition.
- CRandTrees_Stored contains the stored model which can be used to apply the fitted model to new data. See the Scoring chapter within the Analytic Solver Data Science User Guide for an example of scoring new data using the stored model.

Click **CRandTrees_TrainingScore** to view the Classification Summary for the Training partition.

Figure 15: Training: Classification Summary



Training: Classification Summary

Confusion Matrix

Actual\Predicted	0	1
0	103	14
1	17	45

Error Report

Class	# Cases	# Errors	% Error
0	117	14	11.96581197
1	62	17	27.41935484
Overall	179	31	17.31843575

Metrics

Metric	Value
Accuracy (#correct)	148
Accuracy (%correct)	82.68156425
Specificity	0.88034188
Sensitivity (Recall)	0.725806452
Precision	0.762711864
F1 score	0.743801653
Success Class	1
Success Probability	0.5

Training: Classification Details

Record ID	DEATH_EVENT	Prediction: DEATH_EVENT	PostProb: 0	PostProb: 1
Record 1	1	1	0	1
Record 5	1	1	0	1
Record 8	1	0	0.8	0.2
Record 15	0	1	0.5	0.5
Record 18	1	1	0.1	0.9

The overall error for the training partition was 17.32% with 14 surviving patients reported as deceased and 17 deceased patients reported as survivors.

- Accuracy: 82.68% -- Accuracy refers to the ability of the classifier to predict a class label correctly.
- Specificity: 0.88 – (True Negative)/(True Negative + False Positives)

  Specificity is defined as the proportion of negative classifications that were actually negative, or the fraction of survivors that actually survived. In this model, 103 actual surviving patients were classified correctly as survivors.

There were 14 false positives or 14 actual survivors classified incorrectly as deceased.

- Sensitivity or Recall: 0.726 – (True Positive)/(True Positive + False Negative)

  Sensitivity is defined as the proportion of positive cases there were classified correctly as positive, or the proportion of actually deceased patients there were classified as deceased. In this model, 45 actual deceased patients were correctly classified as deceased. There were 17 false negatives or 17 actual deceased patients were incorrectly classified as survivors.

  Note: Since the object of this model is to correctly classify which patients will succumb to heart failure, this is an important statistic as it is very important for a physician to be able to accurately predict which patients require mitigation.

- Precision: 0.763 – (True Positives)/(True Positives + False Positives)

  Precision is defined as the proportion of positive results that are true positive. In this model, 45 actual deceased patients were classified correctly as deceased. There were 14 false positives or14 actual survivors classified incorrectly as deceased.

- F-1 Score: 0.743 –2 x (Precision * Sensitivity)/(Precision + Sensitivity)

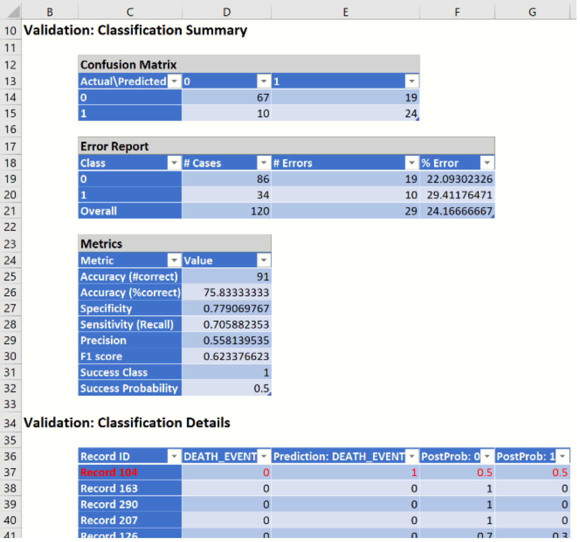
  The F-1 Score provides a statistic to balance between Precision and Sensitivity, especially if an uneven class distribution exists, as in this example, (103 survivors vs 45 deceased). The closer the F-1 score is to 1 (the upper bound) the better the precision and recall.

- Success Class and Success Probability simply reports the settings for these two values as input on the Random Trees: Classification, Data tab.

- View individual records and their classifications beneath Training: Classification Details.

Click **the CRandTrees_ValidationScore** tab to view the Summary Results for the Validation partition.

Figure 16: Validation: Classification Summery



Validation: Classification Summary

Confusion Matrix

Actual\Predicted	0	1
0	67	19
1	10	24

Error Report

Class	# Cases	# Errors	% Error
0	86	19	22.09302326
1	34	10	29.41176471
Overall	120	29	24.16666667

Metrics

Metric	Value
Accuracy (#correct)	91
Accuracy (%correct)	75.83333333
Specificity	0.779069767
Sensitivity (Recall)	0.705882353
Precision	0.558139535
F1 score	0.623376623
Success Class	1
Success Probability	0.5

Validation: Classification Details

Record ID	DEATH_EVENT	Prediction: DEATH_EVENT	PostProb: 0	PostProb: 1
Record 104	0	1	0.5	0.5
Record 163	0	0	1	0
Record 290	0	0	1	0
Record 207	0	0	1	0
Record 126	0	0	0.7	0.3

The overall error for the validation partition was 24.17 with 19 false positives (surviving patients reported as deceased) and 10 false negatives (deceased patients reported as survivors).

Note the following metrics:

- Accuracy: 75.83
- Specificity: 0.779
- Sensitivity or Recall: 0.706
- Precision: 0.558
- F1 Score: 0.623

These steps were performed multiple times while adding additional Selected Variables according to the variable's importance or significance found by Feature Selection. The results are summarized in the table below.

The lowest Overall Error in the Validation Partition for any of the variable combinations occurs when just two variables, ejection_fraction and serum_creatinine, are present in the fitted model. In addition, this fitted model also exhibits the highest Accuracy, Sensitivity, Precision and F1 Score metrics. These results suggest that by obtaining these two measurements for a patient, a physician can determine whether the patient should undergo some type of mitigation for their heart failure diagnosis.

	Training Partition						Validation Partition					
Variables	Overall Error	Accuracy (% Correct)	Specificity	Sensitivity (Recall)	Precision	F1 Score	Overall Error	Accuracy	Specificity	Sensitivity (Recall)	Precision	F1 Score
ejection_fraction, serum_creatinine	17.318	82.682	0.880	0.726	0.763	0.744	24.167	75.833	0.779	0.706	0.558	0.623
+ age	3.352	96.648	0.966	0.968	0.938	0.952	30.000	70.000	0.744	0.588	0.476	0.526
+ serum_sodium	3.911	96.089	0.966	0.952	0.937	0.944	28.333	71.667	0.767	0.588	0.500	0.541
+ high_blood_pressure (added as categorical variable)	2.793	97.207	0.966	0.984	0.938	0.961	29.167	70.833	0.756	0.588	0.488	0.533
+ anaemia (added as categorical variable)	5.587	94.413	0.966	0.903	0.933	0.918	27.500	72.500	0.814	0.500	0.515	0.507
+ creatinine_phosphokinase	2.235	97.765	0.974	0.984	0.953	0.968	30.833	69.167	0.744	0.559	0.463	0.507
+ platelets	2.235	97.765	0.974	0.984	0.953	0.968	30.833	69.167	0.744	0.559	0.463	0.507
+ smoking (added as categorical variable)	3.352	96.648	0.983	0.935	0.967	0.951	29.167	70.833	0.756	0.588	0.488	0.533
+ sex (added as categorical variable)	3.352	96.648	0.974	0.952	0.952	0.952	29.167	70.833	0.791	0.500	0.486	0.493
+ diabetes (added as categorical variable)	3.911	96.089	0.974	0.935	0.951	0.943	30.833	69.167	0.767	0.500	0.459	0.479

Text Mining

Introduction

Text mining is the practice of automated analysis of one document or a collection of documents (corpus) and extracting non-trivial information from it. Also, Text Mining usually involves the process of transforming unstructured textual data into structured representation by analyzing the patterns derived from text. The results can be analyzed to discover interesting knowledge, some of which would only be found by a human carefully reading and analyzing the text. Typical widely-used tasks of Text Mining include but are not limited to Automatic Text Classification/Categorization, Topic Extraction, Concept Extraction, Documents/Terms Clustering, Sentiment Analysis, Frequency-based Analysis and many more. Some of these tasks could not be completed by a human, which makes Text Mining a particularly useful and applicable tool in modern Data Science. Analytic Solver Data Science's Text Miner takes an integrated approach to text mining as it does not totally separate analysis of unstructured data from traditional data science techniques applicable for structured information. While Analytic Solver Data Science is a very powerful tool for analyzing text only, it also offers automated treatment of mixed data, i.e. combination of multiple unstructured and structured fields. This is a particularly useful feature that has many real-world applications, such as analyzing maintenance reports, evaluation forms, insurance claims, etc. Text Miner uses the "bag of words" model – the simplified representation of text, where the precise grammatical structure of text and exact word order is disregarded. Instead, syntactic, frequency-based information is preserved and is used for text representation. Although such assumptions might be harmful for some specific applications of Natural Language Processing (NLP), it has been proven to work very well for applications such as Text Categorization, Concept Extraction and others, which are the particular areas addressed by Text Miner's capabilities. It has been shown in many theoretical/empirical studies that syntactic similarity often implies semantic similarity. One way to access syntactic relationships is to represent text in terms of Generalized Vector Space Model (GVSP). Advantage of such representation is a meaningful mapping of text to the numeric space, the disadvantage is that some semantic elements, e.g. order of words, are lost (recall the bag-of-words assumption).

Input to Text Miner (the Text Mining tool within Analytic Solver Data Science) could be of two main types – few relatively large documents (e.g. several books) or relatively large number of smaller documents (e.g. collection of emails, news articles, product reviews, comments, tweets, Facebook posts, etc.). While Text Miner is capable of analyzing large text documents, it is particularly effective for large corpuses of relatively small documents. Obviously, this functionality has limitless number of applications – for instance, email spam detection, topic extraction in articles, automatic rerouting of correspondence, sentiment analysis of product reviews and many more.

The input for text mining is a dataset on a worksheet, with at least one column that contains free-form text (or file paths to documents in a file system containing free-form text), and, optionally, other columns that contain traditional structured data. In the first tab of the Text Mining dialog, the user selects the text variable(s), and the other variable(s) to be processed.

The output for the text mining is a set of reports that contain general explorative information about the collection of documents and structured representations of text (free-form text columns are expanded to a set of new columns with numeric representation. The new columns will each correspond to either (i) a single term (word) found in the "corpus" of documents, or, if requested, (ii) a concept extracted from the corpus through Latent Semantic Indexing (LSI, also called LSA or Latent Semantic Analysis). Each concept represents an automatically derived complex combination of terms/words that have been identified to be related to a particular topic in the corpus of documents. The structural representation of text can serve as an input to any traditional data science techniques available in Text Miner – unsupervised/supervised, affinity, visualization techniques, etc. In addition, Text Miner also presents a visual representation of Text Mining results to allow the user to interactively explore the information, which otherwise would be extremely hard to analyze manually. Typical visualizations that aid in understanding of Text Mining outputs and that are produced by Text Miner are:

- Zipf plot – for visual/interactive exploration of frequency-based information extracted by Text Miner

- Scree Plot, Term-Concept and Document-Concept 2D scatter plots – for visual/interactive exploration of Concept Extraction results

If you are interested in visualizing specific parts of Text Mining analysis outputs, Text Miner provides rich capabilities for charting – the functionality that can be used to explore Text Mining results and supplement standard charts discussed above.

In the example below, you will learn how to use Text Miner in Analytic Solver Data Science to process/analyze approximately 1000 text files and use the results for automatic topic categorization. This will be achieved by using structured representation of text presented to Logistic Regression for building the model for classification.

Text Mining Example

This example uses the text files within the Text Mining Example Documents.zip archive file to illustrate how to use Analytic Solver Data Science's Text Mining tool. These documents were selected from the well-known text dataset (downloadable from http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/news20.html) which consists of 20,000 messages, collected from 20 different internet newsgroups. We selected about 1,200 of these messages that were posted to two interest groups, Autos and Electronics (about 500 documents from each).
Note: Analytic Solver Cloud does not currently support importing from a file folder.

Importing from a File Folder

The Text Mining Example Documents.zip archive file is located at C:\ProgramData\Frontline Systems\Datasets. Unzip the contents of this file to a location of your choice. Four folders will be created beneath Text Mining Example Documents: Autos, Electronics, Additional Autos and Additional Electronics. One thousand, two hundred short text files will be extracted to the location chosen. This example is based on the text dataset at http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/news20.html, which consists of 20,000 messages, collected from 20 different netnews newsgroups.

We selected about 1,200 of these messages that were posted to two interest groups, for Autos and Electronics (about 50% in each).

Select **Get Data – File Folder** to open the *Import From File System* dialog.

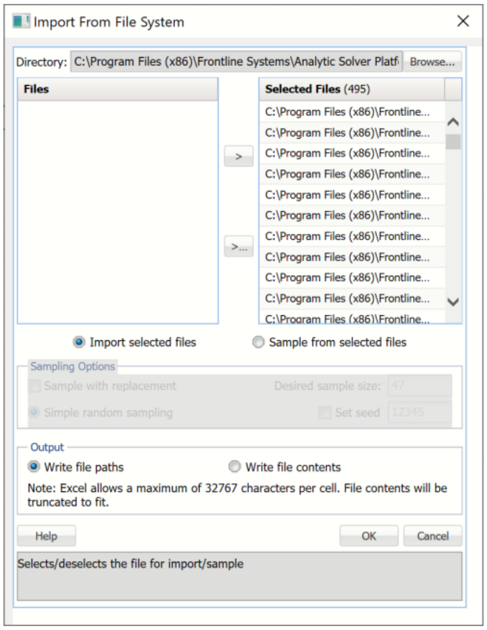
At the top of the dialog, click **Browse…** to navigate to the *Autos* subfolder (C:\ProgramData\Frontline Systems\Datasets\Text Mining Example Documents\Autos). Set the *File Type* to All Files (*.*), then select all files in the folder and click the **Open** button. The files will appear in the left list box under *Files*. Click the >> button to move the files from the *Files* list box to the *Selected Files* list box. Now repeat these steps for the Electronics subfolder. When these steps are completed, 985 files will appear under *Selected Files*.

Select **Sample from selected files** to enable the *Sampling Options*. Text Miner will perform sampling from the files in the *Selected Files* field. Enter **300** for *Desired sample size* while leaving the default settings for *Simple random sampling* and *Set Seed*.

Note: If you are using the educational version of Analytic Solver Data Science, enter "100" for Desired Sample Size. This is the upper limit for the number of files supported when sampling from a file system when using Analytic Solver Data Science. For a complete list of the capabilities of Analytic Solver Data Science and Analytic Solver Data Science for Education, click here.

Text Miner will select 300 files using Simple random sampling with a seed value of 12345. Under *Output*, leave the default setting of *Write file paths*. Rather than writing out the file contents into the report, Text Miner will include the file paths.

Note: Currently, Analytic Solver Data Science only supports the import of delimited text files. A delimited text file is one in which data values are separated by a character such as quotation marks, commas or tabs. These characters define a beginning and end of a string of text.


Import From File System

Directory: C:\Program Files (x86)\Frontline Systems\Analytic Solver Platf

Files

Selected Files (495)

C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...
C:\Program Files (x86)\Frontline...

● Import selected files ○ Sample from selected files

Sampling Options

□ Sample with replacement Desired sample size: 47

● Simple random sampling □ Set seed 12345

Output

● Write file paths ○ Write file contents

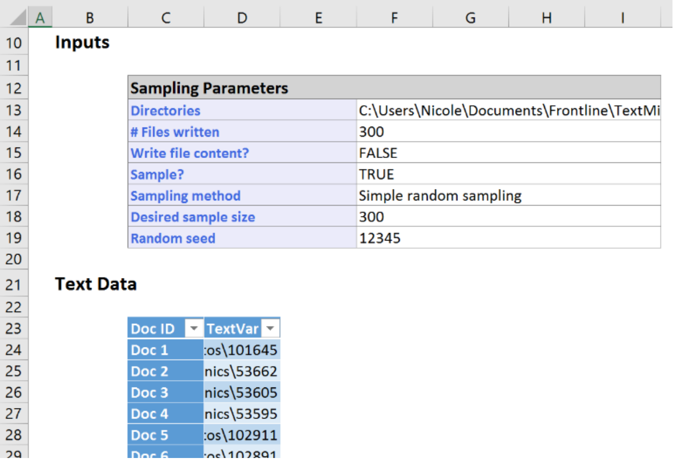
Note: Excel allows a maximum of 32767 characters per cell. File contents will be truncated to fit.

Selects/deselects the file for import/sample

Click **OK**. The output *XLM_SampleFiles* will be inserted into the Data Science task pane, with contents similar to that shown on the next page.

The *Data* portion of the report displays the selections we made on the *Import From File System* dialog. Here we see the path of the directories, the number of files written, our choice to write the paths or contents (*File Paths*), the sampling method, the desired sample size, the actual size of the sample, and the seed value (*12345*).

Underneath the *Data* portion are paths to the 300 text files in random order that were sampled by Analytic Solver Data Science. If *Write file contents* had been selected, rather than *Write file paths*, the report would contain the RowID, File Path, and the first 32,767 characters present in the document.



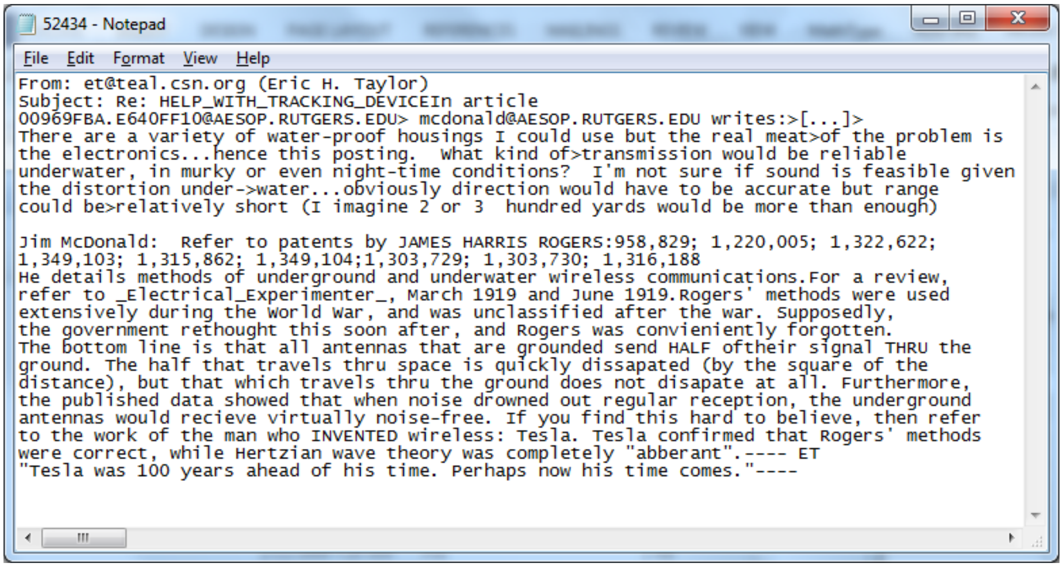
Inputs

Sampling Parameters	
Directories	C:\Users\Nicole\Documents\Frontline\TextMi
# Files written	300
Write file content?	FALSE
Sample?	TRUE
Sampling method	Simple random sampling
Desired sample size	300
Random seed	12345

Text Data

Doc ID	TextVar
Doc 1	:os\101645
Doc 2	nics\53662
Doc 3	nics\53605
Doc 4	nics\53595
Doc 5	:os\102911
Doc 6	:os\102891

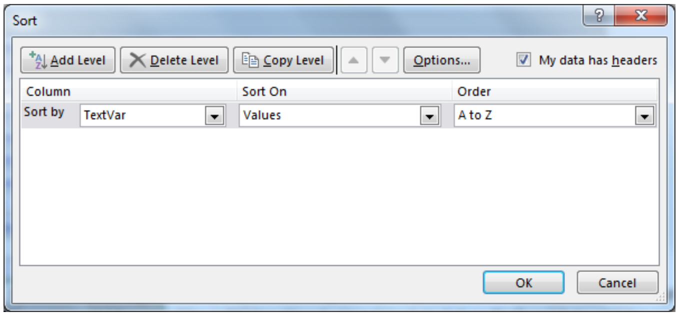
Here is an example of a document that appeared in the Electronics newsgroup. Note the appearance of email addresses, “From” and “Subject” lines. All three appear in each document.



```
From: et@teal.csn.org (Eric H. Taylor)
Subject: Re: HELP_WITH_TRACKING_DEVICEIn article
00969FBA.E640FF10@AESOP.RUTGERS.EDU> mcdonald@AESOP.RUTGERS.EDU writes:>[...]>
There are a variety of water-proof housings I could use but the real meat>of the problem is
the electronics...hence this posting.  What kind of>transmission would be reliable
underwater, in murky or even night-time conditions?  I'm not sure if sound is feasible given
the distortion under->water...obviously direction would have to be accurate but range
could be>relatively short (I imagine 2 or 3  hundred yards would be more than enough)

Jim McDonald:  Refer to patents by JAMES HARRIS ROGERS:958,829; 1,220,005; 1,322,622;
1,349,103; 1,315,862; 1,349,104;1,303,729; 1,303,730; 1,316,188
He details methods of underground and underwater wireless communications.For a review,
refer to _Electrical_Experimenter_, March 1919 and June 1919.Rogers' methods were used
extensively during the World War, and was unclassified after the war. Supposedly,
the government rethought this soon after, and Rogers was convieniently forgotten.
The bottom line is that all antennas that are grounded send HALF oftheir signal THRU the
ground. The half that travels thru space is quickly dissapated (by the square of the
distance), but that which travels thru the ground does not disapate at all. Furthermore,
the published data showed that when noise drowned out regular reception, the underground
antennas would recieve virtually noise-free. If you find this hard to believe, then refer
to the work of the man who INVENTED wireless: Tesla. Tesla confirmed that Rogers' methods
were correct, while Hertzian wave theory was completely "abberant".---- ET
"Tesla was 100 years ahead of his time. Perhaps now his time comes."----
```

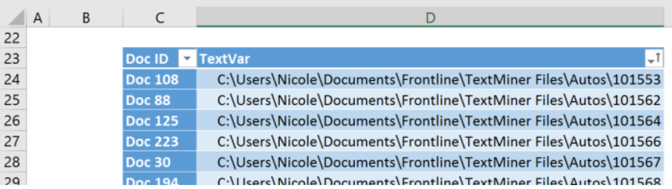
The selected file paths are now in random order, but we will need to categorize the “Autos” and “Electronics” files in order to be able to identify them later. To do this, we’ll use Excel to sort the rows by the file path: Select columns B through D and rows 18 through 317, then choose **Sort** from the Data tab. In the Sort dialog, select column d, where the file paths are located, and click OK.



☑ My data has headers

Column	Sort On	Order
Sort by TextVar	Values	A to Z

The file paths should now be sorted between *Electronics* and *Autos* files.



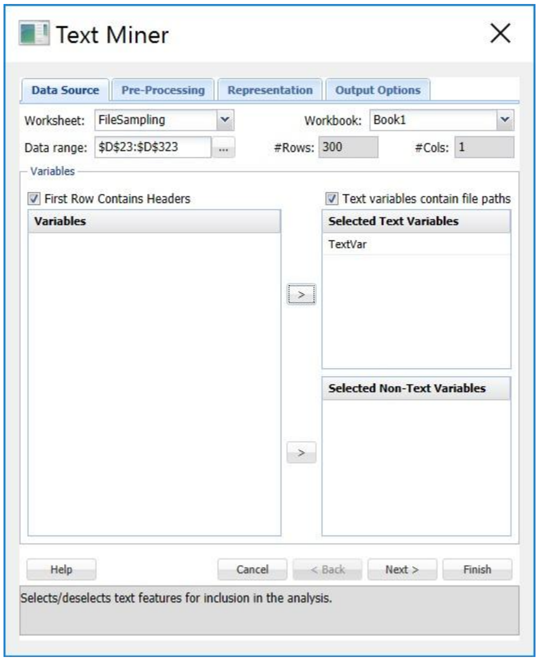
	A	B	C	D
22				
23			Doc ID	TextVar
24			Doc 108	C:\Users\Nicole\Documents\Frontline\TextMiner Files\Autos\101553
25			Doc 88	C:\Users\Nicole\Documents\Frontline\TextMiner Files\Autos\101562
26			Doc 125	C:\Users\Nicole\Documents\Frontline\TextMiner Files\Autos\101564
27			Doc 223	C:\Users\Nicole\Documents\Frontline\TextMiner Files\Autos\101566
28			Doc 30	C:\Users\Nicole\Documents\Frontline\TextMiner Files\Autos\101567
29			Doc 194	C:\Users\Nicole\Documents\Frontline\TextMiner Files\Autos\101568

Using Text Miner

Click the **Text** icon to bring up the *Text Miner* dialog.

Data Source Tab

Confirm that **XLM_SampleFiles** is selected for *Worksheet*.   Select **TextVar** in the *Variables* list box, and click the upper > button to move it to the *Selected Text Variables* list box. By doing so, we are selecting the text in the documents as input to the Text Miner model. Ensure that "Text variables contain file paths" is **checked**.



Worksheet: FileSampling   Workbook: Book1

Data range: $D$23:$D$323   #Rows: 300   #Cols: 1

☑ First Row Contains Headers

☑ Text variables contain file paths

Selected Text Variables: TextVar

Selected Non-Text Variables

Selects/deselects text features for inclusion in the analysis.

Click the **Next** button, or click the **Pre-Processing** tab at the top. b

Pre-Processing Tab

Leave the default setting for *Analyze all terms* selected under *Mode*. When this option is selected, Text Miner will examine all terms in the document. A “term” is defined as an individual entity in the text, which may or may not be an English word. A term can be a word, number, email, url, etc. terms are separated by all possible delimiting characters (i.e. \, ?, ', `, ~, |, \r, \n, \t, :, !, @, #, $, %, ^, &, *, (,), [,], {, }, <>,_, ;, =, -, +, \) with some exceptions related to stopwords, synonyms, exclusion terms and boilerplate normalization (URLs, emails, monetary amounts, etc.). Text Miner will not tokenize on these delimiters.

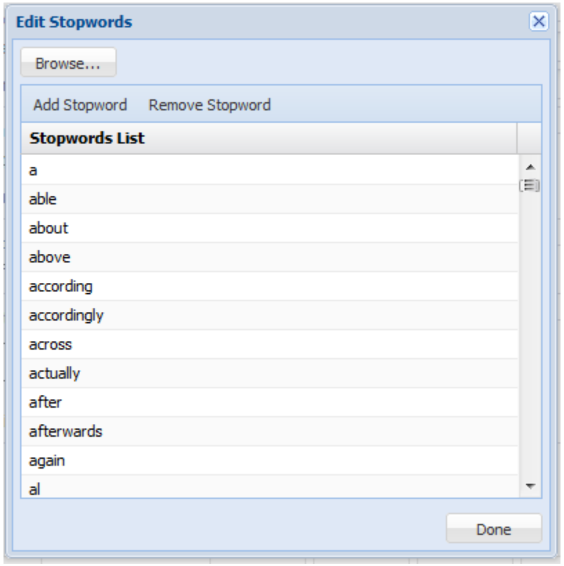
Note: Exceptions are related not to how terms are separated but as to whether they are split based on the delimiter. For example: URL's contain many characters such as "/", ";", etc. Text Miner will not tokenize on these characters in the URL but will consider the URL as a whole and will remove the URL if selected for removal. (See below for more information.)

If *Analyze specified terms only* is selected, the *Edit Terms* button will be enabled. If you click this button, the *Edit Exclusive Terms* dialog opens. Here you can add and remove terms to be considered for text mining. All other terms will be disregarded. For example, if we wanted to mine each document for a specific part name such as “alternator” we would click *Add Term* on *the Edit Exclusive Terms* dialog, then replace “New term” with “alternator” and click *Done* to return to the Pre-Processing dialog. During the text mining process, Text Miner would analyze each document for the term “alternator”, excluding all other terms.

Leave both Start term/phrase and End term/phrase empty under Text Location. If this option is used, text appearing before the first occurrence of the Start Phrase will be disregarded and similarly, text appearing after End Phrase (if used) will be disregarded. For example, if text mining the transcripts from a Live Chat service, you would not be particularly interested in any text appearing before the heading “Chat Transcript” or after the heading “End of Chat Transcript”. Thus you would enter “Chat Transcript” into the Start Phrase field and “End of Chat Transcript” into the End Phrase field.

Leave the default setting for *Stopword removal*. Click Edit to view a list of commonly used words that will be removed from the documents during pre-processing. To remove a word from the Stopword list, simply highlight the desired word, then click **Remove Stopword.** To add a new word to the list, click **Add Stopword**, a new term “stopword” will be added. Double click to edit.

Text Miner also allows additional stopwords to be added or existing to be removed via a text document (*.txt) by using the Browse button to navigate to the file. Terms in the text document can be separated by a space, a comma, or both. If we were supplying our three terms in a text document, rather than in the *Edit Stopwords* dialog, the terms could be listed as: subject emailterm from or subject,emailterm,from or subject, emailterm, from. If we had a large list of additional stopwords, this would be the preferred way to enter the terms.

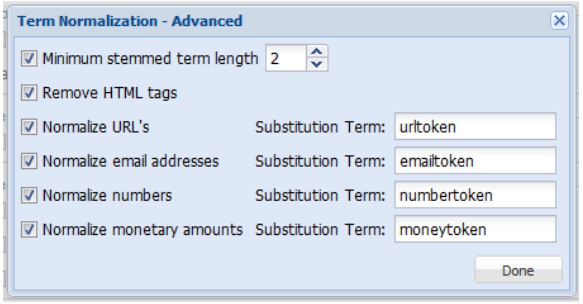


Stopwords List

a
able
about
above
according
accordingly
across
actually
after
afterwards
again
al

Click **Advanced** in the *Term Normalization* group to open the *Term Normalization – Advanced* dialog. **Select all options** as shown below. Then click **Done**. This dialog allows us to indicate to Text Miner, that

- If stemming reduced term length to 2 or less characters, disregard the term (*Minimum stemmed term length).*
- HTML tags, and the text enclosed, will be removed entirely. HTML tags and text contained inside these tags often contain technical, computer-generated information that is not typically relevant to the goal of the text mining application.
- URLs will be replaced with the term, "urltoken". Specific form of URLs do not normally add any meaning, but it is sometimes interesting to know how many URLs are included in a document.
- Email addresses will be replaced with the term, "emailtoken". Since the documents in our collection all contain a great many email addresses (and the distinction between the different emails often has little use in Text Mining), these email addresses will be replaced with the term "emailtoken".
- Numbers will be replaced with the term, "numbertoken".
- Monetary amounts will be substituted with the term, "moneytoken".



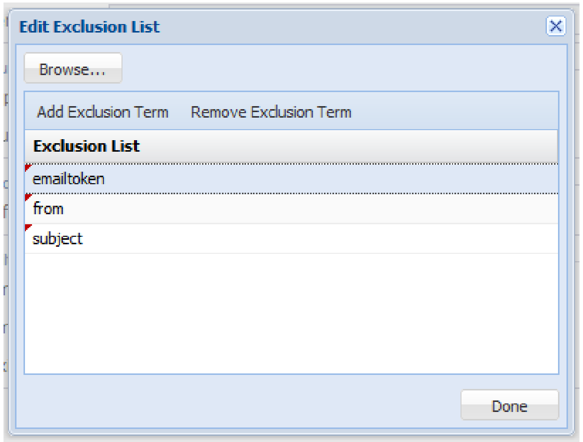
☑ Minimum stemmed term length 2
☑ Remove HTML tags
☑ Normalize URL's Substitution Term: urltoken
☑ Normalize email addresses Substitution Term: emailtoken
☑ Normalize numbers Substitution Term: numbertoken
☑ Normalize monetary amounts Substitution Term: moneytoken

Recall that when we inspected an email from the document collection we saw several terms such as "subject", "from" and email addresses. Since all of our documents contain these terms, including them in the analysis will not provide any benefit and could bias the analysis. As a result, we will exclude these terms from all documents by selecting **Exclusion list** then clicking **Edit**. The *Edit Exclusion List* dialog opens. Click **Add Exclusion Term**. The label "exclusionterm" is added. Click to edit and change to "subject". Then repeat these same steps to add the term "from".

We can take the email issue one step further and completely remove the term "emailtoken" from the collection. Click **Add Exclusion Term** and edit "exclusionterm" to "emailtoken".

To remove a term from the exclusion list, highlight the term and click *Remove Exclusion Term*.

We could have also entered these terms into a text document (*.txt) and added the terms all at once by using the Browse button to navigate to the file and import the list. Terms in the text document can be separated by a space, a comma, or both. If, for example we were supplying excluded terms in a document rather than in the Edit Exclusion List dialog, we would enter the terms as: subject emailtoken from, or subject,emailtoken,from, or subject, emailtoken, from. If we had a large list of terms to be excluded, this would be the preferred way to enter the terms.



Exclusion List
emailtoken
from
subject

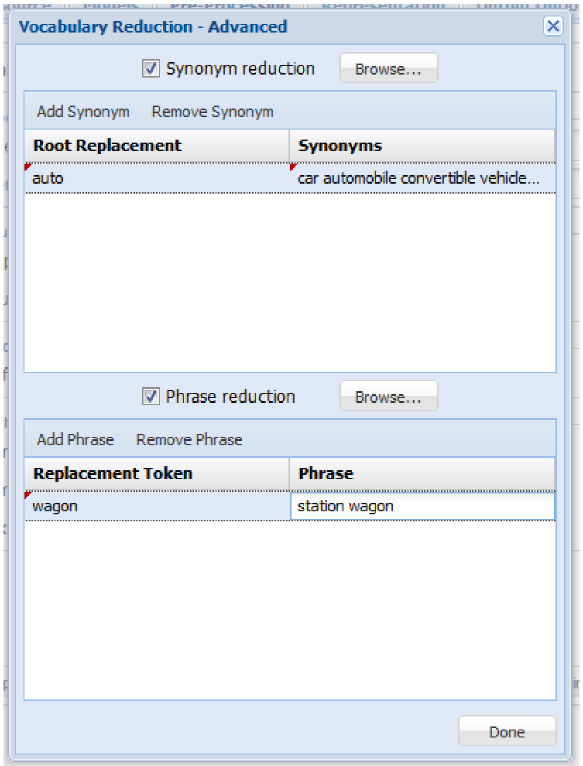
Click **Done** to close the dialog and return to Pre-Processing.

Text Miner also allows the combining of synonyms and full phrases by clicking **Advanced** within *Vocabulary Reduction*. Select **Synonym reduction** at the top of the dialog to replace synonyms such as "car", "automobile", "convertible", "vehicle", "sedan", "coupe", "subcompact", and "jeep" with "auto". Click **Add Synonym** and replace "rootterm" with "auto" then replace "synonym list" with "car, automobile, convertible, vehicle, sedan, coupe, subcompact, jeep" (without the quotes). During pre-processing, Text Miner will replace the terms "car", "automobile", "convertible", "vehicle", "sedan", "coupe", "subcompact" and "jeep" with the term "auto". To remove a synonym from the list, highlight the term and click *Remove Synonym*.

If adding synonyms from a text file, each line must be of the form rootterm:synonymlist or using our example: auto:car automobile convertible vehicle sedan coup or auto:car,automobile,convertible,vehicle,sedan,coup. Note separation between the terms in the synonym list be either a space, a comma or both. If we had a large list of synonyms, this would be the preferred way to enter the terms.

Text Miner also allows the combining of words into phrases that indicate a singular meaning such as "station wagon" which refers to a specific type of car rather than two distinct tokens – station and wagon. To add a phrase in the *Vocabulary Reduction – Advanced* dialog, select **Phrase reduction** and click

Add Phrase. The term “phrasetoken” will be appear, click to edit and enter “wagon”. Click “phrase” to edit and enter “station wagon”. If supplying phrases through a text file (*.txt), each line of the file must be of the form phrasetoken:phrase or using our example, wagon:station wagon. If we had a large list of phrases, this would be the preferred way to enter the terms.



☑ Synonym reduction

Root Replacement	Synonyms
auto	car automobile convertible vehicle...

☑ Phrase reduction

Replacement Token	Phrase
wagon	station wagon

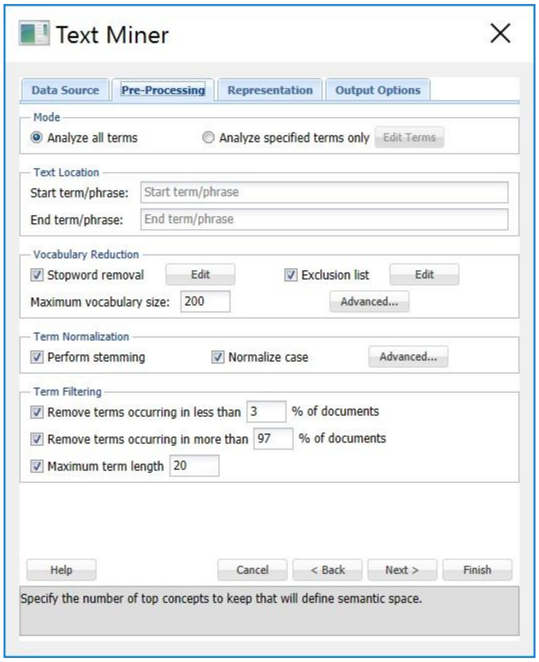
Enter **200** for *Maximum Vocabulary Size*. Text Miner will reduce the number of terms in the final vocabulary to the top 200 most frequently occurring in the collection.

Leave *Perform stemming* at the selected default. Stemming is the practice of stripping words down to their “stems” or “roots”, for example, stemming terms such as “argue”, “argued”, “argues”, “arguing”, and “argus” would result in the stem “argu. However “argument” and “arguments” would stem to “argument”. The stemming algorithm utilized in Text Miner is “smart” in the sense that while “running” would be stemmed to “run”, “runner” would not. . Text Miner uses the Porter Stemmer 2 algorithm for the English Language. For more information on this algorithm, please see the Webpage: http://tartarus.org/martin/PorterStemmer/

Leave the default selection for *Normalize case.* When this option is checked, Text Miner converts all text to a consistent (lower) case, so that Term, term, TERM, etc. are all normalized to a single token “term” before any processing, rather than creating three independent tokens with different case. This simple method can dramatically affect the frequency distributions of the corpus, leading to biased results.

Enter **3** for *Remove terms occurring in less than _% of documents* and **97** for *Remove terms occurring in more than _% of documents*. For many text mining applications, the goal is to identify terms that are useful for discriminating between documents. If a particular term occurs in all or almost all documents, it may not be possible to highlight the differences. If a term occurs in very few documents, it will often indicate great specificity of this term, which is not very useful for some Text Mining purposes.

Enter **20** for *Maximum term length.* Terms that contain more than 20 characters will be excluded from the text mining analysis and will not be present in the final reports. This option can be extremely useful for removing some parts of text which are not actual English words, for example, URLs or computer-generated tokens, or to exclude very rare terms such as Latin species or disease names, i.e. Pneumonoultramicroscopicsilicovolcanoconiosis.



Mode

● Analyze all terms ○ Analyze specified terms only

Text Location

Start term/phrase: Start term/phrase

End term/phrase: End term/phrase

Vocabulary Reduction

☑ Stopword removal ☑ Exclusion list

Maximum vocabulary size: 200

Term Normalization

☑ Perform stemming ☑ Normalize case

Term Filtering

☑ Remove terms occurring in less than 3 % of documents

☑ Remove terms occurring in more than 97 % of documents

☑ Maximum term length 20

Specify the number of top concepts to keep that will define semantic space.

Click **Next** to advance to *the Representation* tab or simply click **Representation** at the top.

Representation Tab

Keep the default selection of *TF-IDF* (Term Frequency – Inverse Document Frequency) for *Term-Document Matrix Scheme*. A term-document matrix is a matrix that displays the frequency-based information of terms occurring in a document or collection of documents. Each column is assigned a term and each row a document. If a term appears in a document, a weight is placed in the corresponding column indicating the term's importance or contribution. Text Miner offers four different commonly used methods of weighting scheme used to represent each value in the matrix: Presence/Absence, Term Frequency, TF-IDF (the default) and Scaled term frequency. If *Presence/Absence* is selected, Text Miner will enter a 1 in the corresponding row/column if the term appears in the document and 0 otherwise. This matrix scheme does not take into account the number of times the term occurs in each document. If *Term Frequency* is selected, Text Miner will count the number of times the term appears in the document and enter this value into the corresponding row/column in the matrix. The default setting – Term Frequency – Inverse Document Frequency (*TF-IDF)* is the product of scaled term frequency and inverse document frequency. Inverse document frequency is calculated by taking the logarithm of the total number of documents divided by the number of documents that contain the term. A high value for TF-IDF indicates that a term that does not occur frequently in the collection of documents taken as a whole, appears quite frequently in the specified document. A TF-IDF value close to 0 indicates that the term appears frequently in the collection or rarely in a specific document. If *Scaled term frequency* is selected, Text Miner will normalize (bring to the same scale) the number of occurrences of a term in the documents (see the table below).

It's also possible to create your own scheme by clicking the *Advanced* command button to open the *Term Document Matrix – Advanced* dialog. Here users can select their own choices for local weighting, global weighting, and normalization. Please see the table below for definitions regarding options for Term Frequency, Document Frequency and Normalization.

Local Weighting		**Global Weighting**		**Normalization**	
Binary	$lw_{td} = \begin{cases} 1, \text{if } tf_{td} > 0 \\ 0, \text{if } tf_{td} = 0 \end{cases}$	None	$gw_t = 1$	None	$n_d = 1$
Raw Frequency	$lw_{td} = tf_{td}$	Inverse	$gw_t = \log_2 \frac{N}{1 + df_t}$	Cosine	$n_d = \frac{1}{\lVert \overline{g_d} \rVert_2}$
Logarithmic	$lw_{td} = \log(1 + tf_{td})$	Normal	$gw_t = \frac{1}{\sqrt{\sum_d tf_{td}^2}}$		
Augnorm	$lw_{td} = \frac{\left(\frac{tf_{td}}{\max\limits_t tf_{td}}\right) + 1}{2}$	GF-IDF	$gw_t = \frac{cf_t}{df_t}$		
		Entropy	$gw_t = 1 + \sum_d \frac{p_{td} \log p_{td}}{\log N}$		
		IDF probability	$gw_t = \log_2 \frac{N}{1 + df_t}$		

Notations:

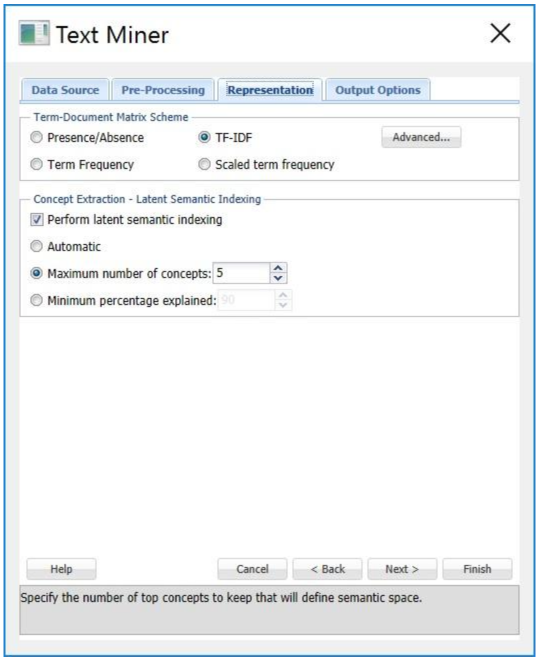
- tf_{td} – frequency of term t in a document d;
- df_t – document frequency of term t;
- lw_{td} – local weighting of term t in a document d;
- gw_{td} – global weighting of term t in a document d;
- n_d – normalization of vector of terms representing the document d;
- N – total number of documents in the collection;
- cf_t – collection frequency of term t;
- p_{td} – estimated probability of term t to appear in a document d $\left(p_{td} = {}^{tf_{td}}/_{cf_t}\right)$;
- $\overline{g_d}$ – vector of terms representing the document d.

Finally, the element T_{td} of Term-Document Matrix is computed as $T_{td} = lw_{td} * gw_t * n_d, \forall t, d$

Leave *Perform latent semantic indexing* selected (the default). When this option is selected, Text Miner will use Latent Semantic Indexing (LSI) to detect patterns in the associations between terms and concepts to discover the meaning of the document.

The statistics produced and displayed in the Term-Document Matrix contain basic information on the frequency of terms appearing in the document collection. With this information we can "rank" the significance or importance of these terms relative to the collection and particular document. Latent Semantic Indexing, in comparison, uses singular value decomposition (SVD) to map the terms and documents into a common space to find patterns and relationships. For example: if we inspected our document collection, we might find that each time the term "alternator" appeared in an automobile document, the document also included the terms "battery" and "headlights". Or each time the term "brake" appeared in an automobile document, the terms "pads" and "squeaky" also appeared. However there is no detectable pattern regarding the use of the terms "alternator" and "brake". Documents including "alternator" might not include "brake" and documents including "brake" might not include "alternator". Our four terms, battery, headlights, pads, and squeaky describe two different automobile repair issues: failing brakes and a bad alternator. Latent Semantic Indexing will attempt to 1. Distinguish between these two different topics, 2. Identify the documents that deal with faulty brakes, alternator problems or both and 3. Map the terms into a common semantic space using singular value decomposition. SVD is a tool used by Text Miner to extract concepts that explain the main dimensions of meaning of the documents in the collection. The results of LSA are usually hard to examine because the construction of the concept representations will not be fully explained. Interpreting these results is actually more of an art, than a science. However, Text Miner provides several visualizations that simplify this process greatly.

Select **Maximum number of concepts** and leave the default setting of 5. Doing so will tell Text Miner to retain the top 5 most significant concepts. If *Automatic* is selected, Text Miner will calculate the importance of each concept, take the difference between each and report any concepts above the largest difference. For example if three concepts were identified (Concept1, Concept2, and Concept3) and given importance factors of 10, 8, and 2, respectively, Text Miner would keep Concept1 and Concept2 since the difference between Concept2 and Concept 3 (8-2=6) is larger than the difference between Concept1 and Concept2 (10-8=2). If *Minimum percentage explained* is selected, Text Miner will identify the concepts with singular values that, when taken together, sum to the minimum percentage explained, 90% is the default.



Term-Document Matrix Scheme

○ Presence/Absence ● TF-IDF

○ Term Frequency ○ Scaled term frequency

Concept Extraction - Latent Semantic Indexing

☑ Perform latent semantic indexing

○ Automatic

● Maximum number of concepts: 5

○ Minimum percentage explained: 90

Specify the number of top concepts to keep that will define semantic space.

Click **Next** or the **Output Options** tab.

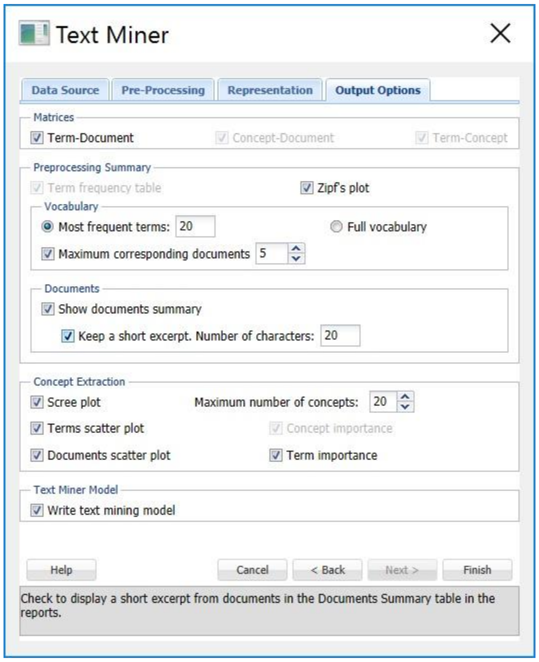
Options Tab

Keep *Term-Document* and *Concept-Document* selected under *Matrices* (the default) and select *Term-Concept* to print each matrix in the output. The *Term-Document* matrix displays the terms across the top of the matrix and the documents down the left side of the matrix. The *Concept – Document* and *Term – Concept* matrices are output from the *Perform latent semantic indexing* option that we selected on the *Representation* tab. In the first matrix, *Concept – Document*, 20 concepts will be listed across the top of the matrix and the documents will be listed down the left side of the matrix. The values in this matrix represent concept coordinates in the identified semantic space. In the *Term-Concept* matrix, the terms will be listed across the top of the matrix and the concepts will be listed down the left side of the matrix. The values in this matrix represent terms in the extracted semantic space.

Keep *Term frequency table* selected (the default) under *Preprocessing Summary* and select *Zipf's plot*. Increase the *Most frequent term*s to 20 and select *Maximum corresponding* documents. The Term frequency table will include the top 20 most frequently occurring terms. The first column, Collection Frequency, displays the number of times the term appears in the collection. The 2nd column, Document Frequency, will display the number of documents that include the term. The third column, Top Documents, will display the top 5 documents where the corresponding term appears the most frequently. The Zipf Plot graphs the document frequency against the term ranks in descending order of frequency.. Zipf's law states that the frequency of terms used in a free-form text drops exponentially, i.e. that people tend to use a relatively small number of words extremely frequently and use a large number of words very rarely.

Keep *Show documents summary* selected and check *Keep a short excerpt.* under *Documents*. Text Miner will produce a table displaying the document ID, length of the document, number of terms and 20 characters of the text of the document.

Select **all plots** under *Concept Extraction* to produce various plots in the output. Select *Write text mining model* under *Text Miner Model* to write the model to an output sheet.



Matrices

☑ Term-Document ☑ Concept-Document ☑ Term-Concept

Preprocessing Summary

☑ Term frequency table ☑ Zipf's plot

Vocabulary

● Most frequent terms: 20 ○ Full vocabulary

☑ Maximum corresponding documents 5

Documents

☑ Show documents summary

☑ Keep a short excerpt. Number of characters: 20

Concept Extraction

☑ Scree plot Maximum number of concepts: 20

☑ Terms scatter plot ☑ Concept importance

☑ Documents scatter plot ☑ Term importance

Text Miner Model

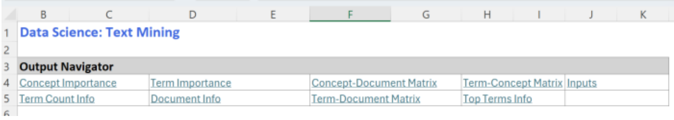
☑ Write text mining model

Check to display a short excerpt from documents in the Documents Summary table in the reports.

Click the **Finish** button to run the Text Mining analysis. Result worksheets are inserted to the right.

Output Results

The Output Navigator appears at the top of each output worksheet. Clicking any of these links will allow you to "jump" to the desired output.



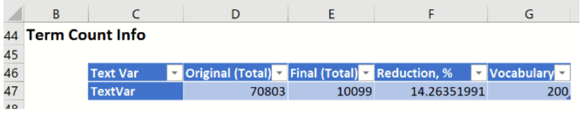
Data Science: Text Mining

Output Navigator

Concept Importance	Term Importance	Concept-Document Matrix	Term-Concept Matrix	Inputs
Term Count Info	Document Info	Term-Document Matrix	Top Terms Info	

Term Count and Document Info

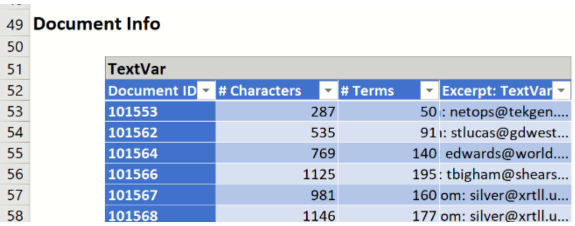
Select the *TM_Output tab*. The *Term Count* table shows that the original term count in the documents was reduced by 14.26% by the removal of stopwords, excluded terms, synonyms, phrase removal and other specified preprocessing procedures.



Term Count Info

Text Var	Original (Total)	Final (Total)	Reduction, %	Vocabulary
TextVar	70803	10099	14.26351991	200

Scroll down to the *Documents* table. This table lists each Document with its length, number of terms, and if *Keep a short excerpt* is selected on the *Output Options* tab and a value is present for *Number of characters*, then an excerpt from each document will be displayed.

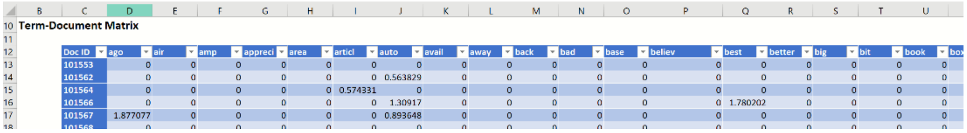


Document Info

TextVar			
Document ID	# Characters	# Terms	Excerpt: TextVar
101553	287	50	: netops@tekgen....
101562	535	91	ı: stlucas@gdwest...
101564	769	140	edwards@world....
101566	1125	195	: tbigham@shears...
101567	981	160	om: silver@xrtll.u...
101568	1146	177	om: silver@xrtll.u...

Term-Document Matrix

Click *TM_TDM*, to display the Term – Document Matrix. As discussed above, this matrix lists the 200 most frequently appearing terms across the top and the document IDs down the left. A portion of this table is shown below. If a term appears in a document, a weight is placed in the corresponding column indicating the importance of the term using our selection of TF-IDF on the Representation dialog.

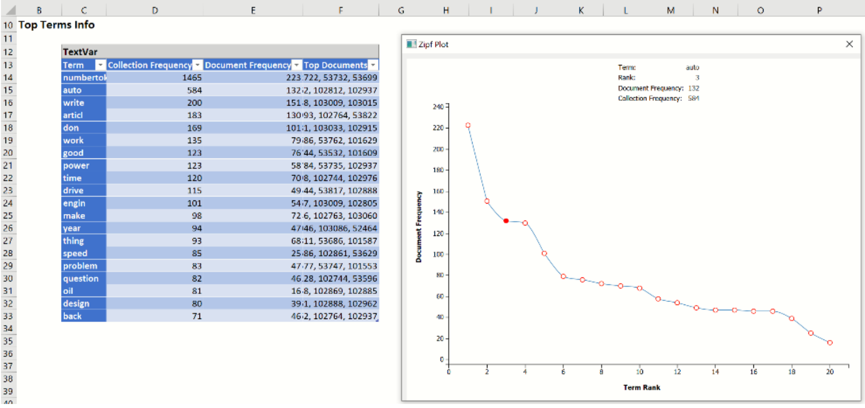


Term-Document Matrix

Doc ID	ago	air	amp	appreci	area	articl	auto	avail	away	back	bad	base	believ	best	better	big	bit	book	box
101553	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
101562	0	0	0	0	0	0	0.563829	0	0	0	0	0	0	0	0	0	0	0	
101564	0	0	0	0	0	0.574331	0	0	0	0	0	0	0	0	0	0	0	0	
101566	0	0	0	0	0	0	1.30917	0	0	0	0	0	0	1.780202	0	0	0	0	
101567	1.877077	0	0	0	0	0	0.893648	0	0	0	0	0	0	0	0	0	0	0	
101568	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Vocabulary Matrix

Click TM_Vocabulary to view the Final List of Terms table. This table contains the top 20 terms occurring in the document collection, the number of documents that include the term and the top 5 document IDs where the corresponding term appears most frequently. In this list we see terms such as “car”, “power”, “engine”, “drive”, and “dealer” which suggests that many of the documents, even the documents from the electronic newsgroup, were related to autos.



Top Terms Info

TextVar			
Term	Collection Frequency	Document Frequency	Top Documents
numbertok	1465	223	722, 53732, 53699
auto	584	132	2, 102812, 102937
write	200	151	8, 103009, 103015
articl	183	130	93, 102764, 53822
don	169	101	1, 103033, 102915
work	135	79	86, 53762, 101629
good	123	76	44, 53532, 101609
power	123	58	84, 53735, 102937
time	120	70	8, 102744, 102976
drive	115	49	44, 53817, 102888
engin	101	54	7, 103009, 102805
make	98	72	6, 102763, 103060
year	94	47	46, 103086, 52464
thing	93	68	11, 53686, 101587
speed	85	25	86, 102861, 53629
problem	83	47	77, 53747, 101553
question	82	46	28, 102744, 53596
oil	81	16	8, 102869, 102885
design	80	39	1, 102888, 102962
back	71	46	2, 102764, 102937

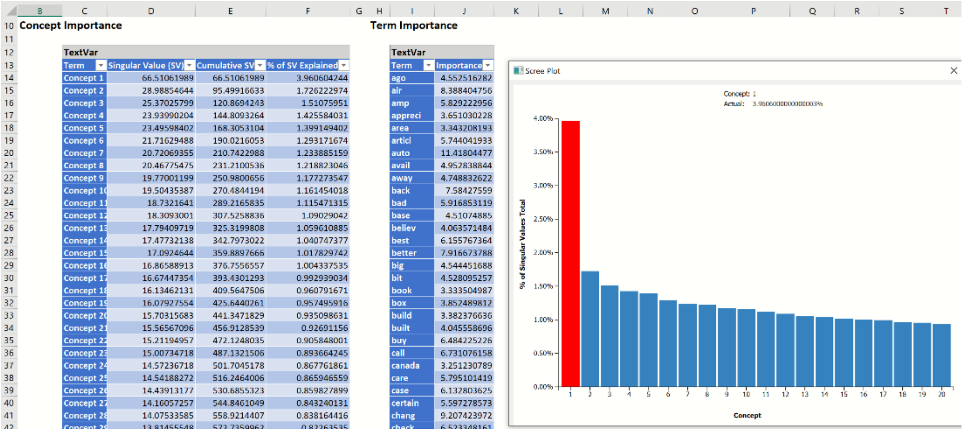
When you click on the TM_Vocabulary tab, the Zipf Plot opens. We see that our collection of documents obey the power law stated by Zipf (see above). As we move from left to right on the graph, the documents that contain the most frequently appearing terms (when ranked from most frequent to least frequent) drop quite steeply. Hover over each data point to see the detailed information about the term corresponding to this data point.

Note: To view charts in the Data Science Cloud app, click Charts on the Ribbon, select the desired worksheet, in this case TM_Vocabulary, then select the desired chart.

The term “numbertoken” is the most frequently occurring term in the document collection appearing in 223 documents (out of 300), 1,083 times total. Compare this to a less frequently occurring term such as "thing" which appears in only 64 documents and only 82 times total.

Concept Importance

Click TM_LSASummary to view the Concept Importance and Term Importance tables. The first table, the Concept Importance table, lists each concept, its singular value, the cumulative singular value and the % singular value explained. (The number of concepts extracted is the minimum of the number of documents (985) and the number of terms (limited to 200).) These values are used to determine which concepts should be used in the Concept – Document Matrix, Concept – Term Matrix and the Scree Plot according to the Users selection on the Representation tab. In this example, we entered “20” for *Maximum number of concepts*.



Concept Importance

TextVar

Term	Singular Value (SV)	Cumulative SV	% of SV Explained
Concept 1	66.51061989	66.51061989	3.960604244
Concept 2	28.98854644	95.49916633	1.726222974
Concept 3	25.37025799	120.8694243	1.51075951
Concept 4	23.93990204	144.8093264	1.425584031
Concept 5	23.49598402	168.3053104	1.399149402
Concept 6	21.71629488	190.0216053	1.293171674
Concept 7	20.72069355	210.7422988	1.233885159
Concept 8	20.46775475	231.2100536	1.218823046
Concept 9	19.77001199	250.9800656	1.177273547
Concept 10	19.50435387	270.4844194	1.161454018
Concept 11	18.7321641	289.2165835	1.115471315
Concept 12	18.3093001	307.5258836	1.09029042
Concept 13	17.79409719	325.3199808	1.059610885
Concept 14	17.47732138	342.7973022	1.040747377
Concept 15	17.0924644	359.8897666	1.017829742
Concept 16	16.86588913	376.7556557	1.004337535
Concept 17	16.67447354	393.4301293	0.992939034
Concept 18	16.13462131	409.5647506	0.960791671
Concept 19	16.07927554	425.6440261	0.957495916
Concept 20	15.70315683	441.3471829	0.935098631
Concept 21	15.56567096	456.9128539	0.92691156
Concept 22	15.21194957	472.1248035	0.905848001
Concept 23	15.00734718	487.1321506	0.893664245
Concept 24	14.57236718	501.7045178	0.867761861
Concept 25	14.54188272	516.2464006	0.865946559
Concept 26	14.43913177	530.6855323	0.859827899
Concept 27	14.16057257	544.8461049	0.843240131
Concept 28	14.07533585	558.9214407	0.838164416
Concept 29	13.81455548	572.7359962	0.82263535

Term Importance

TextVar

Term	Importance
ago	4.552516282
air	8.388404756
amp	5.829222956
appreci	3.651030228
area	3.343208193
articl	5.744041933
auto	11.41804477
avail	4.952838844
away	4.748832622
back	7.58427559
bad	5.916853119
base	4.51074885
believ	4.063571484
best	6.155767364
better	7.916673788
big	4.544451688
bit	4.528095257
book	3.333504987
box	3.852489812
build	3.382376636
built	4.045558696
buy	6.484225226
call	6.731076158
canada	3.251230789
care	5.795101419
case	6.132803625
certain	5.597278573
chang	9.207423972
check	6.523348161

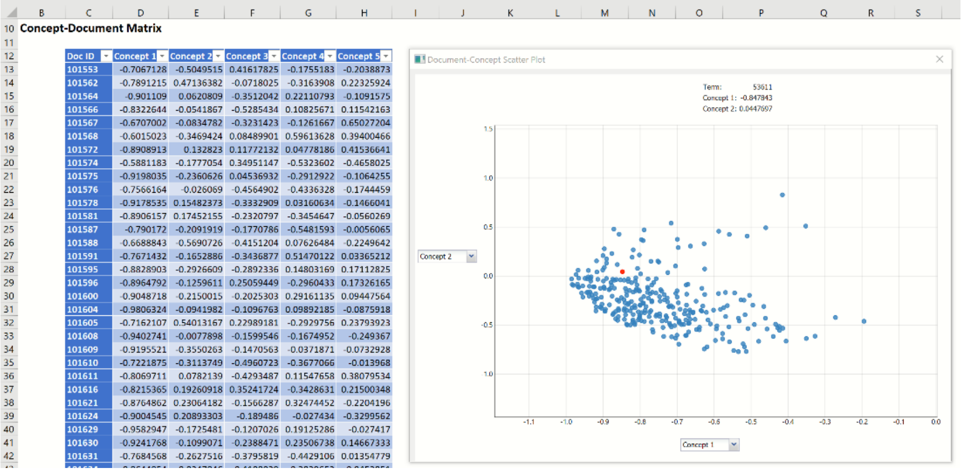
The Term Importance table lists the 200 most important terms. (To increase the number of terms from 200, enter a larger value for Maximum Vocabulary on the Pre-processing tab of Text Miner.)

When you click the TM_LSASummary tab, the Scree Plot opens. This plot gives a graphical representation of the contribution or importance of each concept. The largest “drop” or “elbow” in the plot appears between the 1st and 2nd concept. This suggests that the first top concept explains the leading topic in our collection of documents. Any remaining concepts have significantly reduced importance. However, we can always select more than 1 concept to increase the accuracy of the analysis – it is advised to examine the Concept Importance table and the “Cumulative Singular Value” in particular to identify how many top concepts capture enough information for your application.

Concept Document Matrix

Click *TM_LSA_CDM* to display the Concept – Document Matrix. This matrix displays the top concepts (as selected on the Representation tab) along the top of the matrix and the documents down the left side of the matrix.

When you click on the TM_LSA_CDM tab, the Concept-Document Scatter Plot opens. This graph is a visual representation of the Concept – Document matrix. Note that Analytic Solver Data Science normalizes each document representation so it lies on a unit hypersphere. Documents that appear in the middle of the plot, with concept coordinates near 0 are not explained well by either of the shown concepts. The further the magnitude of coordinate from zero, the more effect that particular concept has for the corresponding document. In fact, two documents placed at extremes of a concept (one close to -1 and other to +1) indicates strong differentiation between these documents in terms of the extracted concept. This provides means for understanding actual meaning of the concept and investigating which concepts have the largest discriminative power, when used to represent the documents from the text collection.



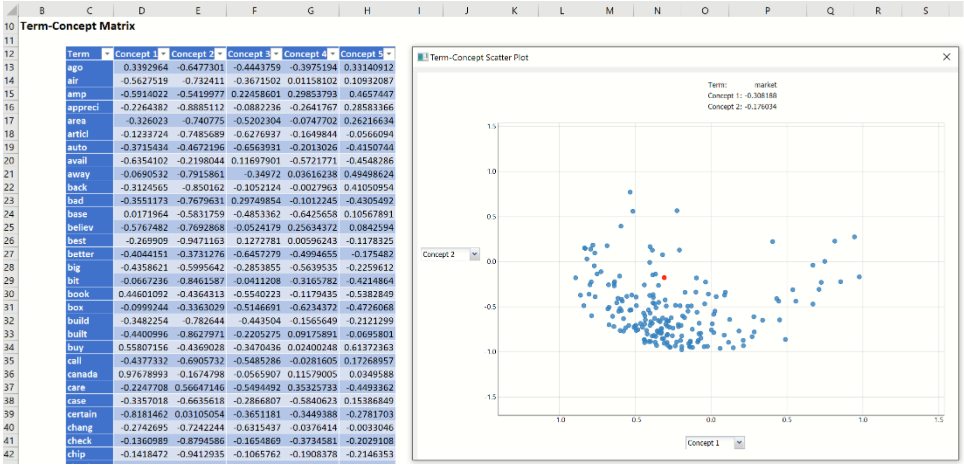
Concept-Document Matrix

Doc ID	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5
101553	-0.7067128	-0.5049515	0.41617825	-0.1755183	-0.2038873
101562	-0.7891215	0.47136382	-0.0718025	-0.3163908	0.22325924
101564	-0.901109	0.0620809	-0.3512042	0.22110793	-0.1091575
101566	-0.8322644	-0.0541867	-0.5285434	0.10825671	0.11542163
101567	-0.6707002	-0.0834782	-0.3231423	-0.1261667	0.65027204
101568	-0.6015023	-0.3469424	0.08489901	0.59613628	0.39400466
101572	-0.8908913	0.132823	0.11772132	0.04778186	0.41536641
101574	-0.5881183	-0.1777054	0.34951147	-0.5323602	-0.4658025
101575	-0.9198035	-0.2360626	0.04536932	-0.2912922	-0.1064255
101576	-0.7566164	-0.026069	-0.4564902	-0.4336328	-0.1744459
101578	-0.9178535	0.15482373	-0.3332909	0.03160634	-0.1466041
101581	-0.8906157	0.17452155	-0.2320797	-0.3454647	-0.0560269
101587	-0.790172	-0.2091919	-0.1770786	-0.5481593	-0.0056065
101588	-0.6688843	-0.5690726	-0.4151204	0.07626484	-0.2249642
101591	-0.7671432	-0.1652886	-0.3436877	0.51470122	0.03365212
101595	-0.8828903	-0.2926609	-0.2892336	0.14803169	0.17112825
101596	-0.8964792	-0.1259611	0.25059449	-0.2960433	0.17326165
101600	-0.9048718	-0.2150015	-0.2025303	0.29161135	0.09447564
101604	-0.9806324	-0.0941982	-0.1096763	0.09892185	-0.0875918
101605	-0.7162107	0.54013167	0.22989181	-0.2929756	0.23793923
101608	-0.9402741	-0.0077898	-0.1599546	-0.1674952	-0.249367
101609	-0.9195521	-0.3550263	-0.1470563	-0.0371871	-0.0732928
101610	-0.7221875	-0.3113749	-0.4960723	-0.3677066	-0.013968
101611	-0.8069711	0.0782139	-0.4293487	0.11547658	0.38079534
101616	-0.8215365	0.19260918	0.35241724	-0.3428631	0.21500348
101621	-0.8764862	0.23064182	-0.1566287	0.32474452	-0.2204196
101624	-0.9004545	0.20893303	-0.189486	-0.027434	-0.3299562
101629	-0.9582947	-0.1725481	-0.1207026	0.19125286	-0.027417
101630	-0.9241768	-0.1099071	-0.2388471	0.23506738	0.14667333
101631	-0.7684568	-0.2627516	-0.3795819	-0.4429106	0.01354779

You can examine all extracted concepts by changing the axes on a scatter plot - click the down pointing arrow next to Concept 1 or the concept on the Y axis by clicking the right pointing arrow next to Concept 2. Use your touchscreen or your mouse scroll wheel to zoom in and out.

Term-Concept Matrix

Double click *TM_LSA_CTM* to display the Concept – Term Matrix which lists the top 5 most important concepts along the top of the matrix and the top 200 most frequently appearing terms down the side of the matrix.



Term-Concept Matrix

Term	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5
ago	0.3392964	-0.6477301	-0.4443759	-0.3975194	0.33140912
air	-0.5627519	-0.732411	-0.3671502	0.01158102	0.10932087
amp	-0.5914022	-0.5419977	0.22458601	0.29853793	0.4657447
appreci	-0.2264382	-0.8885112	-0.0882236	-0.2641767	0.28583366
area	-0.326023	-0.740775	-0.5202304	-0.0747702	0.26216634
articl	-0.1233724	-0.7485689	-0.6276937	-0.1649844	-0.0566094
auto	-0.3715434	-0.4672196	-0.6563931	-0.2013026	-0.4150744
avail	-0.6354102	-0.2198044	0.11697901	-0.5721771	-0.4548286
away	-0.0690532	-0.7915861	-0.34972	0.03616238	0.49498624
back	-0.3124565	-0.850162	-0.1052124	-0.0027963	0.41050954
bad	-0.3551173	-0.7679631	0.29749854	-0.1012245	-0.4305492
base	0.0171964	-0.5831759	-0.4853362	-0.6425658	0.10567891
believ	-0.5767482	-0.7692868	-0.0524179	0.25634372	0.0842594
best	-0.269909	-0.9471163	0.1272781	0.00596243	-0.1178325
better	-0.4044151	-0.3731276	-0.6457279	-0.4994655	-0.175482
big	-0.4358621	-0.5995642	-0.2853855	-0.5639535	-0.2259612
bit	-0.0667236	-0.8461587	-0.0411208	-0.3165782	-0.4214864
book	0.44601092	-0.4364313	-0.5540223	-0.1179435	-0.5382849
box	-0.0999244	-0.3363029	-0.5146691	-0.6234372	-0.4726068
build	-0.3482254	-0.782644	-0.443504	-0.1565649	-0.2121299
built	-0.4400996	-0.8627971	-0.2205275	0.09175891	-0.0695801
buy	0.55807156	-0.4369028	-0.3470436	0.02400248	0.61372363
call	-0.4377332	-0.6905732	-0.5485286	-0.0281605	0.17268957
canada	0.97678993	-0.1674798	-0.0565907	0.11579005	0.0349588
care	-0.2247708	0.56647146	-0.5494492	0.35325733	-0.4493362
case	-0.3357018	-0.6635618	-0.2866807	-0.5840623	0.15386849
certain	-0.8181462	0.03105054	-0.3651181	-0.3449388	-0.2781703
chang	-0.2742695	-0.7242244	-0.6315437	-0.0376414	-0.0033046
check	-0.1360989	-0.8794586	-0.1654869	-0.3734581	-0.2029108
chip	-0.1418472	-0.9412935	-0.1065762	-0.1908378	-0.2146353

When you click on the TM_LSA-CTM tab, the Term-Concept Scatter Plot opens. This graph is a visual representation of the Concept – Term Matrix. It displays all terms from the final vocabulary in terms of two concepts. Similarly to the Concept-Document scatter plot, the Concept-Term scatter plot visualizes the distribution of vocabulary terms in the semantic space of meaning extracted with LSA. The coordinates are also normalized, so the range of axes is always [-1,1], where extreme values (close to +/-1) highlight the importance or "load" of each term to a particular concept. The terms appearing in a zero-neighborhood of concept range do not contribute much to a concept definition. In our example, if we identify a concept having a set of terms that can be divided into two groups: one related to "Autos" and other to "Electronics", and these groups are distant from each other on the axis corresponding to this concept, this would definitely provide an evidence that this particular concept “caught” some pattern in the text collection that is capable of discriminating the topic of article. Therefore, Term-Concept scatter plot is an extremely valuable tool for examining and understanding the main topics in the collection of documents, finding similar words that indicate similar concept, or the terms explaining the concept from “opposite sides” (e.g. *term1* can be related to cheap affordable electronics and *term2* can be related to expensive luxury electronics)

Recall that if you want to examine different pair of concepts, click the down pointing arrow next to Concept 1 and the right pointing arrow next to Concept 2 to change the concepts on either axis. Use your touchscreen or mouse wheel to scroll in or out.

Stored PMML models for TFIDF and LSA

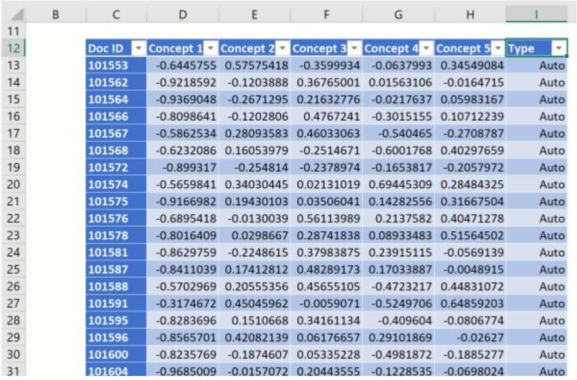
Since "Write text mining model" on the Output Options tab, two more tabs are created containing PMML models for the TFIDF and LSA models. These models can be used when scoring a series of new documents. For more information on how to process this new data using these two saved models, see the Text Mining chapter within the Data Science Reference Guide.

Classification with Concept Document Matrix

From here, we can use any of the six classification algorithms to classify our documents according to some term or concept using the Term – Document matrix, Concept – Document matrix or Concept – Term matrix where each document becomes a “record” and each concept becomes a “variable”. If wanting to classify documents based on a binary variable such as Auto email/non-Auto email, then we would use either the Term – Document or Concept – Document matrix. If wanting to cluster terms or classify terms, then we would use the Term-Concept matrix. We could even use the transpose of the Term – Document matrix where each term would become a “record” and each column would become a “feature”.   See the Analytic Solver Data Science User Guide for an example model that uses the Logistic Regression Classification method to create a classification model using the Concept Document matrix within *TM_LSA_CDM*.

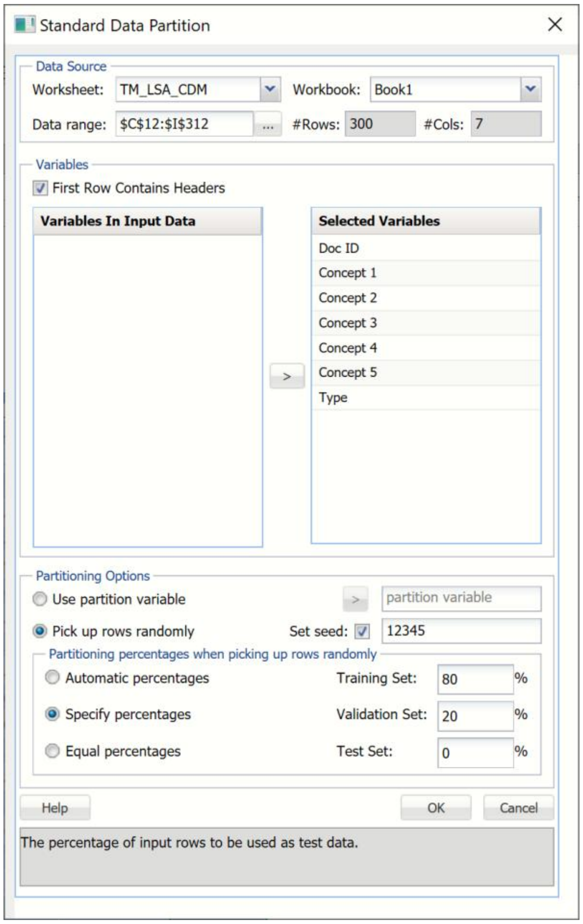
In this example, we will use the Logistic Regression Classification method to create a classification model using the Concept Document matrix on *TM_LSA_CDM*. Recall that this matrix includes the top twenty concepts extracted from the document collection across the top of the matrix and each document in the sample down the left. Each concept will now become a “feature” and each document will now become a “record”.

First, we’ll need to append a new column with the class that the document is currently assigned: electronics or autos. Since we sorted our documents at the beginning of the example starting with Autos, we can simply enter “Autos” into column I for Document IDs 101553 through 103096 (or cells I13:I162) and enter “Electronics” into column I for Document IDs 52434 through 53879 (or cells I163:I312). Give the column the title "Type".



Doc ID	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5	Type
101553	-0.6445755	0.57575418	-0.3599934	-0.0637993	0.34549084	Auto
101562	-0.9218592	-0.1203888	0.36765001	0.01563106	-0.0164715	Auto
101564	-0.9369048	-0.2671295	0.21632776	-0.0217637	0.05983167	Auto
101566	-0.8098641	-0.1202806	0.4767241	-0.3015155	0.10712239	Auto
101567	-0.5862534	0.28093583	0.46033063	-0.540465	-0.2708787	Auto
101568	-0.6232086	0.16053979	-0.2514671	-0.6001768	0.40297659	Auto
101572	-0.899317	-0.254814	-0.2378974	-0.1653817	-0.2057972	Auto
101574	-0.5659841	0.34030445	0.02131019	0.69445309	0.28484325	Auto
101575	-0.9166982	0.19430103	0.03506041	0.14282556	0.31667504	Auto
101576	-0.6895418	-0.0130039	0.56113989	0.2137582	0.40471278	Auto
101578	-0.8016409	0.0298667	0.28741838	0.08933483	0.51564502	Auto
101581	-0.8629759	-0.2248615	0.37983875	0.23915115	-0.0569139	Auto
101587	-0.8411039	0.17412812	0.48289173	0.17033887	-0.0048915	Auto
101588	-0.5702969	0.20555356	0.45655105	-0.4723217	0.44831072	Auto
101591	-0.3174672	0.45045962	-0.0059071	-0.5249706	0.64859203	Auto
101595	-0.8283696	0.1510668	0.34161134	-0.409604	-0.0806774	Auto
101596	-0.8565701	0.42082139	0.06176657	0.29101869	-0.02627	Auto
101600	-0.8235769	-0.1874607	0.05335228	-0.4981872	-0.1885277	Auto
101604	-0.9685009	-0.0157072	0.20443555	-0.1228535	-0.0698024	Auto

First, we'll need to partition our data into two datasets, a training dataset where the model will be "trained" and a validation dataset where the newly created model can be tested, or validated. When the model is being trained, the actual class label assignments are "shown" to the algorithm in order for it to "learn" which variables (or concepts) result in an "auto" or "electronic" assignment. When the model is being validated or tested, the known classification is only used to evaluate the performance of the algorithm. Click **Partition – Standard Partition** on the Text Miner ribbon to open the *Standard Data Partition* dialog. Select all variables in the *Variables In Input Data* list box, then click > to move all to the *Selected Variables* list box. Select **Specify percentages** Under *Partitioning percentages when picking up rows randomly* (at the bottom) and enter **80** for *Training Set*. Automatically, *20* will be entered for *Validation Set.*



Standard Data Partition

Data Source

Worksheet: TM_LSA_CDM

Workbook: Book1

Data range: $C$12:$I$312

#Rows: 300

#Cols: 7

Variables

☑ First Row Contains Headers

Variables In Input Data

Selected Variables: Doc ID, Concept 1, Concept 2, Concept 3, Concept 4, Concept 5, Type

Partitioning Options

○ Use partition variable

● Pick up rows randomly

Set seed: ☑ 12345

Partitioning percentages when picking up rows randomly

○ Automatic percentages

● Specify percentages

○ Equal percentages

Training Set: 80 %

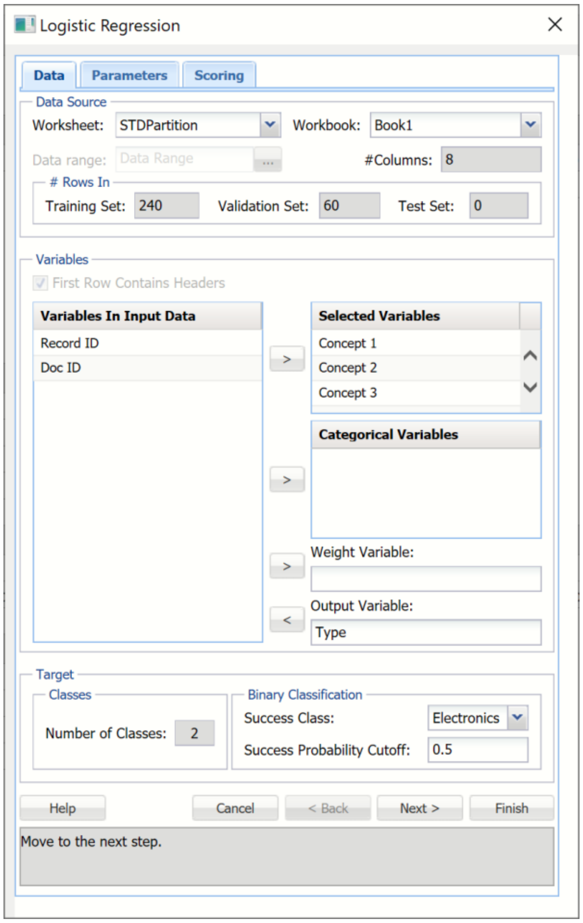
Validation Set: 20 %

Test Set: 0 %

The percentage of input rows to be used as test data.

Click **Finish** to partition the data into two randomly selected datasets: The Training dataset containing 80% of the "records" (or documents) and the Validation dataset containing 20% of the "records". (For more information on partitioning, please see the *Standard Partitioning* chapter that appears in the Analytic Solver Data Science Reference Guide.)

Now click Classify – Logistic Regression to open the *Logistic Regression – Step 1 of 3* dialog. Select all 5 concents under *Variables In Input Data* list box and click > to move them to the *Selected Variables* list box. Doing so selects these variables as inputs to the classification method. Select **Type**, then click the > next *to Output Variable* to add this variable as the Output Variable.



Logistic Regression

Data Source
Worksheet: STDPartition
Workbook: Book1
Data range: Data Range
#Columns: 8

Rows In
Training Set: 240
Validation Set: 60
Test Set: 0

Variables
☑ First Row Contains Headers

Variables In Input Data
Record ID
Doc ID

Selected Variables
Concept 1
Concept 2
Concept 3

Categorical Variables

Weight Variable:

Output Variable: Type

Target
Classes
Number of Classes: 2

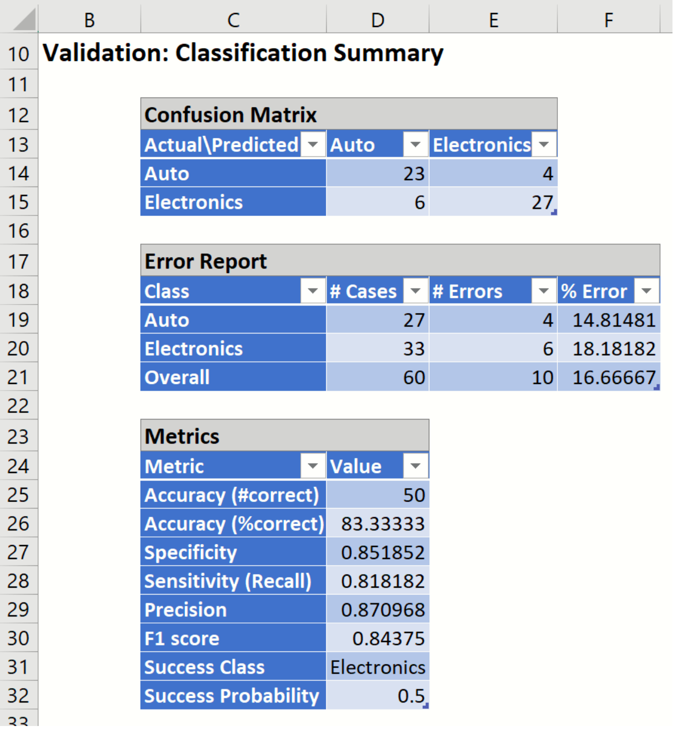
Binary Classification
Success Class: Electronics
Success Probability Cutoff: 0.5

Move to the next step.

Click **Finish** to accept all defaults and run Logistic Regression.

Select *DA_ValidationScore* tab and scroll down to the Validation Classification Summary, shown below.

Text Miner used the training dataset to "train" the Logistic Regression model to classify each "record" (or document) as an "autos" or "electronics" document. Afterwards, Text Miner tested the newly created Logistic Regression model on the records in the validation dataset and assigned each record (or document) a classification.



Validation: Classification Summary

Confusion Matrix

Actual\Predicted	Auto	Electronics
Auto	23	4
Electronics	6	27

Error Report

Class	# Cases	# Errors	% Error
Auto	27	4	14.81481
Electronics	33	6	18.18182
Overall	60	10	16.66667

Metrics

Metric	Value
Accuracy (#correct)	50
Accuracy (%correct)	83.33333
Specificity	0.851852
Sensitivity (Recall)	0.818182
Precision	0.870968
F1 score	0.84375
Success Class	Electronics
Success Probability	0.5

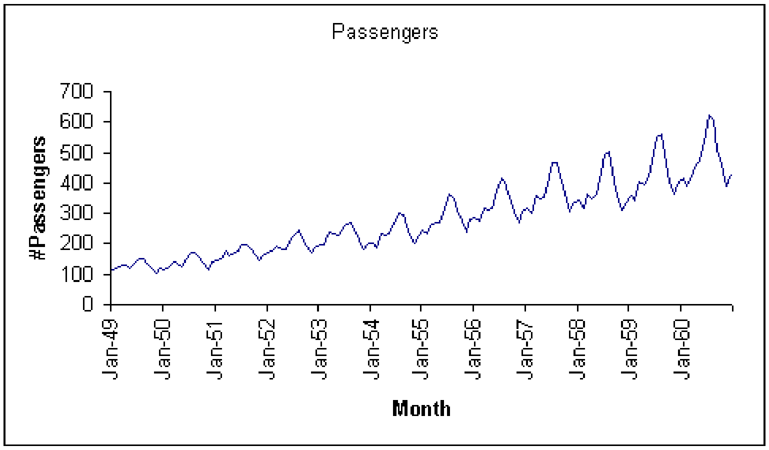
As you can see in the reports above, Logistic Regression was able to correctly classify 50 out of a total of 60 documents in the validation partition, which translates to an overall error of 16.67%, (For more information on how to read the summary report, see the Logistic Regression chapter later on in this guide.)

This concludes our example on how to use Analytic Solver Data Science's Text Miner feature. This example has illustrated how Analytic Solver Data Science provides powerful tools for importing a collection of documents for comprehensive text preprocessing, quantitation, and concept extraction, in order to create a model that can be used to process new documents – all performed without any manual intervention. When using Text Miner in conjunction with our classification algorithms, Analytic Solver Data Science can be used to classify customer reviews as satisfied/not satisfied, distinguish between which products garnered the least negative reviews, extract the topics of articles, cluster the documents/terms, etc. The applications for Text Miner are endless!

Exploring a Time Series Dataset

Introduction

Time series datasets contain a set of observations generated sequentially in time. Organizations of all types and sizes utilize time series datasets for analysis and forecasting for predicting next year's sales figures, raw material demand, monthly airline bookings, etc.   .



Example of a time series dataset: Monthly airline bookings.

A time series model is first used to obtain an understanding of the underlying forces and structure that produced the data and then secondly, to fit a model that will predict future behavior. In the first step, the analysis of the data, a model is created to uncover seasonal patterns or trends in the data, for example bathing suit sales in June. In the second step, forecasting, the model is used to predict the value of the data in the future, for example, next year's bathing suit sales. Separate modeling methods are required to create each type of model.

Analytic Solver Data Science features two techniques for exploring trends in a dataset, ACF (Autocorrelation function) and PACF (Partial autocorrelation function). These techniques help the user to explore various patterns in the data which can be used in the creation of the model. After the data is analyzed, a model can be fit to the data using Analytic Solver Data Science's ARIMA method.

Autocorrelation (ACF)

Autocorrelation (ACF) is the correlation between neighboring observations in a time series. When determining if an autocorrelation exists, the original time series is compared to the "lagged" series. This lagged series is simply the original series moved one time period forward (x_n vs x_{n+1}). Suppose there are 5 time based observations: 10, 20, 30, 40, and 50. When lag = 1, the original series is moved forward one time period. When lag = 2, the original series is moved forward two time periods.

Day	Observed Value	Lag-1	Lag-2
1	10		
2	20	10	
3	30	20	10
4	40	30	20
5	50	40	30

The autocorrelation is computed according to the formula:

$$r_k = \frac{\sum_{i=k+1}^{n}(Y_t-\bar{Y})(Y_{t-k}-\bar{Y})}{\sum_{i=1}^{n}(Y_t-\bar{Y})^2} \quad \text{where k = 0, 1, 2, …., n}$$

Where Y_t is the Observed Value at time t, $\bar{Y}$ is the mean of the Observed Values and Y_{t-k} is the value for Lag-k.

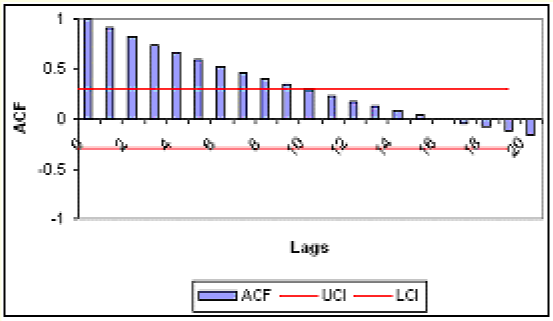
For example, using the values above, the autocorrelation for Lag-1 and Lag - 2 can be calculated as follows.

$\bar{Y}$ = (10 + 20 + 30 + 40 + 50) / 5 = 30

r_1 = ((20 – 30) * (10 - 30) + (30 - 30) * (20 - 30) + (40 - 30) * (30 - 30) + (50 – 30) * (40 – 30)) / ((10 – 30)^2 + (20 - 30)^2 + (30 – 30)^2 + (40 – 30)^2 + (50 – 30)^2) = 0.4

r_2 =((30 – 30) * (10 – 30) + (40 – 30) * (20 – 30) + (50 – 30) * (30 – 30)) / (((10 – 30)^2 + (20 - 30)^2 + (30 – 30)^2 + (40 – 30)^2 + (50 – 30)^2) = -0.1

The two red horizontal lines on the graph below delineate the Upper confidence level (UCL) and the Lower confidence level (LCL). If the data is random, then the plot should be within the UCL and LCL. If the plot exceeds either of these two levels, as seen in the plot above, then it can be presumed that some correlation exists in the data.



Partial Autocorrelation Function (PACF)

This technique is used to compute and plot the partial autocorrelations between the original series and the lags. However, PACF eliminates all linear dependence in the time series beyond the specified lag.

ARIMA

An ARIMA (autoregressive integrated moving-average models) model is a regression-type model that includes autocorrelation. The basic assumption in estimating the ARIMA coefficients is that the data are stationary, that is, the trend or seasonality cannot affect the variance. This is generally not true. To achieve the stationary data, Analytic Solver Data Science will first apply “differencing”: ordinary, seasonal or both.

After Analytic Solver Data Science fits the model, various results will be available. The quality of the model can be evaluated by comparing the time plot of the actual values with the forecasted values. If both curves are close, then it can be assumed that the model is a good fit. The model should expose any trends and seasonality, if any exist. If the residuals are random then the model can be assumed a good fit. However, if the residuals exhibit a trend, then the model should be refined. Fitting an ARIMA model with parameters (0,1,1) will give the same results as exponential smoothing. Fitting an ARIMA model with parameters (0,2,2) will give the same results as double exponential smoothing.

Partitioning

To avoid over fitting of the data and to be able to evaluate the predictive performance of the model on new data, we must first partition the data into training and validation sets using Analytic Solver Data Science’s time series partitioning utility. After the data is partitioned, ACF, PACF, and ARIMA can be applied to the dataset.

Examples for Time Series Analysis

The examples below illustrate how Analytic Solver Data Science can be used to explore the Income.xlsx dataset to uncover trends and seasonalities in a dataset. Click **Help – Examples** on the Data Science ribbon, then **Forecasting/Data Science Examples** and open the example dataset, **Income.xlsx**. This dataset contains the average income of tax payers by state.

Typically the following steps are performed in a time series analysis.

1. The data is first partitioned into two sets with 60% of the data assigned to the training set and 40% of the data assigned to validation.
2. Exploratory techniques are applied to both the training and validation sets. If the results are in synch then the model can be fit. If the ACF and PACF plots are the same, then the same model can be used for both sets.
3. The model is fit using the ARIMA method.
4. When we fit a model using the ARIMA method, Analytic Solver displays the ACF and PACF plots for residuals. If these plots are in the band of UCL and LCL then it indicates that the residuals are random and the model is adequate.
2. If the residuals are not within the bands, then some correlation exists, and the model should be improved.

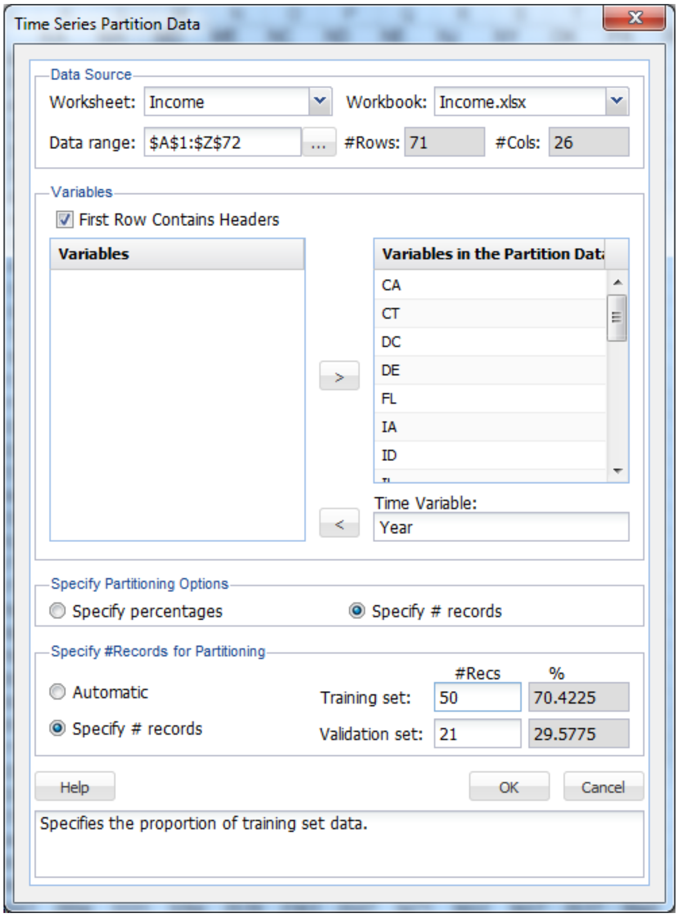
First we must perform a partition on the data. Click **Partition** within the Time Series group on the Data Science ribbon to open the following dialog.

Select **Year** under *Variables* and click > to define the variable as the *Time Variable*. Select the remaining variables under *Variables* and click > to include them in the partitioned data.

Select **Specify #Records** under *Specify Partitioning Options* to specify the number of records assigned to the training and validation sets. Then select

Specify #Records under *Specify #Records for Partitioning*. Enter **50** for the number of *Training Set* records and **21** for the number of *Validation Set* records.

If **Specify Percentages** is selected under *Specify Partitioning Options*, Analytic Solver Data Science will assign a percentage of records to each set according to the values entered by the user or automatically entered by Analytic Solver Data Science under *Specify Percentages for Partitioning.*



Time Series Partition Data

Data Source

Worksheet: Income

Workbook: Income.xlsx

Data range: $A$1:$Z$72

#Rows: 71

#Cols: 26

Variables

☑ First Row Contains Headers

Variables	Variables in the Partition Data
	CA
	CT
	DC
	DE
	FL
	IA
	ID

Time Variable: Year

Specify Partitioning Options

○ Specify percentages ● Specify # records

Specify #Records for Partitioning

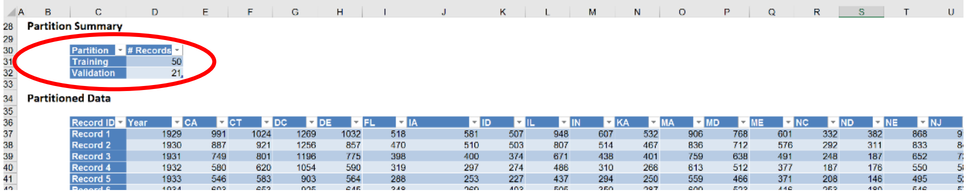
○ Automatic

● Specify # records

	#Recs	%
Training set:	50	70.4225
Validation set:	21	29.5775

Specifies the proportion of training set data.

Click **OK**. *TSPartition* is inserted into the Model tab of the Solver task pane under Reports – Time Series Partition – Run 1.



Partition Summary

Partition	# Records
Training	50
Validation	21

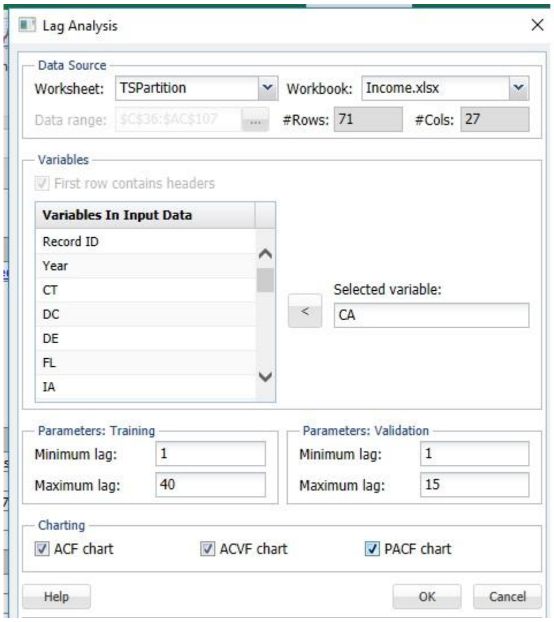
Partitioned Data

Record ID	Year	CA	CT	DC	DE	FL	IA	ID	IL	IN	KA	MA	MD	ME	NC	ND	NE	NJ
Record 1	1929	991	1024	1269	1032	518	581	507	948	607	532	906	768	601	332	382	868	9
Record 2	1930	887	921	1256	857	470	510	503	807	514	467	836	712	576	292	311	833	8
Record 3	1931	749	801	1196	775	398	400	374	671	438	401	759	638	491	248	187	652	7
Record 4	1932	580	620	1054	590	319	297	274	486	310	266	613	512	377	187	176	550	5
Record 5	1933	546	583	903	564	288	253	227	437	294	250	559	466	371	208	146	495	5

Note in the output above, the partitioning method is sequential (rather than random). The first 50 observations have been assigned to the training set and the remaining 21 observations have been assigned to the validation set.

Open the Lag Analysis dialog by clicking **ARIMA – Lag Analysis**. Select **CA** under *Variables In input data*, then click > to move the variable to *Selected variable*. Enter **1** for Minimum Lag and **40** for *Maximum Lag under Parameters: Training* and 1 for Minimum Lag and 15 for Maximum Lag under *Parameters: Validation.*

Under Charting, select ACF chart, ACVF chart, and PACF chart to include each chart in the output.



Lag Analysis

Data Source

Worksheet: TSPartition Workbook: Income.xlsx

Data range: $C$36:$AC$107 #Rows: 71 #Cols: 27

Variables

☑ First row contains headers

Variables In Input Data
Record ID
Year
CT
DC
DE
FL
IA

Selected variable: CA

Parameters: Training

Minimum lag: 1

Maximum lag: 40

Parameters: Validation

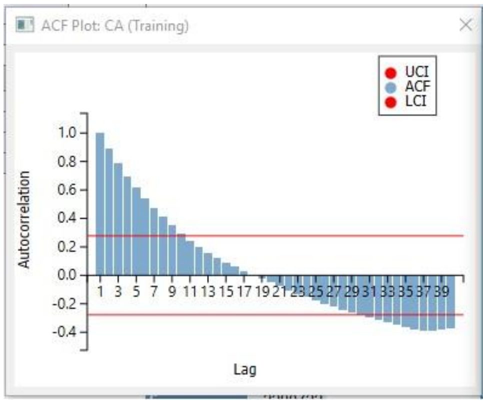
Minimum lag: 1

Maximum lag: 15

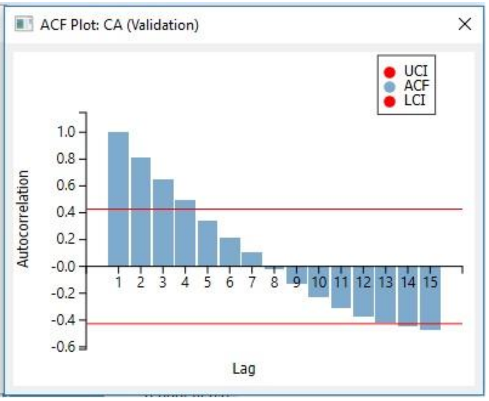
Charting

☑ ACF chart ☑ ACVF chart ☑ PACF chart

Click **OK**. *TS_Lags* is inserted into the task pane under Reports – Autocorrelations – Run 1.

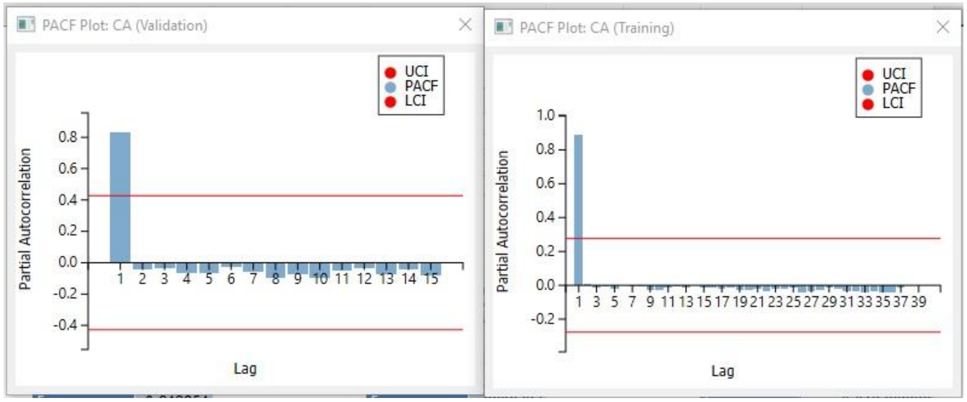


ACF Plot: CA (Training)



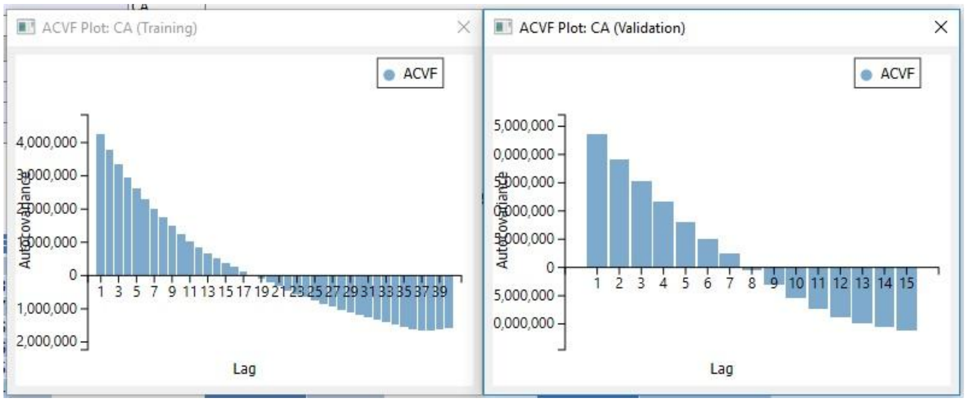
ACF Plot: CA (Validation)

First, let's take a look at the ACF charts. Note on each chart, the autocorrelation decreases as the number of lags increase. This suggests that a definite pattern does exist in each partition. However, since the pattern does not repeat, it can be assumed that no seasonality is included in the data. In addition, both charts appear to exhibit a similar pattern.



The PACF functions show a definite pattern which means there is a trend in the data. However, since the pattern does not repeat, we can conclude that the data does not show any seasonality.

A plot of the autocovariance values has been added to the output.



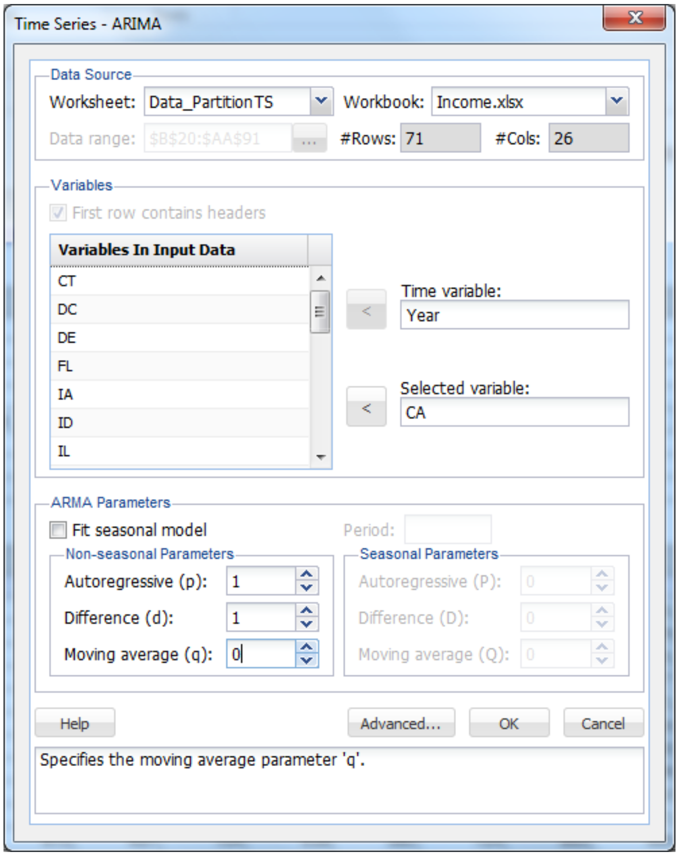
All three charts suggest that a definite pattern exists in the data, but no seasonality. In addition, both datasets exhibit the same behavior in both the training and validation sets which suggests that the same model could be appropriate for each. Now we are ready to fit the model.

The ARIMA model accepts three parameters: p – the number of autoregressive terms, d – the number of non-seasonal differences, and q – the number of lagged errors (moving averages).

Recall that the ACF plot showed no seasonality in the data which means that autocorrelation is almost static, decreasing with the number of lags increasing. This suggests setting q = 0 since there appears to be no lagged errors. The PACF plot displayed a large value for the first lag but minimal plots for successive lags. This suggest setting p =1. With most datasets, setting d =1 is sufficient or can at least be a starting point.

Click back to the TSPartition tab and then click **ARIMA – ARIMA Model** to bring up the *Time Series – ARIMA* dialog.

Select **CA** under *Variables In input data* then click > to move the variable to the *Selected Variable* field. Under *Nonseasonal Parameters* set **Autoregressive (p)** to **1**, **Difference (d)** to **1** and **Moving Average (q)** to **0**.



Time Series - ARIMA

Data Source

Worksheet: Data_PartitionTS
Workbook: Income.xlsx
Data range: $B$20:$AA$91
#Rows: 71
#Cols: 26

Variables

☑ First row contains headers

Variables In Input Data: CT, DC, DE, FL, IA, ID, IL

Time variable: Year

Selected variable: CA

ARMA Parameters

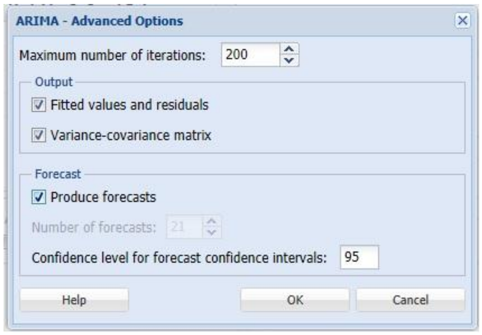
☐ Fit seasonal model
Period:

Non-seasonal Parameters
Autoregressive (p): 1
Difference (d): 1
Moving average (q): 0

Seasonal Parameters
Autoregressive (P): 0
Difference (D): 0
Moving average (Q): 0

Specifies the moving average parameter 'q'.

Click **Advanced** to open the *ARIMA – Advanced Options* dialog. Select **Fitted Values and residuals**, **Produce forecasts**, and **Report Forecast Confidence Intervals**. The default *Confidence Level* setting of *95* is automatically entered. The option *Variance-covariance matrix* is selected by default.



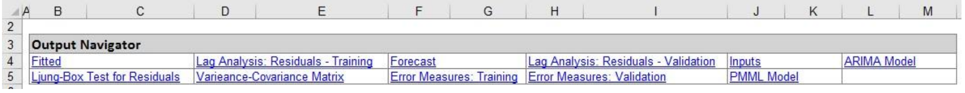
ARIMA - Advanced Options

Maximum number of iterations: 200

Output
☑ Fitted values and residuals
☑ Variance-covariance matrix

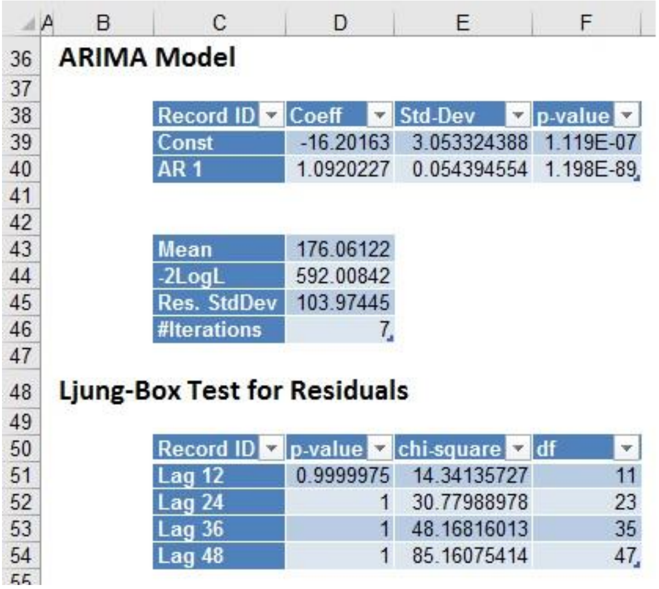
Forecast
☑ Produce forecasts
Number of forecasts: 21
Confidence level for forecast confidence intervals: 95

Click **OK** on the *ARIMA-Advanced Options* dialog and again on the *Time Series – ARIMA* dialog. Analytic Solver Data Science calculates and displays various parameters and charts in four output sheets, *Arima_Output*, *Arima_Fitted, Arima_Forecast* and *Arima_Stored*. Click the *Arima_Output* tab to view the Output Navigator.



Output Navigator					
Fitted	Lag Analysis: Residuals - Training	Forecast	Lag Analysis: Residuals - Validation	Inputs	ARIMA Model
Ljung-Box Test for Residuals	Varieance-Covariance Matrix	Error Measures: Training	Error Measures: Validation	PMML Model	

Click the ARIMA Model link on the Output Navigator to move to display the ARIMA Model and Ljung-Box Test Results on Residuals.



ARIMA Model

Record ID	Coeff	Std-Dev	p-value
Const	-16.20163	3.053324388	1.119E-07
AR 1	1.0920227	0.054394554	1.198E-89

Mean	176.06122
-2LogL	592.00842
Res. StdDev	103.97445
#Iterations	7

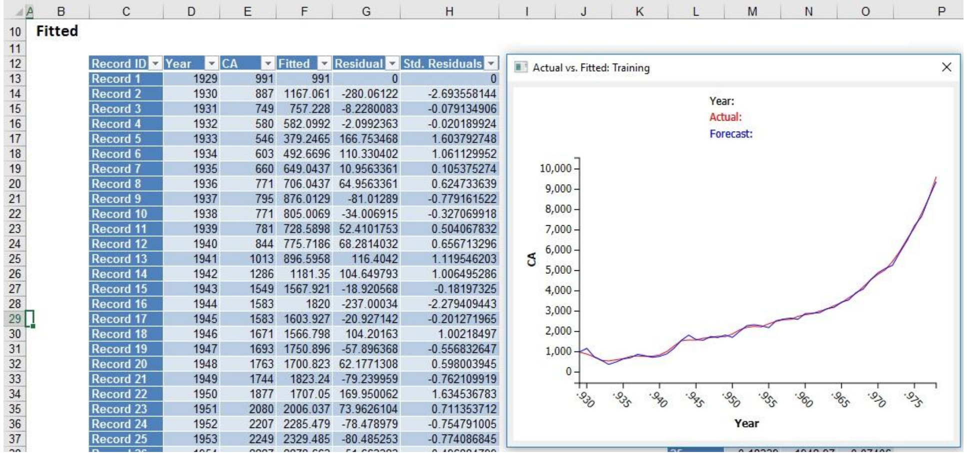
Ljung-Box Test for Residuals

Record ID	p-value	chi-square	df
Lag 12	0.9999975	14.34135727	11
Lag 24	1	30.77988978	23
Lag 36	1	48.16816013	35
Lag 48	1	85.16075414	47

Analytic Solver has calculated the constant term and the AR1 term for our model, as seen above. These are the constant and f1 terms of our forecasting equation. See the following output of the Chi - square test.

The very small p-values for the constant term (1.119E-7) and AR1 term (1.19e-89) suggest that the model is a good fit to our data.

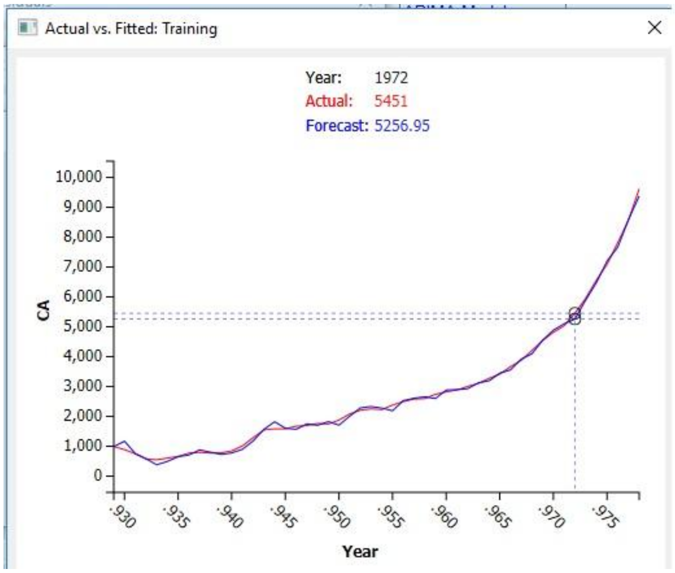
Click the Lag Analysis: Residuals – Training link. This table plots the actual and fitted values and the resulting residuals for the training partition. As shown in the graph below, the Actual and Forecasted values match up fairly well. The usefulness of the model in forecasting will depend upon how close the actual and forecasted values are in the Forecast, which we will inspect later.



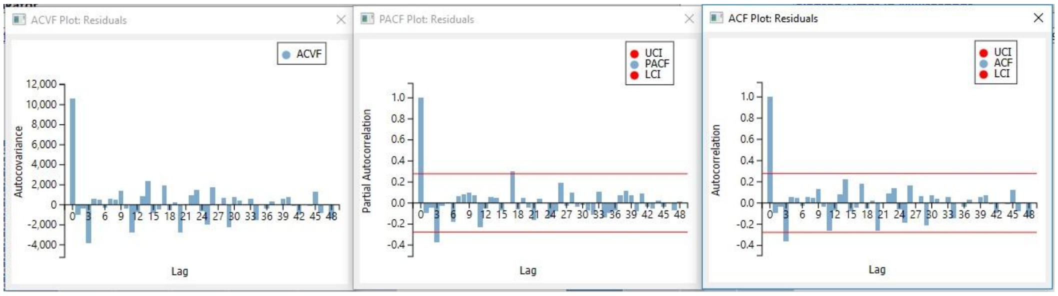
Fitted

Record ID	Year	CA	Fitted	Residual	Std. Residuals
Record 1	1929	991	991	0	0
Record 2	1930	887	1167.061	-280.06122	-2.693558144
Record 3	1931	749	757.228	-8.2280083	-0.079134906
Record 4	1932	580	582.0992	-2.0992363	-0.020189924
Record 5	1933	546	379.2465	166.753468	1.603792748
Record 6	1934	603	492.6696	110.330402	1.061129952
Record 7	1935	660	649.0437	10.9563361	0.105375274
Record 8	1936	771	706.0437	64.9563361	0.624733639
Record 9	1937	795	876.0129	-81.01289	-0.779161522
Record 10	1938	771	805.0069	-34.006915	-0.327069918
Record 11	1939	781	728.5898	52.4101753	0.504067832
Record 12	1940	844	775.7186	68.2814032	0.656713296
Record 13	1941	1013	896.5958	116.4042	1.119546203
Record 14	1942	1286	1181.35	104.649793	1.006495286
Record 15	1943	1549	1567.921	-18.920568	-0.18197325
Record 16	1944	1583	1820	-237.00034	-2.279409443
Record 17	1945	1583	1603.927	-20.927142	-0.201271965
Record 18	1946	1671	1566.798	104.20163	1.00218497
Record 19	1947	1693	1750.896	-57.896368	-0.556832647
Record 20	1948	1763	1700.823	62.1771308	0.598003945
Record 21	1949	1744	1823.24	-79.239959	-0.762109919
Record 22	1950	1877	1707.05	169.950062	1.634536783
Record 23	1951	2080	2006.037	73.9626104	0.711353712
Record 24	1952	2207	2285.479	-78.478979	-0.754791005
Record 25	1953	2249	2329.485	-80.485253	-0.774086845

Use your mouse to select a point on the graph to compare the Actual value to the Forecasted value.

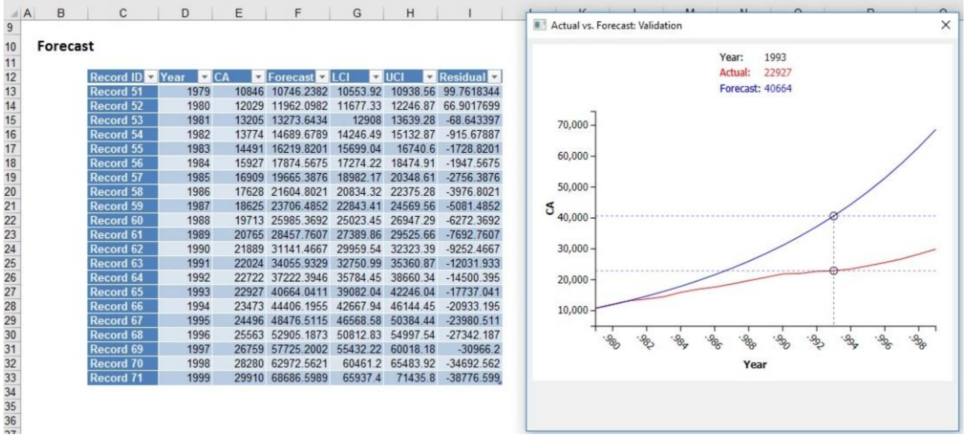


Take a look at the ACF and PACF plots for Errors found at the bottom of *ARIMA_Output*. One additional chart was added starting in V2017 - the ACVF Plot for the Residuals.



With the exception of Lag1, the majority of the lags in the PACF and ACF charts are either clearly within the UCL and LCL bands or just outside of these bands. This suggests that the residuals are random and are not correlated.

Click the Forecast link on the Output Navigator to display the Forecast Data table and charts.


Forecast

Record ID	Year	CA	Forecast	LCI	UCI	Residual
Record 51	1979	10846	10746.2382	10553.92	10938.56	99.7618344
Record 52	1980	12029	11962.0982	11677.33	12246.87	66.9017699
Record 53	1981	13205	13273.6434	12908	13639.28	-68.643397
Record 54	1982	13774	14689.6789	14246.49	15132.87	-915.67887
Record 55	1983	14491	16219.8201	15699.04	16740.6	-1728.8201
Record 56	1984	15927	17874.5675	17274.22	18474.91	-1947.5675
Record 57	1985	16909	19665.3876	18982.17	20348.61	-2756.3876
Record 58	1986	17628	21604.8021	20834.32	22375.28	-3976.8021
Record 59	1987	18625	23706.4852	22843.41	24569.56	-5081.4852
Record 60	1988	19713	25985.3692	25023.45	26947.29	-6272.3692
Record 61	1989	20765	28457.7607	27389.86	29525.66	-7692.7607
Record 62	1990	21889	31141.4667	29959.54	32323.39	-9252.4667
Record 63	1991	22024	34055.9329	32750.99	35360.87	-12031.933
Record 64	1992	22722	37222.3946	35784.45	38660.34	-14500.395
Record 65	1993	22927	40664.0411	39082.04	42246.04	-17737.041
Record 66	1994	23473	44406.1955	42667.94	46144.45	-20933.195
Record 67	1995	24496	48476.5115	46568.58	50384.44	-23980.511
Record 68	1996	25563	52905.1873	50812.83	54997.54	-27342.187
Record 69	1997	26759	57725.2002	55432.22	60018.18	-30966.2
Record 70	1998	28280	62972.5621	60461.2	65483.92	-34692.562
Record 71	1999	29910	68686.5989	65937.4	71435.8	-38776.599

The table shows the actual and forecasted values along with LCI (Lower Confidence Interval), UCI (Upper Confidence Interval) and Residual values. The "Lower" and "Upper" values represent the lower and upper bounds of the confidence interval. There is a 95% chance that the forecasted value will fall into this range. The graph to the right plots the Actual values for CA against the Forecasted values. Again, click any point on either curve to compare the Actual against the Forecasted values.

Automated Risk Analysis of Machine Learning Models

Introduction

This example illustrates the use of Analytic Solver Data Science to assess the uncertainty and risk of a machine learning (ML) model. The model is “trained” and “validated” on a total of 9,578 known cases of loan applicants, including “outcome” data on whether they defaulted or were fully repaid. Before the model is put into “production use”, Analytic Solver Data Science is used to perform, in a fully automated manner, a risk analysis of the model’s performance on 5,747 *new* cases which are individually different, but whose overall behavior is consistent with the known data. The risk analysis also includes a quantitative assessment of the range of possible loan losses, if loan decisions are made on the basis of this trained ML model.

Risk Analysis of ML Model in Predicting Loan Defaults

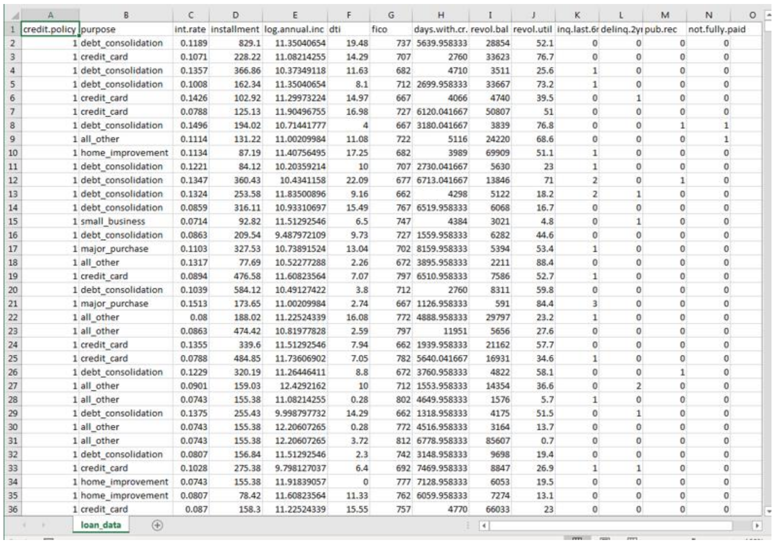
The dataset used in this example originated from LendingClub.com and is publicly available on the Kaggle machine learning website at https://www.kaggle.com/datasets/itssuru/loan-data. The “features” or independent variables in this dataset are summarized below:

1. credit.policy: 1 if the customer meets the credit underwriting criteria of LendingClub.com, and 0 otherwise.
2. purpose: The purpose of the loan (takes values "creditcard", "debtconsolidation", "educational", "majorpurchase", "smallbusiness", and "all_other").
3. int.rate: The interest rate of the loan, as a proportion (a rate of 11% would be stored as 0.11). Borrowers judged by LendingClub.com to be more risky are assigned higher interest rates.
4. installment: The monthly installments owed by the borrower if the loan is funded.
5. log.annual.inc: The natural log of the self-reported annual income of the borrower.
6. dti: The debt-to-income ratio of the borrower (amount of debt divided by annual income).
7. fico: The FICO credit score of the borrower.
8. days.with.cr.line: The number of days the borrower has had a credit line.
9. revol.bal: The borrower's revolving balance (amount unpaid at the end of the credit card billing cycle).
10. revol.util: The borrower's revolving line utilization rate (the amount of the credit line used relative to total credit available).
11. inq.last.6mths: The borrower's number of inquiries by creditors in the last 6 months.
12. delinq.2yrs: The number of times the borrower had been 30+ days past due on a payment in the past 2 years.

13. pub.rec: The borrower's number of derogatory public records (bankruptcy filings, tax liens, or judgments).
14. not.fully.paid: 1 if the borrower defaulted and did not [fully] pau off the loan.

In this example, the risk analysis reveals that the first machine learning model, using the Classification Tree ML method, has a high risk of poor performance on future cases. As a result a second ML model is fit using the Logistic Regression ML model. In this case the risk analysis shows lower risk of poor results, but still yields a ***quantification*** of that ***risk***, not available before in Analytic Solver Data Science.

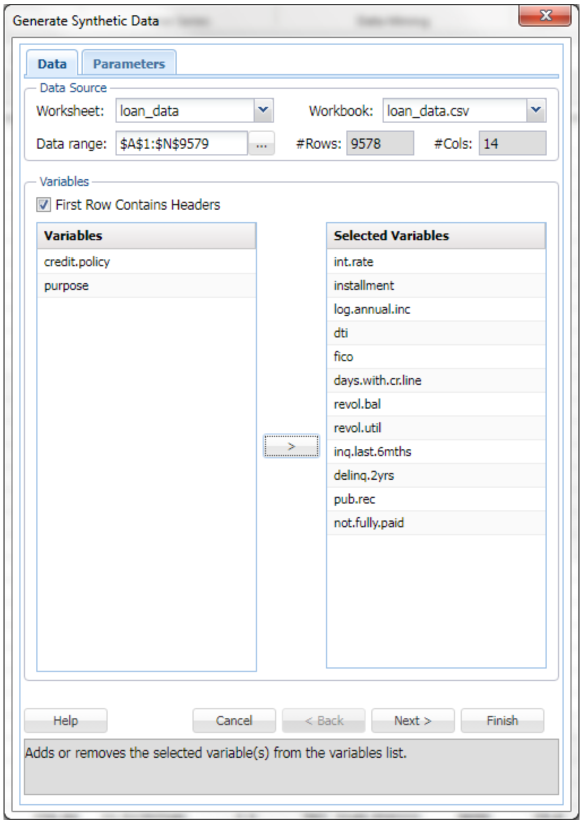
The first figure shows the loan dataset loaded into Microsoft Excel, for use with Analytic Solver Data Science for Excel.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.	revol.bal	revol.util	inq.last.6r	delinq.2yı	pub.rec	not.fully.paid	
2	1	debt_consolidation	0.1189	829.1	11.35040654	19.48	737	5639.958333	28854	52.1	0	0	0	0	
3	1	credit_card	0.1071	228.22	11.08214255	14.29	707	2760	33623	76.7	0	0	0	0	
4	1	debt_consolidation	0.1357	366.86	10.37349118	11.63	682	4710	3511	25.6	1	0	0	0	
5	1	debt_consolidation	0.1008	162.34	11.35040654	8.1	712	2699.958333	33667	73.2	1	0	0	0	
6	1	credit_card	0.1426	102.92	11.29973224	14.97	667	4066	4740	39.5	0	1	0	0	
7	1	credit_card	0.0788	125.13	11.90496755	16.98	727	6120.041667	50807	51	0	0	0	0	
8	1	debt_consolidation	0.1496	194.02	10.71441777	4	667	3180.041667	3839	76.8	0	0	1	1	
9	1	all_other	0.1114	131.22	11.00209984	11.08	722	5116	24220	68.6	0	0	0	1	
10	1	home_improvement	0.1134	87.19	11.40756495	17.25	682	3989	69909	51.1	1	0	0	0	
11	1	debt_consolidation	0.1221	84.12	10.20359214	10	707	2730.041667	5630	23	1	0	0	0	
12	1	debt_consolidation	0.1347	360.43	10.4341158	22.09	677	6713.041667	13846	71	2	0	1	0	
13	1	debt_consolidation	0.1324	253.58	11.83500896	9.16	662	4298	5122	18.2	2	1	0	0	
14	1	debt_consolidation	0.0859	316.11	10.93310697	15.49	767	6519.958333	6068	16.7	0	0	0	0	
15	1	small_business	0.0714	92.82	11.51292546	6.5	747	4384	3021	4.8	0	1	0	0	
16	1	debt_consolidation	0.0863	209.54	9.487972109	9.73	727	1559.958333	6282	44.6	0	0	0	0	
17	1	major_purchase	0.1103	327.53	10.73891524	13.04	702	8159.958333	5394	53.4	1	0	0	0	
18	1	all_other	0.1317	77.69	10.52277288	2.26	672	3895.958333	2211	88.4	0	0	0	0	
19	1	credit_card	0.0894	476.58	11.60823564	7.07	797	6510.958333	7586	52.7	1	0	0	0	
20	1	debt_consolidation	0.1039	584.12	10.49127422	3.8	712	2760	8311	59.8	0	0	0	0	
21	1	major_purchase	0.1513	173.65	11.00209984	2.74	667	1126.958333	591	84.4	3	0	0	0	
22	1	all_other	0.08	188.02	11.22524339	16.08	772	4888.958333	29797	23.2	1	0	0	0	
23	1	all_other	0.0863	474.42	10.81977828	2.59	797	11951	5656	27.6	0	0	0	0	
24	1	credit_card	0.1355	339.6	11.51292546	7.94	662	1939.958333	21162	57.7	0	0	0	0	
25	1	credit_card	0.0788	484.85	11.73606902	7.05	782	5640.041667	16931	34.6	1	0	0	0	
26	1	debt_consolidation	0.1229	320.19	11.26446411	8.8	672	3760.958333	4822	58.1	0	0	1	0	
27	1	all_other	0.0901	159.03	12.4292162	10	712	1553.958333	14354	36.6	0	2	0	0	
28	1	all_other	0.0743	155.38	11.08214255	0.28	802	4649.958333	1576	5.7	1	0	0	0	
29	1	debt_consolidation	0.1375	255.43	9.998797732	14.29	662	1318.958333	4175	51.5	0	1	0	0	
30	1	all_other	0.0743	155.38	12.20607265	0.28	772	4516.958333	3164	13.7	0	0	0	0	
31	1	all_other	0.0743	155.38	12.20607265	3.72	812	6778.958333	85607	0.7	0	0	0	0	
32	1	debt_consolidation	0.0807	156.84	11.51292546	2.3	742	3148.958333	9698	19.4	0	0	0	0	
33	1	credit_card	0.1028	275.38	9.798127037	6.4	692	7469.958333	8847	26.9	1	1	0	0	
34	1	home_improvement	0.0743	155.38	11.91839057	0	777	7128.958333	6053	19.5	0	0	0	0	
35	1	home_improvement	0.0807	78.42	11.60823564	11.33	762	6059.958333	7274	13.1	0	0	0	0	
36	1	credit_card	0.087	158.3	11.22524339	15.55	757	4770	66033	23	0	0	0	0	

Preliminary Illustration to Show Synthetic Data Compared to Known Data

We select the 12 numeric/continuous “features” in this dataset, omitting the “credit.policy” and “purpose” (categorical and text) features, and click the “Next” button at the button of the dialog:


Generate Synthetic Data

Data | Parameters

Data Source

Worksheet: loan_data

Workbook: loan_data.csv

Data range: $A$1:$N$9579

#Rows: 9578

#Cols: 14

Variables

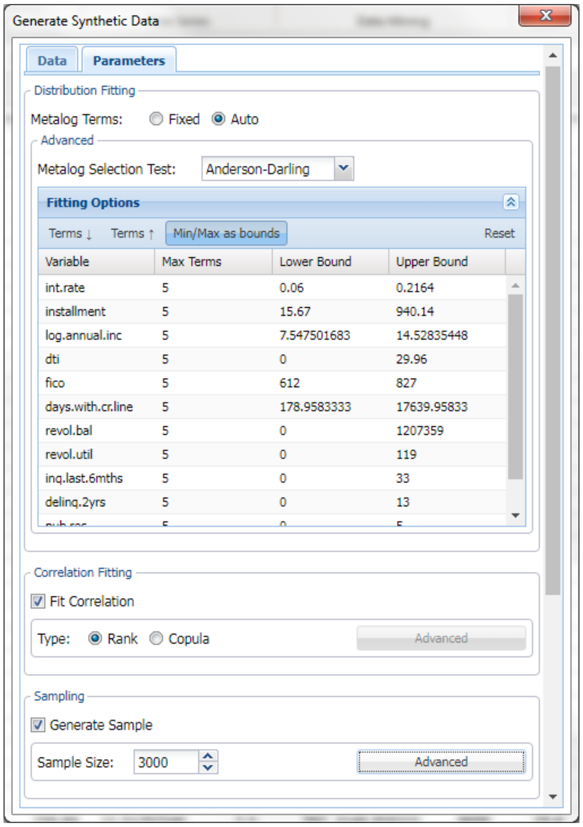
☑ First Row Contains Headers

Variables	Selected Variables
credit.policy	int.rate
purpose	installment
	log.annual.inc
	dti
	fico
	days.with.cr.line
	revol.bal
	revol.util
	inq.last.6mths
	delinq.2yrs
	pub.rec
	not.fully.paid

Adds or removes the selected variable(s) from the variables list.

The next two images, from the same step "Parameters" dialog, illustrate settings that the user can ***optionally*** modify to further control the simulation and synthetic data generation process; however, in most applications the user need not change anything here, and can simply click the "Next" or "Finish" buttons.

For more information on all Classification Tree options, see the Classification Tree chapter within the Data Science Reference Guide.



Generate Synthetic Data

Distribution Fitting

Metalog Terms: ○ Fixed ● Auto

Advanced

Metalog Selection Test: Anderson-Darling

Fitting Options

Variable	Max Terms	Lower Bound	Upper Bound
int.rate	5	0.06	0.2164
installment	5	15.67	940.14
log.annual.inc	5	7.547501683	14.52835448
dti	5	0	29.96
fico	5	612	827
days.with.cr.line	5	178.9583333	17639.95833
revol.bal	5	0	1207359
revol.util	5	0	119
inq.last.6mths	5	0	33
delinq.2yrs	5	0	13
pub.rec	5	0	5

Correlation Fitting

☑ Fit Correlation

Type: ● Rank ○ Copula

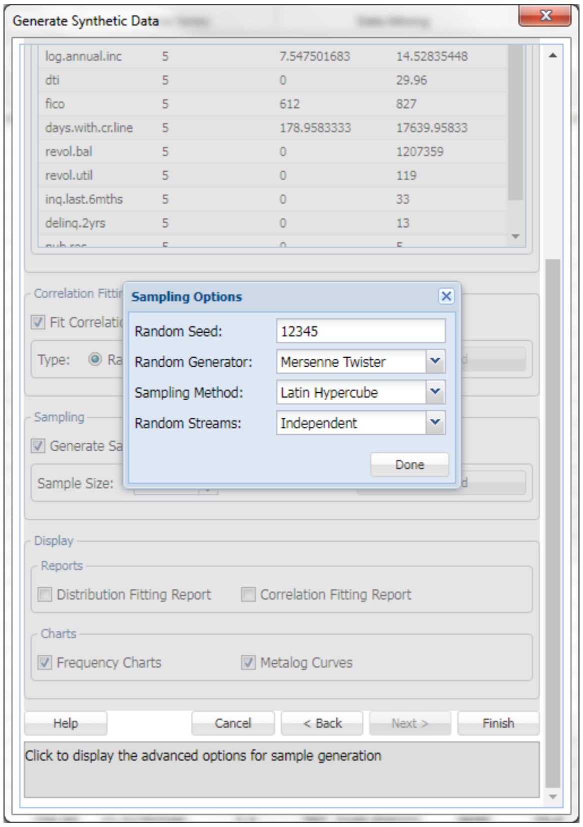
Sampling

☑ Generate Sample

Sample Size: 3000

Above, we've expanded the "Fitting Options", which control the automated fitting of Metalog probability distributions to each feature. For example, by default, bounds in the dataset are used as bounds for the family of Metalog distributions, but users can easily and quickly adjust or remove the lower and/or upper bounds. "Correlation Fitting" will by default construct a rank-order correlation matrix that includes all of the features, but choosing "Copula" activates the "Advanced" button, where types of copulas (Clayton, Frank, Gumbel, etc.) can be chosen.

For more information on these options, see the Generate Data chapter found within the Data Science Reference Guide.



Generate Synthetic Data

log.annual.inc	5	7.547501683	14.52835448
dti	5	0	29.96
fico	5	612	827
days.with.cr.line	5	178.9583333	17639.95833
revol.bal	5	0	1207359
revol.util	5	0	119
inq.last.6mths	5	0	33
delinq.2yrs	5	0	13

Sampling Options

Random Seed: 12345

Random Generator: Mersenne Twister

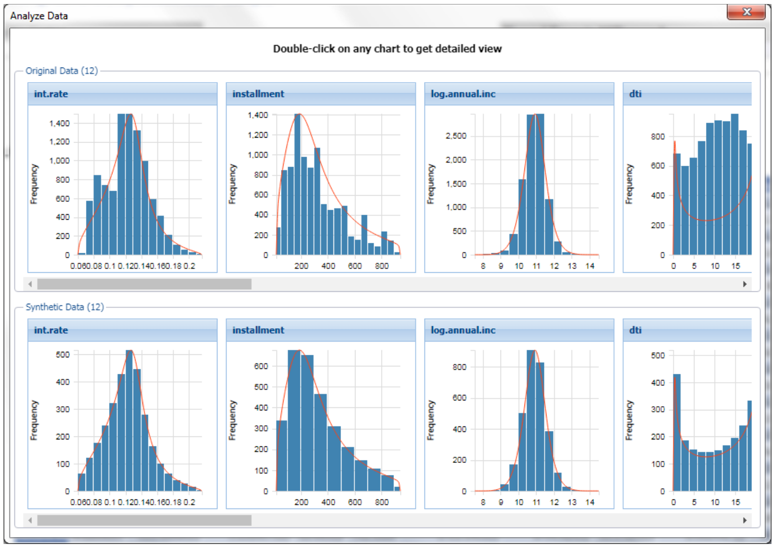
Sampling Method: Latin Hypercube

Random Streams: Independent

Click to display the advanced options for sample generation

Above, we’re illustrating the lower portion of the same “Parameters” dialog, with further options including the “Sampling Options” shown. Again, in typical applications the user can simply click the “Next” or “Finish” buttons, without choosing anything here.

The next image shows frequency histograms of the ***known*** cases versus the ***simulated*** or synthetically generated cases: It is the ***differences*** (rather than the similarities) in this data that give rise to the uncertainty and risk in the performance of the machine learning model.



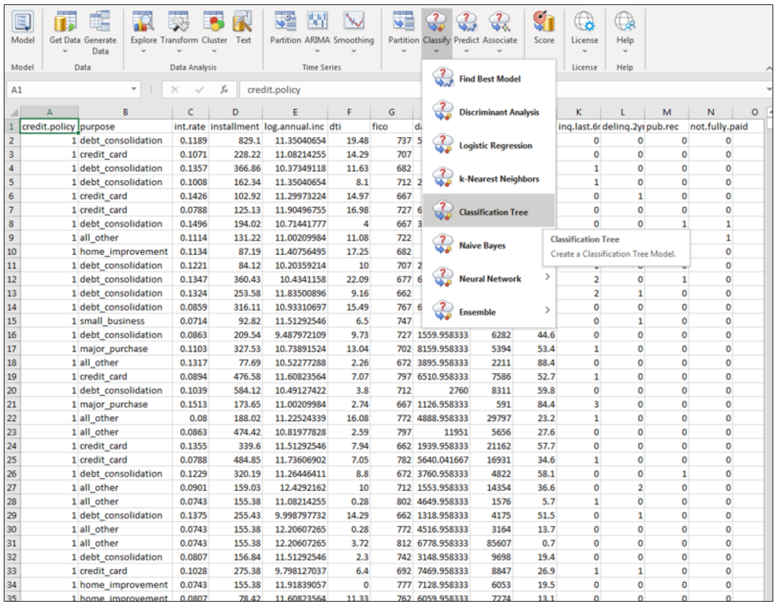
Analyze Data

Double-click on any chart to get detailed view

In the above frequency histograms, the original data (known cases) appear in the top row, and the simulated or synthetically generated cases appear in the bottom row. Note that the distribution of "dti" (debt to income ratio) is ***different*** in the simulated cases, with more frequent cases having a higher debt to income ratio.

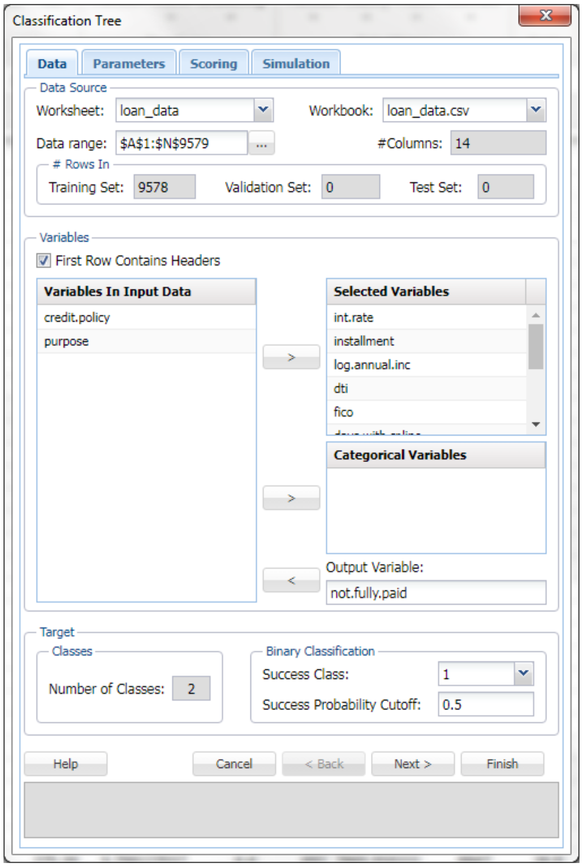
Illustration: Automated Risk Analysis of Just-Trained Machine Learning Model

With the above as background, the fully automated risk analysis process is illustrated. First, a "Classification Tree", a typical ML model based on the CART methodology is built:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	d			inq.last.6r	delinq.2yr	pub.rec	not.fully.paid	
2	1	debt_consolidation	0.1189	829.1	11.35040654	19.48	737	5			0	0	0	0	
3	1	credit_card	0.1071	228.22	11.08214255	14.29	707				0	0	0	0	
4	1	debt_consolidation	0.1357	366.86	10.37349118	11.63	682				1	0	0	0	
5	1	debt_consolidation	0.1008	162.34	11.35040654	8.1	712	2			1	0	0	0	
6	1	credit_card	0.1426	102.92	11.29973224	14.97	667				0	1	0	0	
7	1	credit_card	0.0788	125.13	11.90496755	16.98	727	6			0	0	0	0	
8	1	debt_consolidation	0.1496	194.02	10.71441777	4	667	3			0	0	1	1	
9	1	all_other	0.1114	131.22	11.00209984	11.08	722							1	
10	1	home_improvement	0.1134	87.19	11.40756495	17.25	682							0	
11	1	debt_consolidation	0.1221	84.12	10.20359214	10	707	2						0	
12	1	debt_consolidation	0.1347	360.43	10.4341158	22.09	677	6			2	0	1	0	
13	1	debt_consolidation	0.1324	253.58	11.83500896	9.16	662				2	1	0	0	
14	1	debt_consolidation	0.0859	316.11	10.93310697	15.49	767	6			0	0	0	0	
15	1	small_business	0.0714	92.82	11.51292546	6.5	747				0	1	0	0	
16	1	debt_consolidation	0.0863	209.54	9.487972109	9.73	727	1559.958333	6282	44.6	0	0	0	0	
17	1	major_purchase	0.1103	327.53	10.73891524	13.04	702	8159.958333	5394	53.4	1	0	0	0	
18	1	all_other	0.1317	77.69	10.52277288	2.26	672	3895.958333	2211	88.4	0	0	0	0	
19	1	credit_card	0.0894	476.58	11.60823564	7.07	797	6510.958333	7586	52.7	1	0	0	0	
20	1	debt_consolidation	0.1039	584.12	10.49127422	3.8	712	2760	8311	59.8	0	0	0	0	
21	1	major_purchase	0.1513	173.65	11.00209984	2.74	667	1126.958333	591	84.4	3	0	0	0	
22	1	all_other	0.08	188.02	11.22524339	16.08	772	4888.958333	29797	23.2	1	0	0	0	
23	1	all_other	0.0863	474.42	10.81977828	2.59	797	11951	5656	27.6	0	0	0	0	
24	1	credit_card	0.1355	339.6	11.51292546	7.94	662	1939.958333	21162	57.7	0	0	0	0	
25	1	credit_card	0.0788	484.85	11.73606902	7.05	782	5640.041667	16931	34.6	1	0	0	0	
26	1	debt_consolidation	0.1229	320.19	11.26446411	8.8	672	3760.958333	4822	58.1	0	0	1	0	
27	1	all_other	0.0901	159.03	12.4292162	10	712	1553.958333	14354	36.6	0	2	0	0	
28	1	all_other	0.0743	155.38	11.08214255	0.28	802	4649.958333	1576	5.7	1	0	0	0	
29	1	debt_consolidation	0.1375	255.43	9.998797732	14.29	662	1318.958333	4175	51.5	0	1	0	0	
30	1	all_other	0.0743	155.38	12.20607265	0.28	772	4516.958333	3164	13.7	0	0	0	0	
31	1	all_other	0.0743	155.38	12.20607265	3.72	812	6778.958333	85607	0.7	0	0	0	0	
32	1	debt_consolidation	0.0807	156.84	11.51292546	2.3	742	3148.958333	9698	19.4	0	0	0	0	
33	1	credit_card	0.1028	275.38	9.798127037	6.4	692	7469.958333	8847	26.9	1	1	0	0	
34	1	home_improvement	0.0743	155.38	11.91839057	0	777	7128.958333	6053	19.5	0	0	0	0	
35	1	home_improvement	0.0807	78.42	11.60823564	11.33	762	6059.958333	7274	13.1	0	0	0	0	

In the first step, the features to be used are selected.



Worksheet: loan_data

Workbook: loan_data.csv

Data range: $A$1:$N$9579

#Columns: 14

Training Set: 9578

Validation Set: 0

Test Set: 0

☑ First Row Contains Headers

Variables In Input Data
credit.policy
purpose

Selected Variables
int.rate
installment
log.annual.inc
dti
fico

Categorical Variables

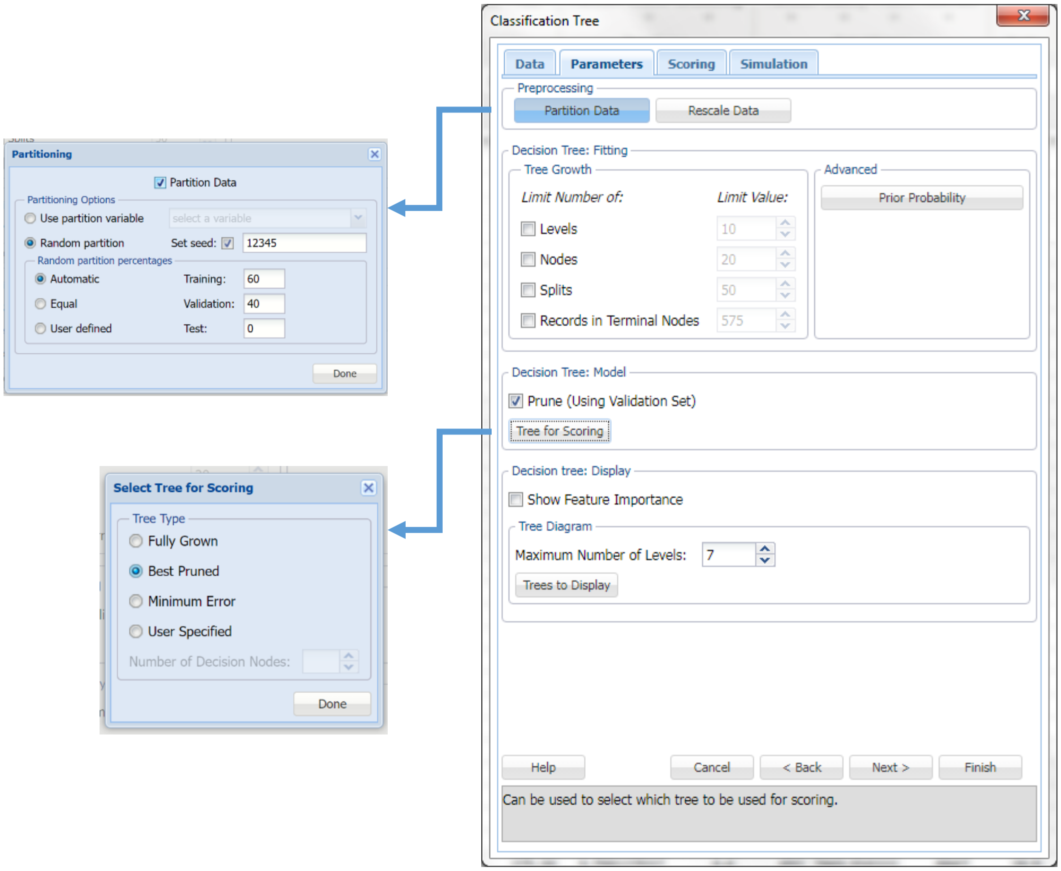
Output Variable: not.fully.paid

Number of Classes: 2

Success Class: 1

Success Probability Cutoff: 0.5

Click the “Next” button to select options for a Classification Tree.



Classification Tree

Data | Parameters | Scoring | Simulation

Preprocessing

Partition Data | Rescale Data

Decision Tree: Fitting

Tree Growth

Limit Number of:	*Limit Value:*
□ Levels	10
□ Nodes	20
□ Splits	50
□ Records in Terminal Nodes	575

Advanced

Prior Probability

Decision Tree: Model

☑ Prune (Using Validation Set)

Tree for Scoring

Decision tree: Display

□ Show Feature Importance

Tree Diagram

Maximum Number of Levels: 7

Trees to Display

Help | Cancel | < Back | Next > | Finish

Can be used to select which tree to be used for scoring.

Partitioning

☑ Partition Data

Partitioning Options

○ Use partition variable | select a variable

● Random partition | Set seed: ☑ 12345

Random partition percentages

● Automatic | Training: 60

○ Equal | Validation: 40

○ User defined | Test: 0

Done

Select Tree for Scoring

Tree Type

○ Fully Grown

● Best Pruned

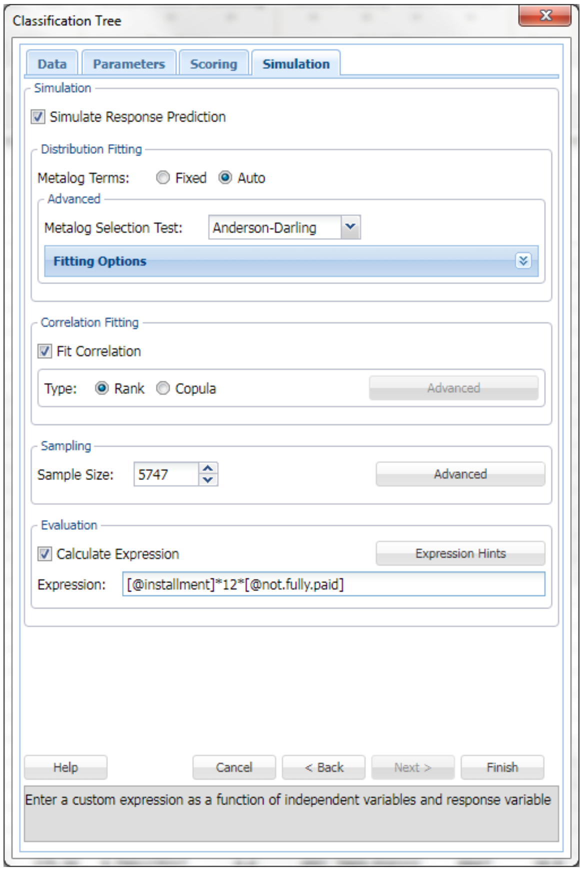
○ Minimum Error

○ User Specified

Number of Decision Nodes:

Done

The 9,578 row dataset is first partitioned into a “Training Set” with 60% of the data (5,747 rows), and a “Validation Set” with 40% of the data by clicking Partition Data in the Preprocessing section of the Parameters tab. Select “Prune (Using the Validation Set)”, and then click the Tree for Scoring button to select “Best Pruned” to use the Pruned Tree for Scoring. Then click the “Next” button twice to advance to the Simulation tab. No options will be changed on the Scoring tab.)



☑ Simulate Response Prediction

Metalog Terms: ○ Fixed ● Auto

Metalog Selection Test: Anderson-Darling

☑ Fit Correlation

Type: ● Rank ○ Copula

Sample Size: 5747

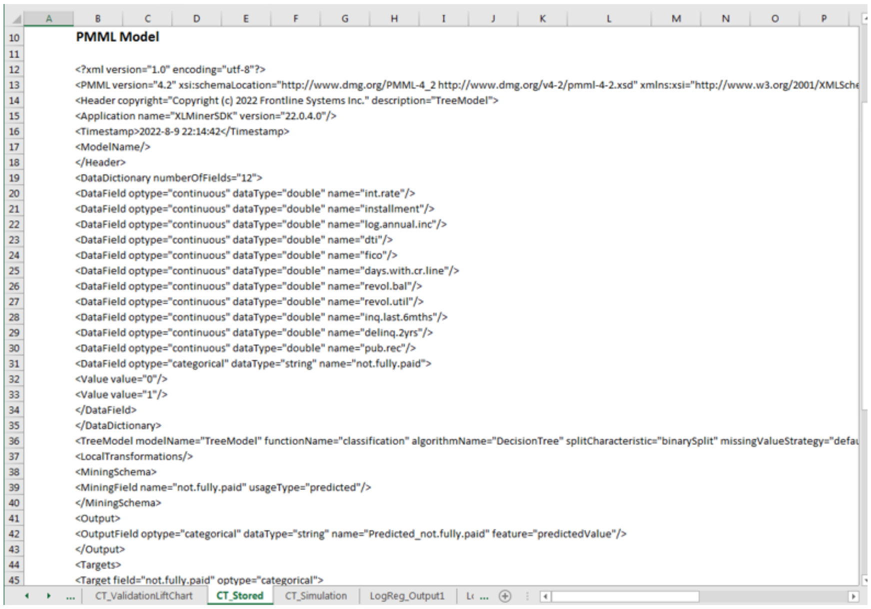
☑ Calculate Expression

Expression: [@installment]*12*[@not.fully.paid]

Enter a custom expression as a function of independent variables and response variable

Options for the ***entire risk analysis*** of the ***just-trained*** and validated machine learning model ***"on the fly"*** are set on the Simulation tab (shown above). This example asks for 5,747 simulated cases (the same number as in the Training Set, for convenience) and enters a "financial payoff" calculation – via commonly-used Microsoft Excel Table Expression syntax, which calculates a loss of 12 times the monthly installment if the borrower defaults on the loan. Results will appear within moments after clicking the "Finish" button.

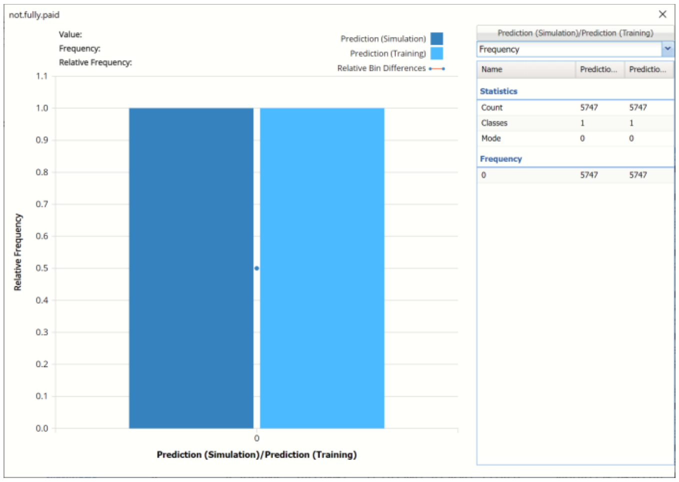
Analytic Solver Data Science now performs the "training" and "validation" steps to create a Classification Tree ML model with parameters fitted to best predict or classify the training set known cases, then validates and refines the model based on the validation set known cases. The trained model is rendered in PMML (Predictive Modeling Markup Language, an industry standard across different software tools).



PMML Model

```
<?xml version="1.0" encoding="utf-8"?>
<PMML version="4.2" xsi:schemaLocation="http://www.dmg.org/PMML-4_2 http://www.dmg.org/v4-2/pmml-4-2.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSche
<Header copyright="Copyright (c) 2022 Frontline Systems Inc." description="TreeModel">
<Application name="XLMinerSDK" version="22.0.4.0"/>
<Timestamp>2022-8-9 22:14:42</Timestamp>
<ModelName/>
</Header>
<DataDictionary numberOfFields="12">
<DataField optype="continuous" dataType="double" name="int.rate"/>
<DataField optype="continuous" dataType="double" name="installment"/>
<DataField optype="continuous" dataType="double" name="log.annual.inc"/>
<DataField optype="continuous" dataType="double" name="dti"/>
<DataField optype="continuous" dataType="double" name="fico"/>
<DataField optype="continuous" dataType="double" name="days.with.cr.line"/>
<DataField optype="continuous" dataType="double" name="revol.bal"/>
<DataField optype="continuous" dataType="double" name="revol.util"/>
<DataField optype="continuous" dataType="double" name="inq.last.6mths"/>
<DataField optype="continuous" dataType="double" name="delinq.2yrs"/>
<DataField optype="continuous" dataType="double" name="pub.rec"/>
<DataField optype="categorical" dataType="string" name="not.fully.paid">
<Value value="0"/>
<Value value="1"/>
</DataField>
</DataDictionary>
<TreeModel modelName="TreeModel" functionName="classification" algorithmName="DecisionTree" splitCharacteristic="binarySplit" missingValueStrategy="defau
<LocalTransformations/>
<MiningSchema>
<MiningField name="not.fully.paid" usageType="predicted"/>
</MiningSchema>
<Output>
<OutputField optype="categorical" dataType="string" name="Predicted_not.fully.paid" feature="predictedValue"/>
</Output>
<Targets>
<Target field="not.fully.paid" optype="categorical">
```

The surprise – and financial consequences – appear on the “CT_Simulation” tab next to the “CT_Stored” (stored PMML model) tab in the Excel workbook. The simulated data, ML model predictions on that data, and financial consequences are calculated and stored on this “CT_Simulation” worksheet; clicking that tab displays the chart shown below (in one of its variants).

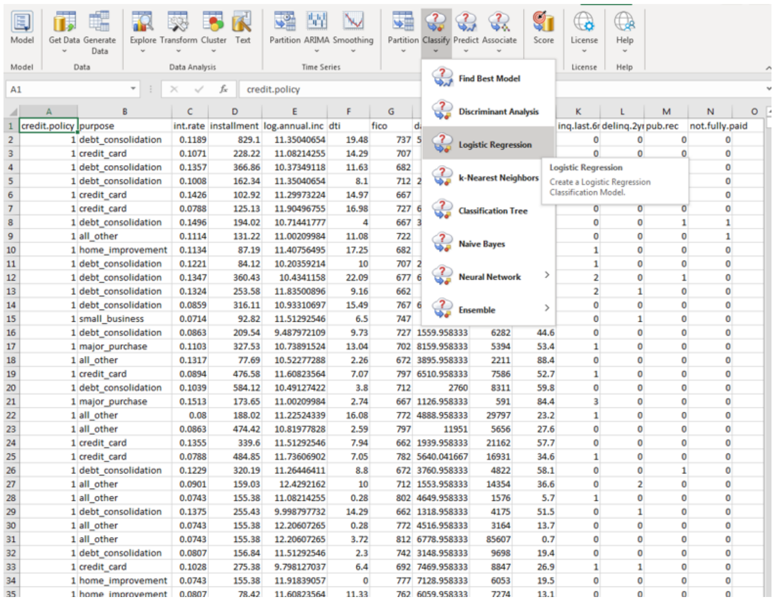


Name	Predictio...	Predictio...
Statistics		
Count	5747	5747
Classes	1	1
Mode	0	0
Frequency		
0	5747	5747

The light blue bar displays the frequencies of “no default” (0) and “default” (1) on the 5,747 known cases in the Training Set. The dark blue bar displays the same frequencies for the simulated data: This shows that the Classification Tree model is ***unlikely*** to predict (or classify) ***any*** future cases as likely to “default”. There is a risk that the model’s future predictive power is not very good at all.

[This is an illustration of an “extreme” case of poor performance, due in part to selecting and training only one type of ML model, which is then “caught” by performing the risk analysis; a more thorough preparation, data exploration, selection and training of alternative models could have yielded better results – but even then, as shown below, the user would lack any quantitative information about the ***uncertainty*** and ***risk*** in the finally-chosen ML model.]

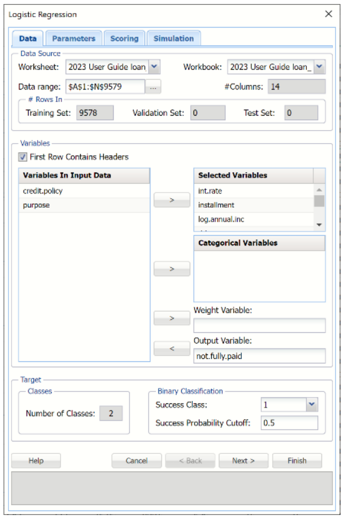
In this example, re-examination of the Classification Tree's performance leads the user to try a different statistical / machine learning approach, Logistic Regression:



A1 credit.policy

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	d			inq.last.6r	delinq.2yr	pub.rec	not.fully.paid
2	1	debt_consolidation	0.1189	829.1	11.35040654	19.48	737	5			0	0	0	0
3	1	credit_card	0.1071	228.22	11.08214255	14.29	707				0	0	0	0
4	1	debt_consolidation	0.1357	366.86	10.37349118	11.63	682							0
5	1	debt_consolidation	0.1008	162.34	11.35040654	8.1	712	2						0
6	1	credit_card	0.1426	102.92	11.29973224	14.97	667							0
7	1	credit_card	0.0788	125.13	11.90496755	16.98	727	6			0	0	0	0
8	1	debt_consolidation	0.1496	194.02	10.71441777	4	667	3			0	0	1	1
9	1	all_other	0.1114	131.22	11.00209984	11.08	722				0	0	0	1
10	1	home_improvement	0.1134	87.19	11.40756495	17.25	682				1	0	0	0
11	1	debt_consolidation	0.1221	84.12	10.20359214	10	707	2			1	0	0	0
12	1	debt_consolidation	0.1347	360.43	10.4341158	22.09	677	6			2	0	1	0
13	1	debt_consolidation	0.1324	253.58	11.83500896	9.16	662				2	1	0	0
14	1	debt_consolidation	0.0859	316.11	10.93310697	15.49	767	6			0	0	0	0
15	1	small_business	0.0714	92.82	11.51292546	6.5	747				0	1	0	0
16	1	debt_consolidation	0.0863	209.54	9.487972109	9.73	727	1559.958333	6282	44.6	0	0	0	0
17	1	major_purchase	0.1103	327.53	10.73891524	13.04	702	8159.958333	5394	53.4	1	0	0	0
18	1	all_other	0.1317	77.69	10.52277288	2.26	672	3895.958333	2211	88.4	0	0	0	0
19	1	credit_card	0.0894	476.58	11.60823564	7.07	797	6510.958333	7586	52.7	1	0	0	0
20	1	debt_consolidation	0.1039	584.12	10.49127422	3.8	712	2760	8311	59.8	0	0	0	0
21	1	major_purchase	0.1513	173.65	11.00209984	2.74	667	1126.958333	591	84.4	3	0	0	0
22	1	all_other	0.08	188.02	11.22524339	16.08	772	4888.958333	29797	23.2	1	0	0	0
23	1	all_other	0.0863	474.42	10.81977828	2.59	797	11951	5656	27.6	0	0	0	0
24	1	credit_card	0.1355	339.6	11.51292546	7.94	662	1939.958333	21162	57.7	0	0	0	0
25	1	credit_card	0.0788	484.85	11.73606902	7.05	782	5640.041667	16931	34.6	1	0	0	0
26	1	debt_consolidation	0.1229	320.19	11.26446411	8.8	672	3760.958333	4822	58.1	0	0	1	0
27	1	all_other	0.0901	159.03	12.4292162	10	712	1553.958333	14354	36.6	0	2	0	0
28	1	all_other	0.0743	155.38	11.08214255	0.28	802	4649.958333	1576	5.7	1	0	0	0
29	1	debt_consolidation	0.1375	255.43	9.998797732	14.29	662	1318.958333	4175	51.5	0	1	0	0
30	1	all_other	0.0743	155.38	12.20607265	0.28	772	4516.958333	3164	13.7	0	0	0	0
31	1	all_other	0.0743	155.38	12.20607265	3.72	812	6778.958333	85607	0.7	0	0	0	0
32	1	debt_consolidation	0.0807	156.84	11.51292546	2.3	742	3148.958333	9698	19.4	0	0	0	0
33	1	credit_card	0.1028	275.38	9.798127037	6.4	692	7469.958333	8847	26.9	1	1	0	0
34	1	home_improvement	0.0743	155.38	11.91839057	0	777	7128.958333	6053	19.5	0	0	0	0
35	1	home_improvement	0.0807	78.42	11.60823564	11.33	762	6059.958333	7274	13.1	0	0	0	0

A very similar sequence of steps, selecting features of the dataset, partitioning the data into Training and Validation Sets as before, and choosing exactly the same parameters (sample size and the financial calculation) on the Simulation tab.

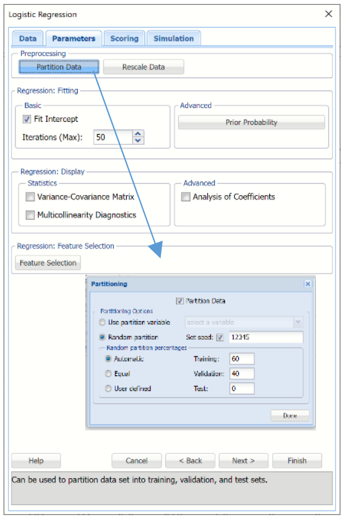


Data Source
Worksheet: 2023 User Guide loan
Workbook: 2023 User Guide loan_
Data range: $A$1:$N$9579
#Columns: 14
Rows In
Training Set: 9578
Validation Set: 0
Test Set: 0

Variables
☑ First Row Contains Headers

Variables In Input Data: credit.policy, purpose
Selected Variables: int.rate, installment, log.annual.inc
Categorical Variables:
Weight Variable:
Output Variable: not.fully.paid

Target
Classes
Number of Classes: 2
Binary Classification
Success Class: 1
Success Probability Cutoff: 0.5



Preprocessing

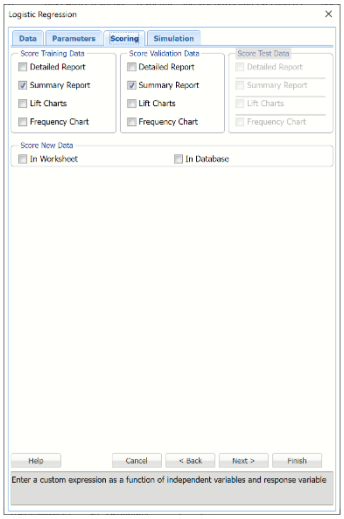
Regression: Fitting
Basic
☑ Fit Intercept
Iterations (Max): 50
Advanced

Regression: Display
Statistics
☐ Variance-Covariance Matrix
☐ Multicollinearity Diagnostics
Advanced
☐ Analysis of Coefficients

Regression: Feature Selection

Partitioning
☑ Partition Data
Partitioning Options
○ Use partition variable   select a variable
● Random partition   Set seed: ☑ 12345
Random partition percentages
● Automatic   Training: 60
○ Equal   Validation: 40
○ User defined   Test: 0

Can be used to partition data set into training, validation, and test sets.



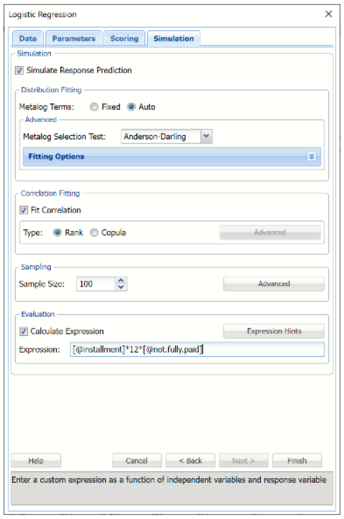
Score Training Data
☐ Detailed Report
☑ Summary Report
☐ Lift Charts
☐ Frequency Chart

Score Validation Data
☐ Detailed Report
☑ Summary Report
☐ Lift Charts
☐ Frequency Chart

Score Test Data
☐ Detailed Report
☐ Summary Report
☐ Lift Charts
☐ Frequency Chart

Score New Data
☐ In Worksheet
☐ In Database

Enter a custom expression as a function of independent variables and response variable



Simulation
☑ Simulate Response Prediction

Distribution Fitting
Metalog Terms: ○ Fixed ● Auto
Advanced
Metalog Selection Test: Anderson-Darling

Correlation Fitting
☑ Fit Correlation
Type: ● Rank ○ Copula

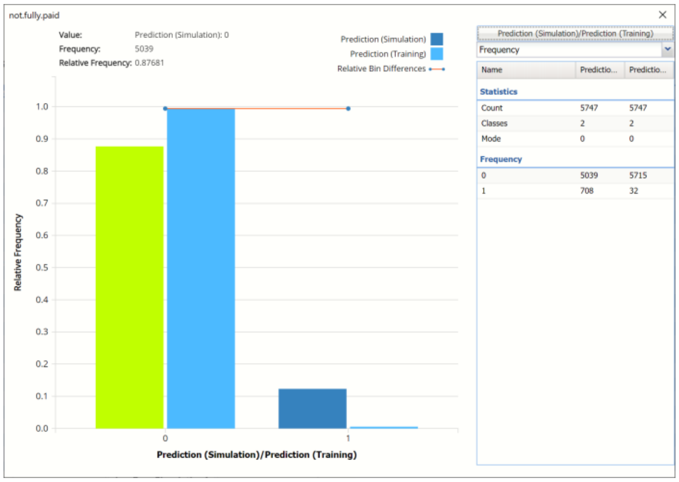
Sampling
Sample Size: 100

Evaluation
☑ Calculate Expression
Expression: [@installment]*12*[@not.fully.paid]

Enter a custom expression as a function of independent variables and response variable

When Finish is clicked on the simulation tab, Analytic Solver Data Science again performs the "training" and "validation" steps to create a Logistic Regression ML model with parameters fitted to the data, and again renders the model in PMML (Predictive Modeling Markup Language) form.

This time, the chart on the Logreg_Simulation tab in the Excel workbook shows quite different results:



Value: Prediction (Simulation): 0
Frequency: 5039
Relative Frequency: 0.87681

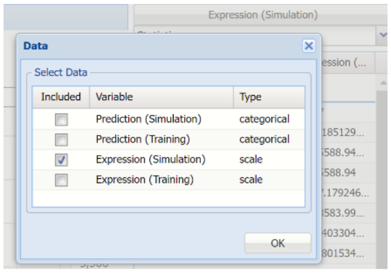
Prediction (Simulation)/Prediction (Training)

Frequency

Name	Predictio...	Predictio...
Statistics		
Count	5747	5747
Classes	2	2
Mode	0	0
Frequency		
0	5039	5715
1	708	32

Again, the light blue bars display the frequencies of "no default" (0) and "default" (1) on the 5,747 predicted cases in the Training Set. The dark blue bars display the predicted values in the simulated data: This shows that the Logistic Regression model's future predictive power is quite good: It is predicting slightly more cases of "default" (1) and fewer cases of "no default". Note the red Relative Bin Differences line that indicates that the absolute differences between the "no fault" and "default" bins is 0. Click the down arrow next to Frequency and select Bin Details for more information.

While the risk analysis of the Classification Tree model could "save" the user from an expensive error if he/she had put the Classification Tree model into production use, the much-better Logistic Regression model still has some uncertainty and risk. By clicking the button in the top right of the above chart, currently labeled "Prediction (Simulation) / Prediction (Training)", and selecting instead "Expression (Simulation)", the user can gain a ***better, quantified*** understanding of this ***uncertainty and risk***:

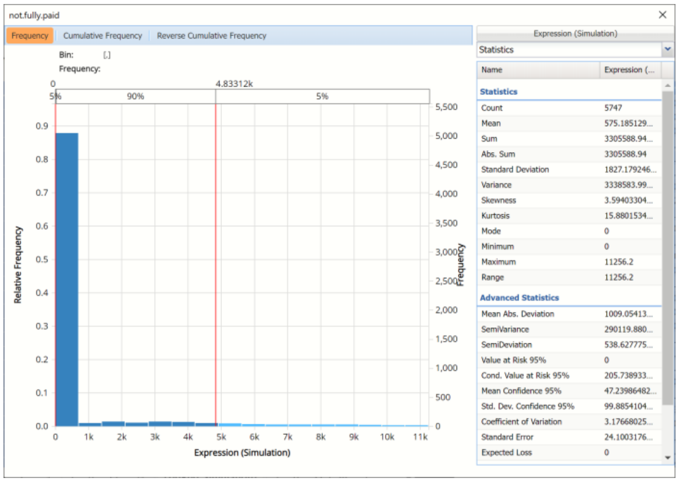


Data

Select Data

Included	Variable	Type
□	Prediction (Simulation)	categorical
□	Prediction (Training)	categorical
☑	Expression (Simulation)	scale
□	Expression (Training)	scale

OK



Bin: [,]
Frequency:

Expression (Simulation)

Statistics

Name	Expression (...
Statistics	
Count	5747
Mean	575.185129...
Sum	3305588.94...
Abs. Sum	3305588.94
Standard Deviation	1827.179246...
Variance	3338583.99...
Skewness	3.59403304...
Kurtosis	15.8801534...
Mode	0
Minimum	0
Maximum	11256.2
Range	11256.2
Advanced Statistics	
Mean Abs. Deviation	1009.05413...
SemiVariance	290119.880...
SemiDeviation	538.627775...
Value at Risk 95%	0
Cond. Value at Risk 95%	205.738933...
Mean Confidence 95%	47.23986482...
Std. Dev. Confidence 95%	99.8854104...
Coefficient of Variation	3.17668025...
Standard Error	24.1003176...
Expected Loss	0

This form of the chart – still a frequency histogram, but of the user-specified ***financial consequence*** – shows that in most cases the loan won't default and the loss will be $0, but *across all cases* the loss will average $575 (the Mean value) and may be as large as $11,256 (the Maximum value) in some cases. Other statistics such as Standard Deviation and Conditional Value at Risk give a quantitatively-oriented business manager some further insights into the model's performance. The insights are available ***before*** the model is put into production use, and can be useful in determining how and when the model's performance should be evaluated, and the model potentially re-trained in the future.

Fitting the Best Model

Introduction

Analytic Solver Data Science includes comprehensive, powerful support for data science and machine learning. Using these tools, you can "train" or fit your data to a wide range of statistical and machine learning models: Classification and regression trees, neural networks, linear and logistic regression, discriminant analysis, naïve Bayes, k-nearest neighbors and more. But the task of choosing and comparing these models, and selecting parameters for each one was up to you.

With the Find Best Model options, you can automate this work as well! Find Best Model uses methods similar to those in (expensive high-end) tools like DataRobot and RapidMiner, to automatically choose types of ML models and their parameters, validate and compare them according to criteria that you choose, and deliver the model that best fits your data.

Continue reading to discover how to utilize Find Best Model through an easy-to-follow example. For more information on how to run each classification or regression learner independently of Find Best Model, see the Data Science Reference Guide.

Find Best Model Classification Example

This example illustrates how to utilize the Find Best Model method for Classification, included in Analytic Solver Data Science for Desktop Excel or Excel Online, by using the Heart Failure Clinical Records Dataset[6]. This dataset contains thirteen variables describing 299 patients experiencing heart failure. Find Best Model fits a model to all selected supervised learning (classification) methods in order to observe which method provides the best fit to the data. The goal of this example is to fit the best model to the dataset, then use this fitted model to determine if a new patient is at risk of perishing due to heart failure.

A description of each variable contained in the dataset appears in the table below.

VARIABLE	DESCRIPTION
AGE	Age of patient
ANAEMIA	Decrease of red blood cells or hemoglobin (boolean)
CREATINE_PHOSPHOKINASE	Level of the CPK enzyme in the blood (mcg/L)
DIABETES	If the patient has diabetes (boolean)

[6] Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020). (link)

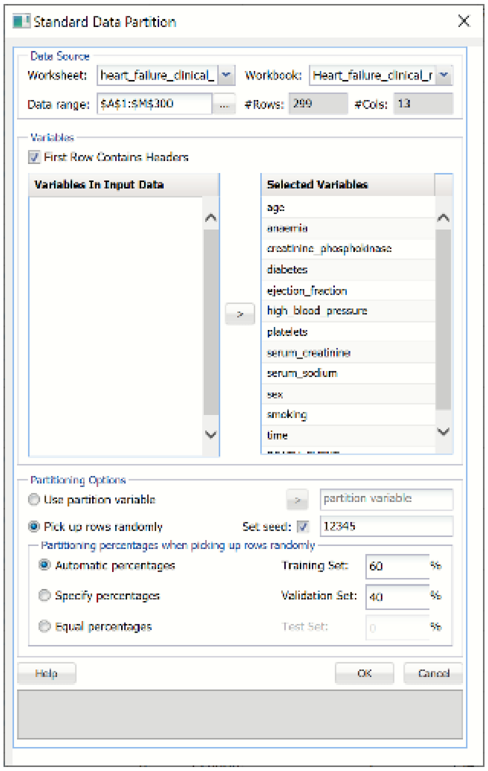
EJECTION_FRACTION	Percentage of blood leaving the heart at each contraction (percentage)
HIGH_BLOOD_PRESSURE	If the patient has hypertension (boolean)
PLATELETS	Platelets in the blood (kiloplatelets/mL)
SERUM_CREATININE	Level of serum creatinine in the blood (mg/dL)
SERUM_SODIUM	Level of serum sodium in the blood (mEq/L)
SEX	Woman (0) or man (1)
SMOKING	If the patient smokes or not (boolean)
TIME	Follow-up period (days)
DEATH_EVENT	If the patient deceased during the follow-up period (boolean)

All supervised algorithms include a new Simulation tab. This tab uses the functionality from the Generate Data feature (described in the What's New section of this guide and then more in depth in the Analytic Solver Data Science Reference Guide) to generate synthetic data based on the training partition, and uses the fitted model to produce predictions for the synthetic data. The resulting report, CFBM_Simulation, will contain the synthetic data, the predicted values and the Excel-calculated Expression column, if present. In addition, frequency charts containing the Predicted, Training, and Expression (if present) sources or a combination of any pair may be viewed, if the charts are of the same type. Since this new functionality does not support categorical variables, these types of variables will not be present in the model, only continuous variables.

Opening the Dataset

Open the Heart_failure_clinical_records_dataset.xlsx by clicking Help – Example Models – Forecasting/Data Science Examples.

Standard Data Partition Dialog



Standard Data Partition

Data Source

Worksheet: heart_failure_clinical_ Workbook: Heart_failure_clinical_r

Data range: $A$1:$M$300 #Rows: 299 #Cols: 13

Variables

☑ First Row Contains Headers

Variables In Input Data

Selected Variables: age, anaemia, creatinine_phosphokinase, diabetes, ejection_fraction, high_blood_pressure, platelets, serum_creatinine, serum_sodium, sex, smoking, time

Partitioning Options

○ Use partition variable

● Pick up rows randomly Set seed: ☑ 12345

Partitioning percentages when picking up rows randomly

● Automatic percentages Training Set: 60 %

○ Specify percentages Validation Set: 40 %

○ Equal percentages Test Set: 0 %

Partitioning the Dataset

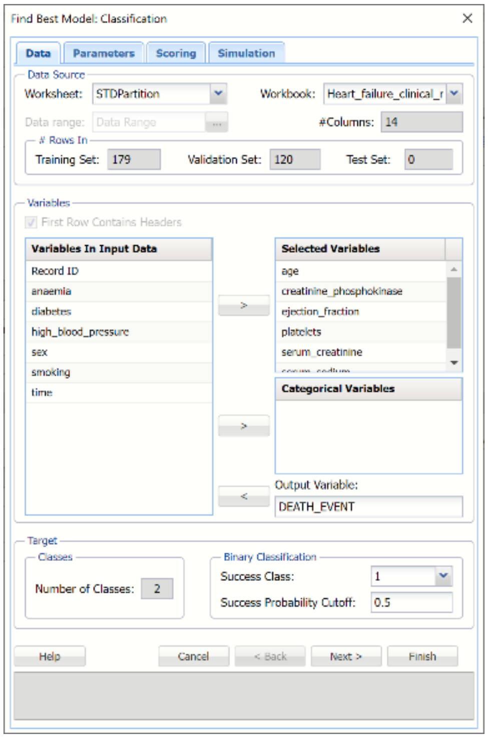
Partition the dataset by clicking **Partition – Standard Partition**.

- **Move all features under Variables In Input Data to Selected Variables.**
- Click **OK** to accept the partitioning defaults and create the partitions.

A new worksheet, STDPartition, is inserted directly to the right of the dataset. Find Best Model will be performed on both the Training and Validation partitions.

For more information on partitioning a dataset, see the Partitioning chapter within the Data Science Reference Guide.

Find Best Model: Classification Data Tab


Find Best Model: Classification

Data Source
Worksheet: STDPartition
Workbook: Heart_failure_clinical_r
Data range: Data Range
#Columns: 14

# Rows In
Training Set: 179
Validation Set: 120
Test Set: 0

Variables
☑ First Row Contains Headers

Variables In Input Data
Record ID
anaemia
diabetes
high_blood_pressure
sex
smoking
time

Selected Variables
age
creatinine_phosphokinase
ejection_fraction
platelets
serum_creatinine
serum_sodium

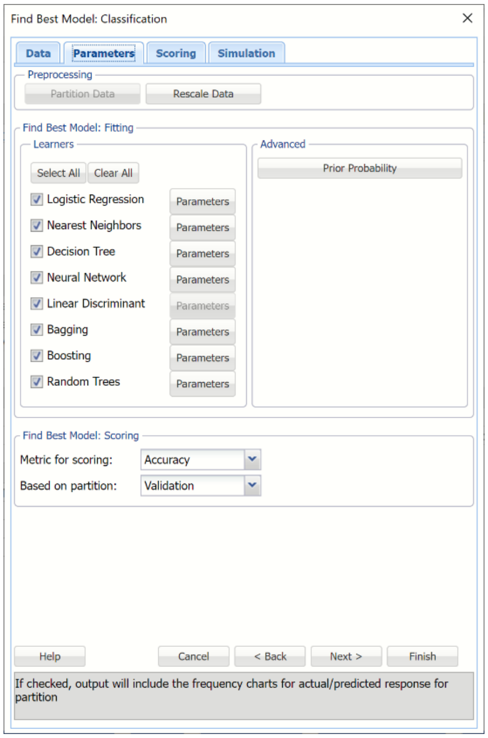
Categorical Variables

Output Variable: DEATH_EVENT

Target
Classes
Number of Classes: 2

Binary Classification
Success Class: 1
Success Probability Cutoff: 0.5

Find Best Model: Classification Parameters Tab


Find Best Model: Classification

Preprocessing

Find Best Model: Fitting
Learners
☑ Logistic Regression
☑ Nearest Neighbors
☑ Decision Tree
☑ Neural Network
☑ Linear Discriminant
☑ Bagging
☑ Boosting
☑ Random Trees

Advanced

Find Best Model: Scoring
Metric for scoring: Accuracy
Based on partition: Validation

If checked, output will include the frequency charts for actual/predicted response for partition

Running Find Best Model

With STDPartition worksheet selected, click **Classify – Find Best Model** to open the Find Best Model Data tab.

Data Tab

The continuous variables are selected on the Data tab, along with the Success Class and Success Probability Cutoff.

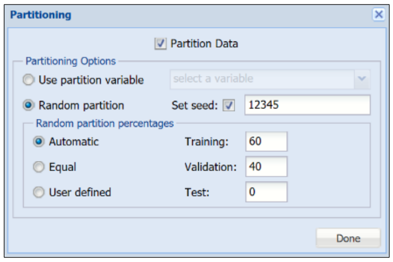
- **Select age, creatinine_phosphokinase, ejection_fraction, platelets, serum_creatinine and serum_sodium for Selected Variables.**
- **Select Death_Event as Output Variable.**
- Leave Success Class and Success Probability Cutoff at their defaults.
- Click **Next** to move to the Parameters tab.

Parameters tab

By default, all eligible supervised learners are automatically enabled based on the presence of categorical features or binary/multiclass classification. Optionally, all possible parameters for each algorithm may be defined using the Parameters button to the right of each learner.

Partition Data

The Partition Data button is disabled because the original dataset has already been partitioned. Optionally, partitioning may be controlled from the Parameters tab, if desired.


Partitioning

☑ Partition Data

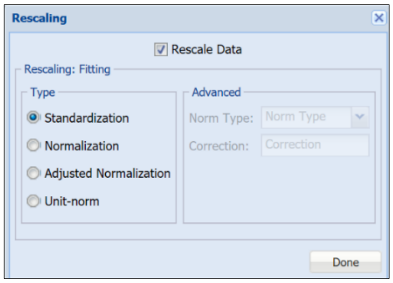
Partitioning Options
○ Use partition variable   select a variable
● Random partition   Set seed: ☑ 12345

Random partition percentages
● Automatic   Training: 60
○ Equal   Validation: 40
○ User defined   Test: 0

Rescale Data

To rescale the data, click Rescale Data and select the Rescale Data option at the top of the Rescaling dialog.

Use Rescaling to normalize one or more features in your data during the data preprocessing stage. Analytic Solver Data Science provides the following methods for feature scaling: Standardization, Normalization, Adjusted Normalization and Unit-norm. For more information on this feature, see the Rescale Continuous Data section within the Transform Continuous Data chapter that occurs in the Data Science Reference Guide.



Rescaling

☑ Rescale Data

Rescaling: Fitting

Type

● Standardization

○ Normalization

○ Adjusted Normalization

○ Unit-norm

Advanced

Norm Type: Norm Type

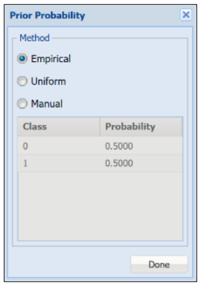
Correction: Correction

Done

- This example does not use rescaling to rescale the dataset data. Uncheck the Rescale Data option at the top of the dialog and click Done.

Prior Probability

Click **Prior Probability**. Three options appear in the *Prior Probability* Dialog: *Empirical, Uniform* and *Manual.*



Prior Probability

Method

● Empirical

○ Uniform

○ Manual

Class	Probability
0	0.5000
1	0.5000

Done

If the first option is selected, *Empirical*, Analytic Solver Data Science will assume that the probability of encountering a particular class in the dataset is the same as the frequency with which it occurs in the training data.

If the second option is selected, *Uniform*, Analytic Solver Data Science will assume that all classes occur with equal probability.

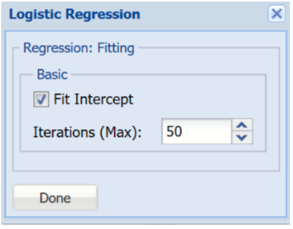
Select the third option, *Manual*, to manually enter the desired probability values for each class.

- Leave the default setting, Empirical, selected and click Done to close the dialog.

Find Best Model: Fitting

To set a parameter for each selected learner, click the Parameters button to the right. The following chart gives a brief description of the parameters for each learner.

Logistic Regression Parameter Dialog



Logistic Regression

Regression: Fitting

Basic

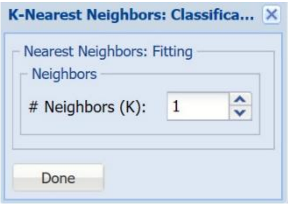
☑ Fit Intercept

Iterations (Max): 50

Done

Logistic Regression	For more information on each parameter, see the Logistic Regression chapter within the Data Science Reference Guide.
Fit Intercept	When this option is selected, the default setting, Analytic Solver Data Science will fit the Logistic Regression intercept. If this option is not selected, Analytic Solver Data Science will force the intercept term to 0.
Iterations (Max)	Estimating the coefficients in the Logistic Regression algorithm requires an iterative non-linear maximization procedure. You

K-Nearest Neighbors Parameter Dialog



K-Nearest Neighbors: Classifica... ✕

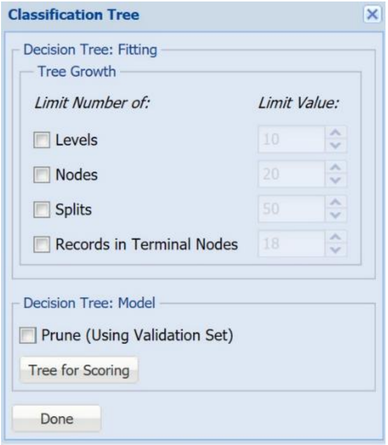
Nearest Neighbors: Fitting

Neighbors

Neighbors (K): 1

Done

Classification Tree Parameter Dialog



Classification Tree ✕

Decision Tree: Fitting

Tree Growth

Limit Number of:	Limit Value:
□ Levels	10
□ Nodes	20
□ Splits	50
□ Records in Terminal Nodes	18

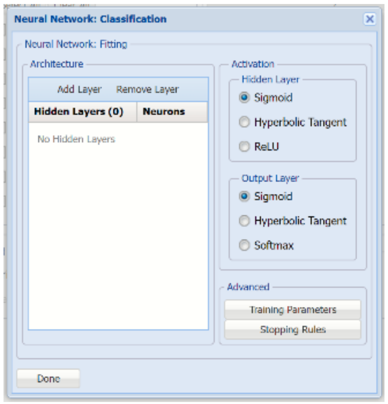
Decision Tree: Model

□ Prune (Using Validation Set)

Tree for Scoring

Done

Neural Network Parameter Dialog



Neural Network: Classification ✕

Neural Network: Fitting

Architecture

Add Layer Remove Layer

Hidden Layers (0)	Neurons
No Hidden Layers	

Activation

Hidden Layer

● Sigmoid
○ Hyperbolic Tangent
○ ReLU

Output Layer

● Sigmoid
○ Hyperbolic Tangent
○ Softmax

Advanced

Training Parameters

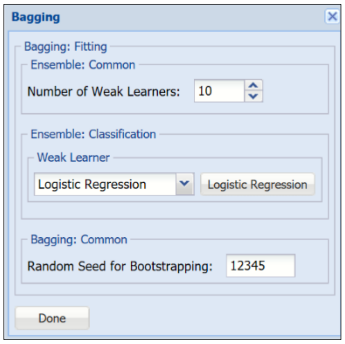
Stopping Rules

Done

	can specify a maximum number of iterations to prevent the program from getting lost in very lengthy iterative loops. This value must be an integer greater than 0 or less than or equal to 100 (1< value <= 100).
K-Nearest Neighbors	For more information on each parameter, see the K-Nearest Neighbors Classification chapter within the Data Science Reference Guide.
# Neighbors (k)	Enter a value for the parameter K in the Nearest Neighbor algorithm.
Classification Tree	For more information on each parameter, see the Classification Tree chapter within the Data Science Reference Guide.
Tree Growth Levels, Nodes, Splits, Tree Records in Terminal Nodes	In the *Tree Growth* section, select Levels, Nodes, Splits, and Records in Terminal Nodes. Values entered for these options limit tree growth, i.e. if 10 is entered for Levels, the tree will be limited to 10 levels.
Prune	If a validation partition exists, this option is enabled. When this option is selected, Analytic Solver Data Science will prune the tree using the validation set. Pruning the tree using the validation set reduces the error from over-fitting the tree to the training data. Click Tree for Scoring to click the Tree type used for scoring: Fully Grown, Best Pruned, Minimum Error, User Specified or Number of Decision Nodes.
Neural Network	For more information on each parameter, see the Neural Network Classification chapter within the Data Science Reference Guide.
Architecture	Click *Add Layer* to add a hidden layer. To delete a layer, click *Remove Layer*. Once the layer is added, enter the desired Neurons.
Hidden Layer	Nodes in the hidden layer receive input from the input layer. The output of the hidden nodes is a weighted sum of the input values. This weighted sum is computed with weights that are initially set at random values. As the network "learns", these weights are adjusted. This weighted sum is used to compute the hidden node's output using a *transfer function*. The default selection is *Sigmoid*.
Output Layer	As in the hidden layer output calculation (explained in the above paragraph), the output layer is also computed using the same transfer function as described for *Activation: Hidden Layer*. The default selection is *Sigmoid*.
Training Parameters	Click Training Parameters to open the Training Parameters dialog to specify parameters related to the training of the Neural Network algorithm.
Stopping Rules	Click Stopping Rules to open the Stopping Rules dialog. Here users can specify a comprehensive set of rules for stopping the

	algorithm early plus cross-validation on the training error.
Linear Discriminant	For more information on this learner, see the Discriminant Analysis Classification chapter within the Data Science Reference Guide.
Bagging Ensemble Method	For more information on each parameter, see the Ensemble Methods Classification chapter within the Data Science Reference Guide.
Number of Weak Learners	This option controls the number of “weak” classification models that will be created. The ensemble method will stop when the number of classification models created reaches the value set for this option. The algorithm will then compute the weighted sum of votes for each class and assign the “winning” classification to each record.
Weak Learner	Under Ensemble: Classification click the down arrow beneath Weak Leaner to select one of the six featured classifiers: Linear Discriminant, Logistic Regression, k-NN, Naïve Bayes, Neural Networks, or Decision Trees. The command button to the right will be enabled. Click this command button to control various option settings for the weak leaner.
Random Seed for Bootstrapping	Enter an integer value to specify the seed for random resampling of the training data for each weak learner. Setting the random number seed to a nonzero value (any number of your choice is OK) ensures that the same sequence of random numbers is used each time the dataset is chosen for the classifier. The default value is “12345”. If left blank, the random number generator is initialized from the system clock, so the sequence of random numbers will be different in each calculation. If you need the results from successive runs of the algorithm to another to be strictly comparable, you should set the seed. To do this, type the desired number you want into the box. This option accepts positive integers with up to 9 digits.
Boosting Ensemble Method	For more information on each parameter, see the Boosting Classification Ensemble Method chapter within the Data Science Reference Guide.
Number of Weak Learners Weak Learner	See description above.
Adaboost Variant	In AdaBoost.M1 (Freund), the constant is calculated as: $\alpha_b = \ln((1-e_b)/e_b)$ In AdaBoost.M1 (Breiman), the constant is calculated as: $\alpha_b = 1/2\ln((1-e_b)/e_b)$ In SAMME, the constant is calculated as: $\alpha_b = 1/2\ln((1-e_b)/e_b + \ln(k-1)$ where k is the number of classes
Random Seed for Resampling	Enter an integer value to specify the seed for random resampling of the training data for

Ensemble Method Bagging Parameter Dialog



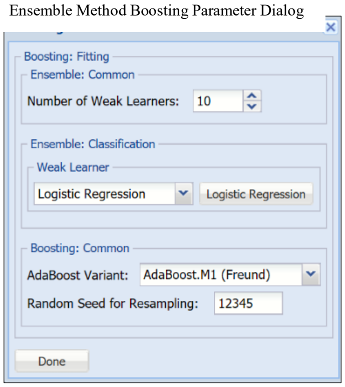
Bagging

Number of Weak Learners: 10

Weak Learner: Logistic Regression

Random Seed for Bootstrapping: 12345

Ensemble Method Boosting Parameter Dialog



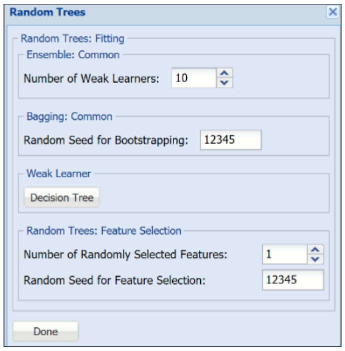
Number of Weak Learners: 10

Weak Learner: Logistic Regression

AdaBoost Variant: AdaBoost.M1 (Freund)

Random Seed for Resampling: 12345

Ensemble Method Random Trees Parameter Dialog



Random Trees

Random Trees: Fitting

Ensemble: Common

Number of Weak Learners: 10

Bagging: Common

Random Seed for Bootstrapping: 12345

Weak Learner

Decision Tree

Random Trees: Feature Selection

Number of Randomly Selected Features: 1

Random Seed for Feature Selection: 12345

Done

	each weak learner. Setting the random number seed to a nonzero value (any number of your choice is OK) ensures that the same sequence of random numbers is used each time the dataset is chosen for the classifier. The default value is "12345".
Random Trees Ensemble Method	For more information on each parameter, see the Boosting Classification Ensemble Method chapter within the Data Science Reference Guide.
Number of Weak Learners Random Seed for Bootstrapping Weak Learner	See description above.
Number of Randomly Selected Features	The Random Trees ensemble method works by training multiple "weak" classification trees using a fixed number of randomly selected features then taking the mode of each class to create a "strong" classifier. The option *Number of randomly selected features* controls the fixed number of randomly selected features in the algorithm. The default setting is **3**.
Feature Selection Random Seed	If an integer value appears for *Feature Selection Random seed*, Analytic Solver Data Science will use this value to set the feature selection random number seed. Setting the random number seed to a nonzero value (any number of your choice is OK) ensures that the same sequence of random numbers is used each time the dataset is chosen for the classifier. The default value is "12345". If left blank, the random number generator is initialized from the system clock, so the sequence of random numbers will be different in each calculation. If you need the results from successive runs of the algorithm to another to be strictly comparable, you should set the seed. To do this, type the desired number you want into the box. This option accepts positive integers with up to 9 digits.

Find Best Model: Scoring

The two parameters at the bottom of the dialog under Find Best Model: Scoring, determine how well each classification method fits the data.

The Metric for Scoring may be changed to Accuracy, Specificity, Sensitivity, Precision or F-1. See the table below for a brief description of each statistic.

- For this example, leave Accuracy selected.

Statistic	Description
Accuracy	Accuracy refers to the ability of the classifier to predict a class label correctly.
Specificity	Specificity is defined as the proportion of negative classifications that were actually negative.

Sensitivity	Sensitivity is defined as the proportion of positive cases there were classified correctly as positive.
Precision	Precision is defined as the proportion of positive results that are truely positive.
F-1	Calculated as 0.743 –2 x (Precision * Sensitivity)/(Precision + Sensitivity) The F-1 Score provides a statistic to balance between Precision and Sensitivity, especially if an uneven class distribution exists. The closer the F-1 score is to 1 (the upper bound) the better the precision and recall.

The options for Based on Partition depend on the number of partitions present. In this example, the original dataset was partitioned into training and validation partitions so those will be the options present in the drop down menu.

- For this example **select Validation**.

Click Next to advance to the Scoring tab.

Scoring tab

Output options are selected on the Scoring tab. In this example select all four options: Detailed Report, Summary Report and Lift Charts and Frequency Chart under both Score Training Data and Score Validation Data.

By default, CFBM_Output and CFBM_Stored are generated and inserted directly to the right of the STDPartition.

- CFBM_Output contains a listing of all model inputs such a input/output variables and parameter settings for all Learners, as well as Model Performance tables containing evaluations for every available metric, every learner on all available partitions. The Learner identified internally as performing the best, is highlighted in red. (Recall that the statistic used for determining which Learner performs best on the dataset was selected on the Parameters tab.)
- FBM_Stored contains the PMML (Predictive Model Markup Language) model which can be utilized to score new data. For more information on scoring, see the Scoring chapter that appears later in this guide.

Selecting *Detailed Report* produces CFBM_TrainingScore and CFBM_ValidationScore.

- Both reports contain detailed scoring information on both the Training and Validation partitions using the "best" learner.

  Misclassified records are highlighted in red.

Summary Report is selected by default. This option produces a summary report at the top of both CFBM_TrainingScore and CFBM_ValidationScore worksheets.

- *Summary Reports* contains a Confusion Matrix, Error Report and various metrics. The chart to the left displays how each metric is calculated.

- True Positive is the number of real positives in the dataset (i.e. the number of positive records that were classified as positive by the learner).
- True Negative is the number of real negatives in the dataset (i.e. the number of negative records that were classified as negative by the learner).
- False Negative is the number of records classified as negative that are actually positive in the dataset.
- False Positive is the number of records classified as positive that are actually negative in the dataset.

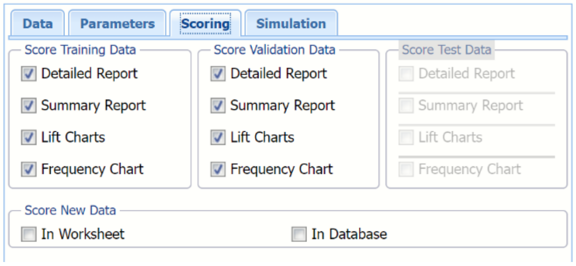
- Selecting *Lift Charts* generates Lift Charts, ROC Curves and Decile-Wise Lift Charts.
- When Frequency Chart is selected, a frequency chart will be displayed when the CFBM_TrainingScore and CFBM_ValidationScore worksheets are selected. This chart will display an interactive application similar to the Analyze Data feature, explained in detail in the Analyze Data chapter that appears earlier in this guide. This chart will include frequency distributions of the actual and predicted responses individually, or side-by-side, depending on the user's preference, as well as basic and advanced statistics for variables, percentiles, six sigma indices.

Since we did not create a test partition, the options for Score test data are disabled. See the chapter "Data Science Partitioning" for information on how to create a test partition.

See the *Scoring New Data* chapter that appears later in this guide for more information on *Score New Data in* options.

Click Finish to run Find Best Model.

Find Best Model Classification Scoring tab



Score Training Data
☑ Detailed Report
☑ Summary Report
☑ Lift Charts
☑ Frequency Chart

Score Validation Data
☑ Detailed Report
☑ Summary Report
☑ Lift Charts
☑ Frequency Chart

Score Test Data
□ Detailed Report
□ Summary Report
□ Lift Charts
□ Frequency Chart

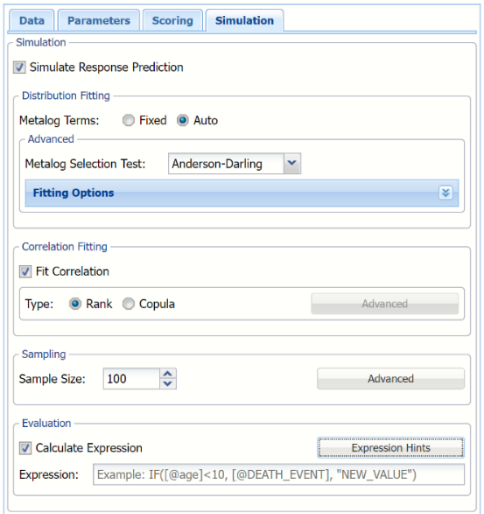
Score New Data
□ In Worksheet
□ In Database

Click **Next** to advance to the Simulation tab.

Select Simulation Response Prediction to enable all options on the Simulation tab of the Find Best Model Classification dialog.

Simulation tab: All supervised algorithms include a new Simulation tab. This tab uses the functionality from the Generate Data feature (described earlier in this guide) to generate synthetic data based on the training partition, and uses the fitted model to produce predictions for the synthetic data. The resulting report, CFMB_Simulation, will contain the synthetic data, the predicted values and the Excel-calculated Expression column, if present. In addition, frequency charts containing the Predicted, Training, and Expression (if present) sources or a combination of any pair may be viewed, if the charts are of the same type.

Find Best Model Classification dialog, Simulation tab



☑ Simulate Response Prediction

Metalog Terms: ○ Fixed ● Auto

Metalog Selection Test: Anderson-Darling

☑ Fit Correlation

Type: ● Rank ○ Copula

Sample Size: 100

☑ Calculate Expression

Expression: Example: IF([@age]<10, [@DEATH_EVENT], "NEW_VALUE")

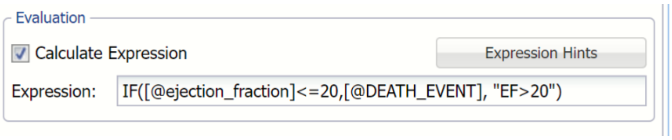
Evaluation: Select Calculate Expression to amend an Expression column onto the frequency chart displayed on the CFBM_Simulation output tab. Expression can be any valid Excel formula that references a variable and the response as [@COLUMN_NAME]. Click the Expression Hints button for more information on entering an expression.

For the purposes of this example, leave all options at their defaults in the Distribution Fitting, Correlation Fitting and Sampling sections of the dialog. For Expression, enter the following formula to display if the patient suffered catastrophic heart failure (@DEATH_EVENT) when his/her Ejection_Fraction was less than or equal to 20.

IF([@ejection_fraction]<=20, [@DEATH_EVENT], “EF>20”)

Note that variable names are case sensitive.

Evaluation section on the Find Best Model dialog, Simulation tab



☑ Calculate Expression

Expression: IF([@ejection_fraction]<=20,[@DEATH_EVENT], "EF>20")

For more information on the remaining options shown on this dialog in the Distribution Fitting, Correlation Fitting and Sampling sections, see the Generate Data chapter that appears earlier in this guide.

Click **Finish** to run Find Best Model on the example dataset.

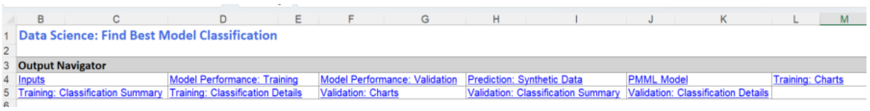
Interpreting the Results

All worksheets containing results from Find Best Model are inserted to the right of the STDPartition worksheet.

CFBM_Output Result Worksheet

The CFBM_Output worksheet is inserted directly to the right of the STDPartition worksheet. This report lists all input variables and all parameter settings for each learner, along with the Model Performance of each Learner on all partitions. This example utilizes two partitions, training and validation.

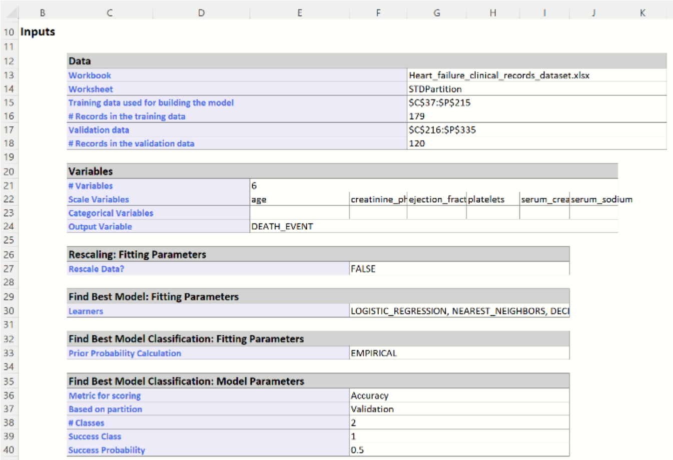
The Output Navigator appears at the very top of this worksheet. Click the links to easily move between each section of the output. The Output Navigator is listed at the top of each worksheet included in the output.



Data Science: Find Best Model Classification

Output Navigator					
Inputs	Model Performance: Training	Model Performance: Validation	Prediction: Synthetic Data	PMML Model	Training: Charts
Training: Classification Summary	Training: Classification Details	Validation: Charts	Validation: Classification Summary	Validation: Classification Details	

The Inputs section includes information pertaining to the dataset, the input variables and parameter settings.



Inputs

Data	
Workbook	Heart_failure_clinical_records_dataset.xlsx
Worksheet	STDPartition
Training data used for building the model	$C$37:$P$215
# Records in the training data	179
Validation data	$C$216:$P$335
# Records in the validation data	120

Variables						
# Variables	6					
Scale Variables	age	creatinine_ph	ejection_fract	platelets	serum_crea	serum_sodium
Categorical Variables						
Output Variable	DEATH_EVENT					

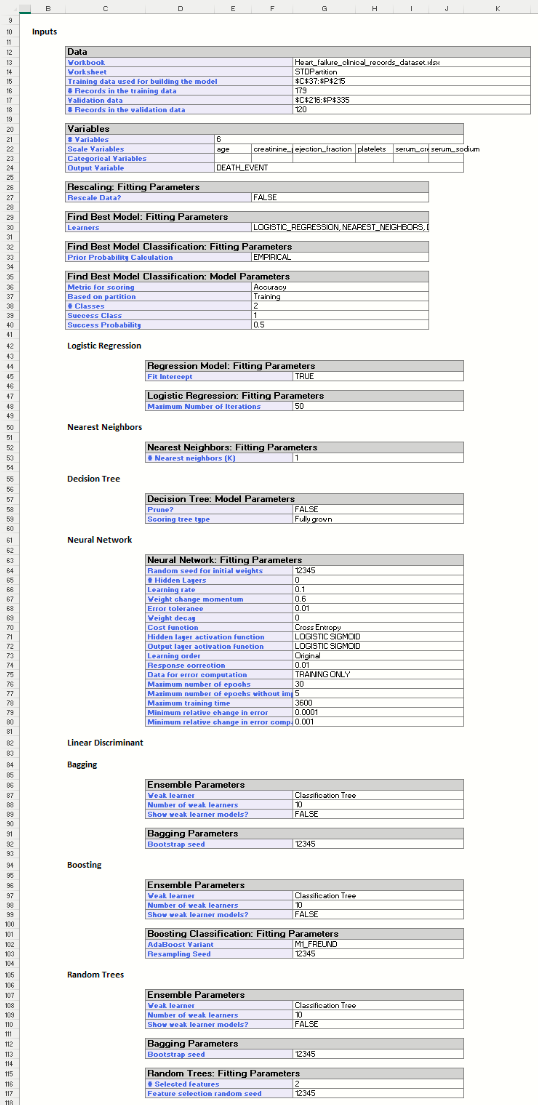
Rescaling: Fitting Parameters	
Rescale Data?	FALSE

Find Best Model: Fitting Parameters	
Learners	LOGISTIC_REGRESSION, NEAREST_NEIGHBORS, DECI

Find Best Model Classification: Fitting Parameters	
Prior Probability Calculation	EMPIRICAL

Find Best Model Classification: Model Parameters	
Metric for scoring	Accuracy
Based on partition	Validation
# Classes	2
Success Class	1
Success Probability	0.5

Further down within Inputs, the parameter selections for each Learner are listed.



Inputs

Data	
Workbook	Heart_failure_clinical_records_dataset.xlsx
Worksheet	STDPartition
Training data used for building the model	$C$37:$P$215
# Records in the training data	179
Validation data	$C$216:$P$335
# Records in the validation data	120

Variables						
# Variables	6					
Scale Variables	age	creatinine_	ejection_fraction	platelets	serum_cre	serum_sodium
Categorical Variables						
Output Variable	DEATH_EVENT					

Rescaling: Fitting Parameters	
Rescale Data?	FALSE

Find Best Model: Fitting Parameters	
Learners	LOGISTIC_REGRESSION, NEAREST_NEIGHBORS, [

Find Best Model Classification: Fitting Parameters	
Prior Probability Calculation	EMPIRICAL

Find Best Model Classification: Model Parameters	
Metric for scoring	Accuracy
Based on partition	Training
# Classes	2
Success Class	1
Success Probability	0.5

Logistic Regression

Regression Model: Fitting Parameters	
Fit Intercept	TRUE

Logistic Regression: Fitting Parameters	
Maximum Number of Iterations	50

Nearest Neighbors

Nearest Neighbors: Fitting Parameters	
# Nearest neighbors (K)	1

Decision Tree

Decision Tree: Model Parameters	
Prune?	FALSE
Scoring tree type	Fully grown

Neural Network

Neural Network: Fitting Parameters	
Random seed for initial weights	12345
# Hidden Layers	0
Learning rate	0.1
Weight change momentum	0.6
Error tolerance	0.01
Weight decay	0
Cost function	Cross Entropy
Hidden layer activation function	LOGISTIC SIGMOID
Output layer activation function	LOGISTIC SIGMOID
Learning order	Original
Response correction	0.01
Data for error computation	TRAINING ONLY
Maximum number of epochs	30
Maximum number of epochs without imp	5
Maximum training time	3600
Minimum relative change in error	0.0001
Minimum relative change in error comp	0.001

Linear Discriminant

Bagging

Ensemble Parameters	
Weak learner	Classification Tree
Number of weak learners	10
Show weak learner models?	FALSE

Bagging Parameters	
Bootstrap seed	12345

Boosting

Ensemble Parameters	
Weak learner	Classification Tree
Number of weak learners	10
Show weak learner models?	FALSE

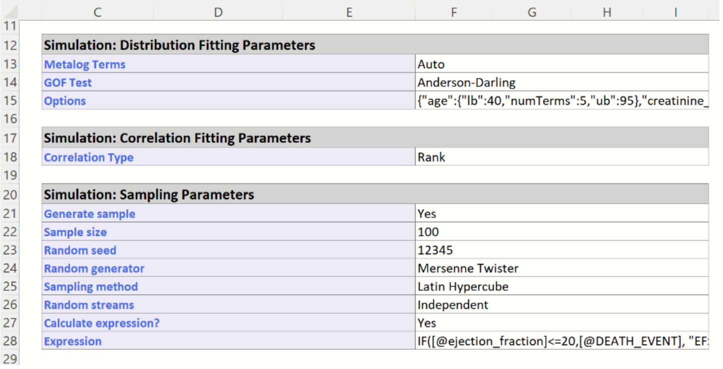
Boosting Classification: Fitting Parameters	
AdaBoost Variant	M1_FREUND
Resampling Seed	12345

Random Trees

Ensemble Parameters	
Weak learner	Classification Tree
Number of weak learners	10
Show weak learner models?	FALSE

Bagging Parameters	
Bootstrap seed	12345

Random Trees: Fitting Parameters	
# Selected features	2
Feature selection random seed	12345

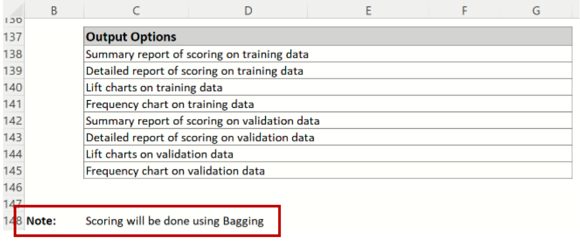


Simulation: Distribution Fitting Parameters	
Metalog Terms	Auto
GOF Test	Anderson-Darling
Options	{"age":{"lb":40,"numTerms":5,"ub":95},"creatinine_

Simulation: Correlation Fitting Parameters	
Correlation Type	Rank

Simulation: Sampling Parameters	
Generate sample	Yes
Sample size	100
Random seed	12345
Random generator	Mersenne Twister
Sampling method	Latin Hypercube
Random streams	Independent
Calculate expression?	Yes
Expression	IF([@ejection_fraction]<=20,[@DEATH_EVENT], "EF:

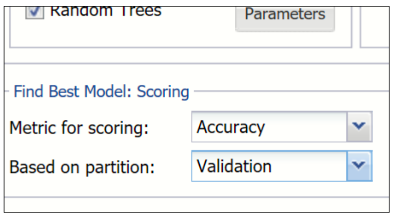
Scroll down to view any generated messages from the Find Best Model feature and also to view the performance of each learner on the training and validation partitions.



Output Options
Summary report of scoring on training data
Detailed report of scoring on training data
Lift charts on training data
Frequency chart on training data
Summary report of scoring on validation data
Detailed report of scoring on validation data
Lift charts on validation data
Frequency chart on validation data

Note: Scoring will be done using Bagging

Metrics/Partition for Selecting Best Model



☑ Random Trees

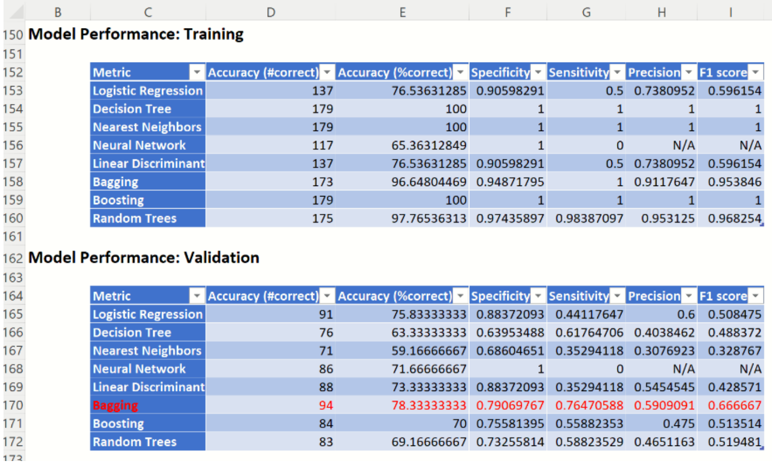
Find Best Model: Scoring

Metric for scoring: Accuracy

Based on partition: Validation

The Messages portion of the report indicates that Scoring will be performed using the Bagging ensemble method, the Learner selected as the "best" choice according to the selection for Find Best Model: Scoring parameters on the Parameters tab: Validation Partition Accuracy Metric.

Since the Bagging Accuracy metric for the Validation Partition has the highest score, that is the Learner that will be used for scoring.



Model Performance: Training

Metric	Accuracy (#correct)	Accuracy (%correct)	Specificity	Sensitivity	Precision	F1 score
Logistic Regression	137	76.53631285	0.90598291	0.5	0.7380952	0.596154
Decision Tree	179	100	1	1	1	1
Nearest Neighbors	179	100	1	1	1	1
Neural Network	117	65.36312849	1	0	N/A	N/A
Linear Discriminant	137	76.53631285	0.90598291	0.5	0.7380952	0.596154
Bagging	173	96.64804469	0.94871795	1	0.9117647	0.953846
Boosting	179	100	1	1	1	1
Random Trees	175	97.76536313	0.97435897	0.98387097	0.953125	0.968254

Model Performance: Validation

Metric	Accuracy (#correct)	Accuracy (%correct)	Specificity	Sensitivity	Precision	F1 score
Logistic Regression	91	75.83333333	0.88372093	0.44117647	0.6	0.508475
Decision Tree	76	63.33333333	0.63953488	0.61764706	0.4038462	0.488372
Nearest Neighbors	71	59.16666667	0.68604651	0.35294118	0.3076923	0.328767
Neural Network	86	71.66666667	1	0	N/A	N/A
Linear Discriminant	88	73.33333333	0.88372093	0.35294118	0.5454545	0.428571
Bagging	94	78.33333333	0.79069767	0.76470588	0.5909091	0.666667
Boosting	84	70	0.75581395	0.55882353	0.475	0.513514
Random Trees	83	69.16666667	0.73255814	0.58823529	0.4651163	0.519481

CFBM_TrainingScore

CFBM_TrainingScore contains the Classification Summary and the Classification Details reports. Both reports have been generated using the Bagging Learner, as discussed above.

CFBM_TrainingScore

Click the *CFBM_TrainingScore* tab to view the newly added Output Variable frequency chart, the Training: Classification Summary and the Training:

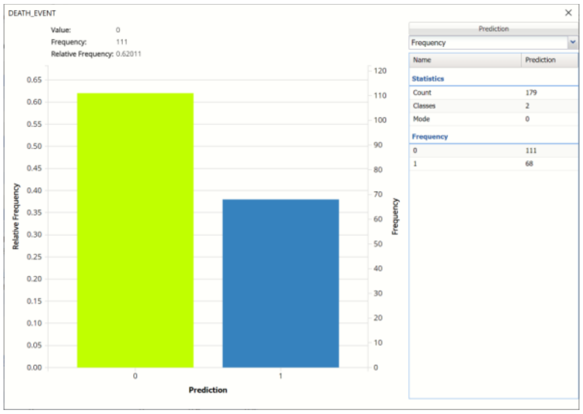
Classification Details report. All calculations, charts and predictions on this worksheet apply to the Training data.

Note: To view charts in the Cloud app, click the Charts icon on the Ribbon, select a worksheet under Worksheet and a chart under Chart.

- **Frequency Charts:** The output variable frequency chart opens automatically once the *CFBM_TrainingScore* worksheet is selected. To close this chart, click the "x" in the upper right hand corner of the chart. To reopen, click onto another tab and then click back to the *CFBM_TrainingScore* tab. To move the chart, grab the dialog's title bar and drag the chart to the desired location on the screen.

  Frequency: This chart shows the frequency for both the predicted and actual values of the output variable, along with various statistics such as count, number of classes and the mode.

  Frequency Chart on CFBM_TrainingScore output sheet



DEATH_EVENT

Value: 0
Frequency: 111
Relative Frequency: 0.62011

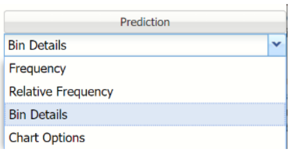
Prediction

Frequency

Name	Prediction
Statistics	
Count	179
Classes	2
Mode	0
Frequency	
0	111
1	68

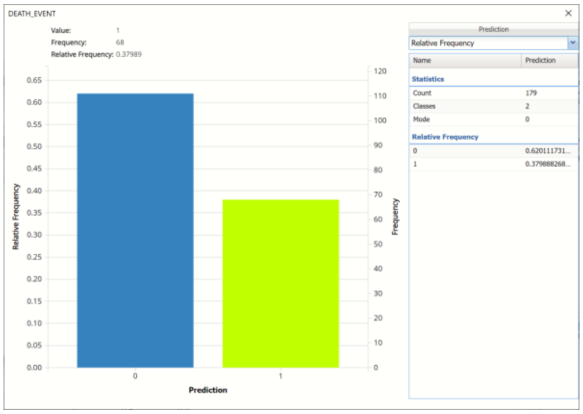
  Click the down arrow next to Frequency to switch to Relative Frequency, Bin Details or Chart Options view.

  Frequency Chart, Frequency View



  Relative Frequency: Displays the relative frequency chart.

  Relative Frequency Chart



DEATH_EVENT

Value: 1
Frequency: 68
Relative Frequency: 0.37989

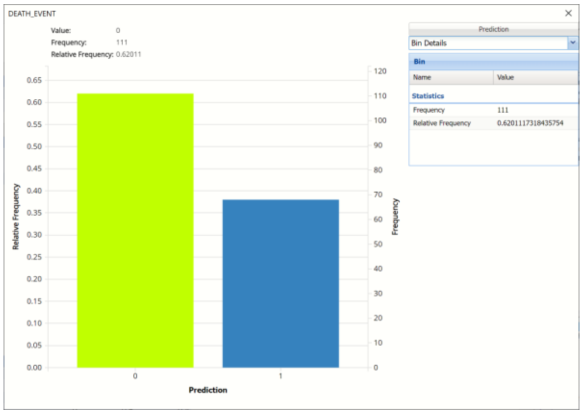
Prediction

Relative Frequency

Name	Prediction
Statistics	
Count	179
Classes	2
Mode	0
Relative Frequency	
0	0.620111731...
1	0.379888268...

Bin Details: Use this view to find metrics related to each bin in the chart.

Bin Details Chart



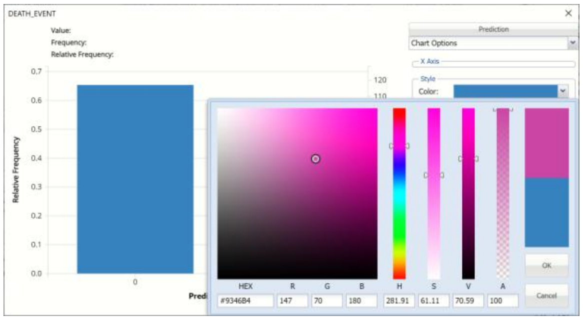
Value: 0
Frequency: 111
Relative Frequency: 0.62011

Bin

Name	Value
Statistics	
Frequency	111
Relative Frequency	0.6201117318435754

Chart Options: Use this view to change the color of the bars in the chart.

Chart Options View

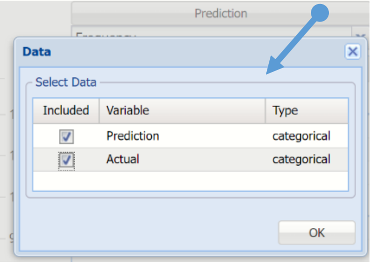


Value:
Frequency:
Relative Frequency:

HEX	R	G	B	H	S	V	A
#9346B4	147	70	180	281.91	61.11	70.59	100

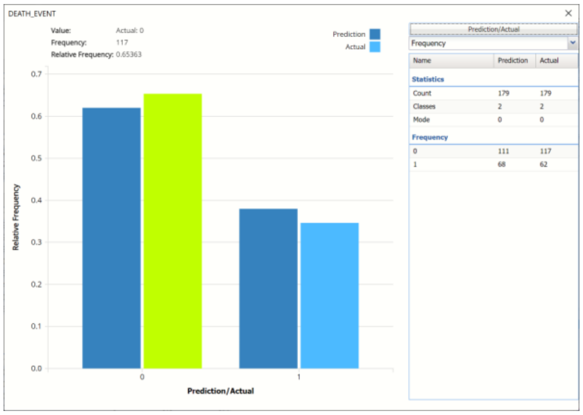
To see both predicted and actual values for the output variable in the training partition, click Prediction and select Actual. This change will be reflected on all charts.

Click Prediction to change view



Included	Variable	Type
☑	Prediction	categorical
☑	Actual	categorical

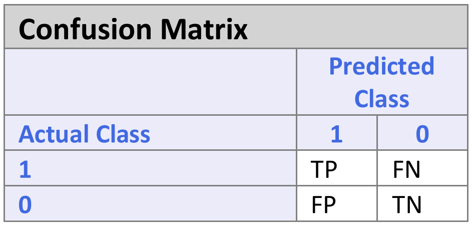
Frequency Chart displaying both Prediction and Actual data



Value: Actual: 0
Frequency: 117
Relative Frequency: 0.65363

Name	Prediction	Actual
Statistics		
Count	179	179
Classes	2	2
Mode	0	0
Frequency		
0	111	117
1	68	62

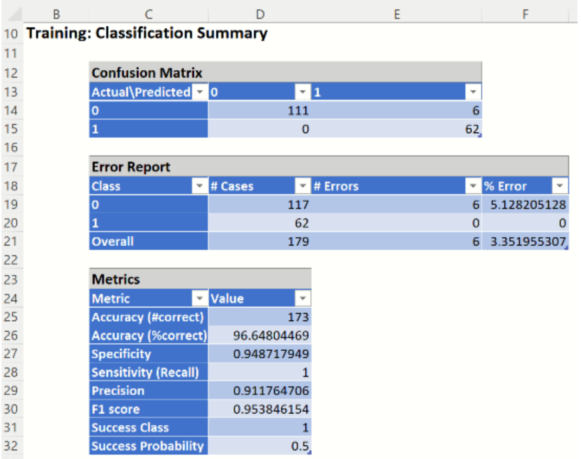
- **Classification Summary:** In the Classification Summary report, a Confusion Matrix is used to evaluate the performance of the classification method.



Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	TP	FN
0	FP	TN

  - TP stands for True Positive. These are the number of cases classified as belonging to the Success class that actually were members of the Success class.
  - FN stands for False Negative. These are the number of cases that were classified as belonging to the Failure class when they were actually members of the Success class
  - FP stands for False Positive. These cases were assigned to the Success class but were actually members of the Failure group
  - TN stands for True Negative. These cases were correctly assigned to the Failure group.

Confusion Matrix for Training Partition



Training: Classification Summary

Confusion Matrix

Actual\Predicted	0	1
0	111	6
1	0	62

Error Report

Class	# Cases	# Errors	% Error
0	117	6	5.128205128
1	62	0	0
Overall	179	6	3.351955307

Metrics

Metric	Value
Accuracy (#correct)	173
Accuracy (%correct)	96.64804469
Specificity	0.948717949
Sensitivity (Recall)	1
Precision	0.911764706
F1 score	0.953846154
Success Class	1
Success Probability	0.5

The Confusion Matrix for the Training Partition indicates:

- True Negative = 111 – These records are truly negative records. All 111 patients are survivors
- True Positive = 62 – These records are truly positive. All 62 patients are deceased.
- False Positive = 6 – These negative records were misclassified as positive. In other words, 6 surviving patients were misclassified as non-survivors.
- False Negative = 0 – No positive records were misclassified as negative, or there were no deceased patients that were misclassified as survivors.

The %Error for the training partition was 3.35%. All of the error is due to False Positives. The Bagging Ensemble misclassified 3.35% (6/179) of surviving patients as non-survivors.

The Metrics indicate:

- Accuracy: The number of records classified correctly by the Bagging Ensemble is 173 (111 + 62).
- %Accuracy: The percentage of records classified correctly is 96.1% (173/179).
- Specificity: (True Negative)/(True Negative + False Positives)

  111/(111 + 6) = 0.949

  Specificity is defined as the proportion of negative classifications that were actually negative, or the fraction of survivors that actually survived. In this model, 111 actual surviving patients were classified correctly as survivors. There were 6 false positives or 6 actual surviving patients that were classified as deceased.
- Sensitivity or Recall: (True Positive)/(True Positive + False Negative)

  62/(62 + 0) = 1.0

  Sensitivity is defined as the proportion of positive cases there were classified correctly as positive, or the proportion of actually deceased patients there were classified as deceased. In this model, 62 actual deceased patients were correctly classified as deceased. There were no false negatives or no actual deceased patients were incorrectly classified as a survivor.

  Note: Since the object of this model is to correctly classify which patients will succumb to heart failure, this is an important statistic as it is very important for a physician to be able to accurately predict which patients require mitigation.
- Precision: (True Positives)/(True Positives + False Positives)

  62/(62 + 6) = 0.912

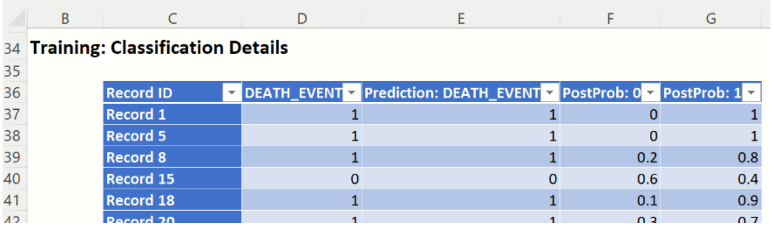
  Precision is defined as the proportion of positive results that are true positive. In this model, 62 actual deceased patients were classified correctly as deceased. There were 6 false positives or 6 actual survivors classified incorrectly as deceased.
- F-1 Score: 2 x (Precision * Sensitivity)/(Precision + Sensitivity)

  2 x (0.912 * 0.949) / (0.12 + 0.949) = 0.954

The F-1 Score provides a statistic to balance between Precision and Sensitivity, especially if an uneven class distribution exists, as in this example, (117 survivors vs 62 deceased). The closer the F-1 score is to 1 (the upper bound) the better the precision and recall.

- Success Class and Success Probability simply reports the settings for these two values as input on the Find Best Model: Classification Data tab.

- **Classification Details**: Individual records and their classifications are shown beneath Training: Classification Details. Any misclassified records are highlighted in red.



Training: Classification Details

Record ID	DEATH_EVENT	Prediction: DEATH_EVENT	PostProb: 0	PostProb: 1
Record 1	1	1	0	1
Record 5	1	1	0	1
Record 8	1	1	0.2	0.8
Record 15	0	0	0.6	0.4
Record 18	1	1	0.1	0.9
Record 20	1	1	0.3	0.7

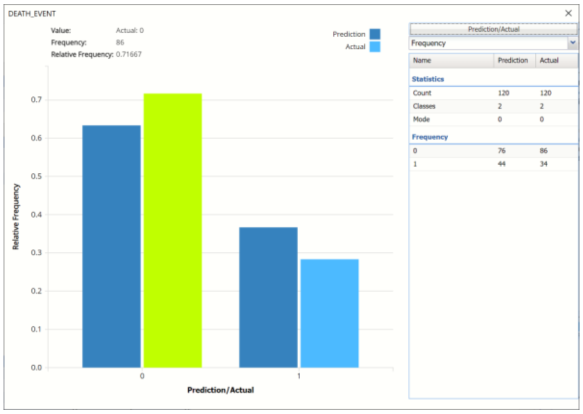
CFBM_ValidationScore

Click the *CFBM_ValidationScore* tab to view the newly added Output Variable frequency chart, the Validation: Classification Summary and the Validation: Classification Details report. All calculations, charts and predictions on this worksheet apply to the Validation data.

- **Frequency Charts:** The output variable frequency chart opens automatically once the CFBM_ValidationScore worksheet is selected. To close this chart, click the "x" in the upper right hand corner. To reopen, click onto another tab and then click back to the CFBM_ValidationScore tab.

  As explained above, this chart displays the frequency for both the predicted and actual values of the output variable, along with various statistics such as count, number of classes and the mode. Select Relative Frequency from the drop down menu, on the right, to see the relative frequencies of the output variable for both actual and predicted. See above for more information on this chart.

  CFBM_ValidationScore Frequency Chart



Name	Prediction	Actual
Statistics		
Count	120	120
Classes	2	2
Mode	0	0
Frequency		
0	76	86
1	44	34

The Confusion Matrix for the Validation Partition indicates:

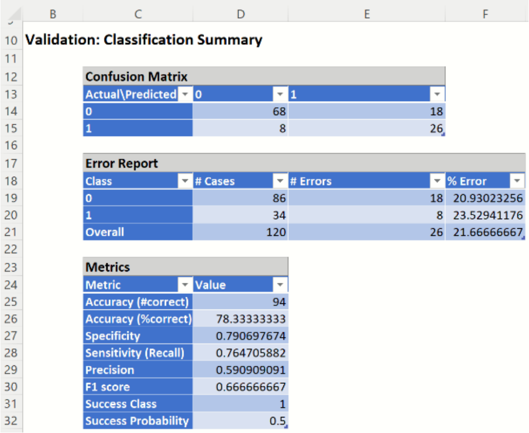
- True Negative = 68 – These records are truly negative records. All 68 patients are survivors

- True Positive = 26 – These records are truly positive. All 26 patients are deceased.

- False Positive = 18 – These negative records were misclassified as positive. In other words, 18 surviving patients were misclassified as non-survivors.

- False Negative = 8 – These positive records were misclassified as negative, or these 8 deceased patients were misclassified as survivors.

Validation Partition Classification Summary



Validation: Classification Summary

Confusion Matrix

Actual\Predicted	0	1
0	68	18
1	8	26

Error Report

Class	# Cases	# Errors	% Error
0	86	18	20.93023256
1	34	8	23.52941176
Overall	120	26	21.66666667

Metrics

Metric	Value
Accuracy (#correct)	94
Accuracy (%correct)	78.33333333
Specificity	0.790697674
Sensitivity (Recall)	0.764705882
Precision	0.590909091
F1 score	0.666666667
Success Class	1
Success Probability	0.5

The %Error for the validation partition was 21.67%. Most of the error is due to False Positives. The Bagging Ensemble misclassified 15% (18/120) of surviving patients as non-survivors and 6.7% of non-surviving patients as surviving (8/120).

The Metrics indicate:

- Accuracy: The number of records classified correctly by the Bagging Ensemble is 94 (66 + 27).

- %Accuracy: The percentage of records classified correctly is 78.33% (94/120).

- Specificity: (True Negative)/(True Negative + False Positives)

  68/(68 + 18) = 0.791

  Specificity is defined as the proportion of negative classifications that were actually negative, or the fraction of survivors that actually survived. In this model, 68 actual surviving patients were classified correctly as survivors. There were 18 false positives or 18 actual surviving patients were classified as deceased.

- Sensitivity or Recall: (True Positive)/(True Positive + False Negative)

  26/(26 + 8) = 0.765

  Sensitivity is defined as the proportion of positive cases there were classified correctly as positive, or the proportion of actually deceased patients there were classified as deceased. In this model, 26 actual deceased patients were correctly classified as deceased. There were 8 false negatives or 8 actual deceased patients were incorrectly classified as survivors.

- Precision: (True Positives)/(True Positives + False Positives)

  26/(26 + 18) = 0.591

  Precision is defined as the proportion of positive results that are true positive. In this model, 26 actual deceased patients were classified correctly as deceased. There were 18 false positives or 18 actual survivors classified incorrectly as deceased.

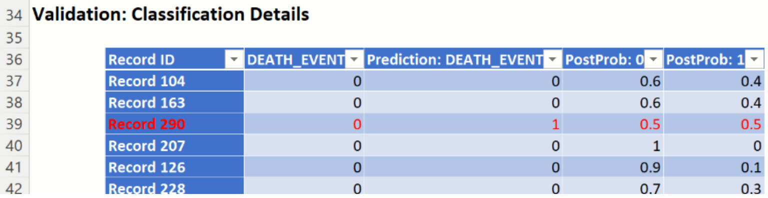
- F-1 Score: 2 x (Precision * Sensitivity)/(Precision + Sensitivity)

  2 x (0.591 * 0.765) / (0.591 + 0.765) = 0.667

  The F-1 Score provides a statistic to balance between Precision and Sensitivity, especially if an uneven class distribution exists, as in this example, (86 survivors vs 34 deceased). The closer the F-1 score is to 1 (the upper bound) the better the precision and recall.

- Success Class and Success Probability simply reports the settings for these two values as input on the Find Best Model: Classification Data tab.

Individual records and their classifications are shown beneath Validation: Classification Details. Any misclassified records are highlighted in red.



34 **Validation: Classification Details**

35

36

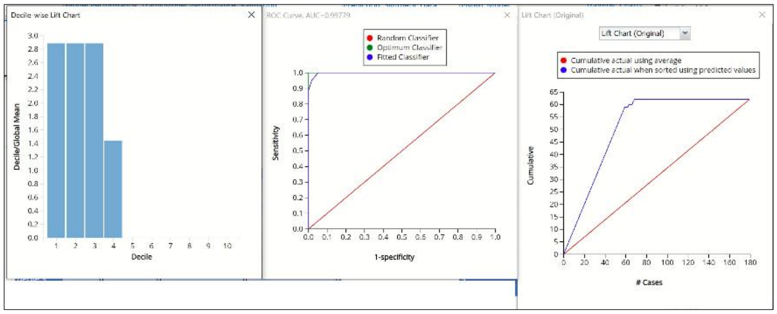
Record ID	DEATH_EVENT	Prediction: DEATH_EVENT	PostProb: 0	PostProb: 1
Record 104	0	0	0.6	0.4
Record 163	0	0	0.6	0.4
Record 290	0	1	0.5	0.5
Record 207	0	0	1	0
Record 126	0	0	0.9	0.1
Record 228	0	0	0.7	0.3

CFBM_TrainingLiftChart and CFBM_ValidationLiftChart

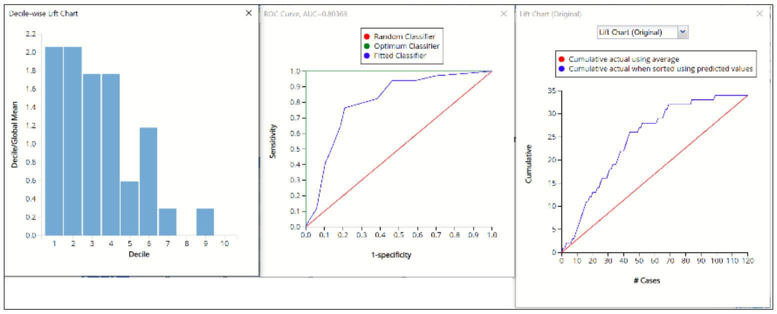
Lift Charts and ROC Curves are visual aids that help users evaluate the performance of their fitted models. Charts found on the CFBM_Training LiftChart tab were calculated using the Training Data Partition. Charts found on the CFBM_ValidationLiftChart tab were calculated using the Validation Data Partition. Since it is good practice to look at both sets of charts to assess model performance, this example will look at the lift charts generated from both partitions.

Note: To view these charts in the Cloud app, click the Charts icon on the Ribbon, select CFBM_TrainingLiftChart or CFBM_ValidationLiftChart.

Decile-wise Lift Chart, ROC Curve, and Lift Charts for Training Partition



Decile-wise Lift Chart, ROC Curve, and Lift Charts for Valid. Partition



Original Lift Chart

After the model is built using the training data set, the model is used to score on the training data set and the validation data set (if one exists). Afterwards, the data set(s) are sorted in decreasing order using the predicted output variable value. After sorting, the actual outcome values of the output variable are cumulated and the lift curve is drawn as the cumulative number of cases in decreasing probability (on the x-axis) vs the cumulative number of true positives on the y-axis. The baseline (red line connecting the origin to the end point of the blue line) is a reference line. For a given number of cases on the x-axis, this line represents the expected number of successes if no model existed, and instead cases were selected at random. This line can be used as a benchmark to measure the performance of the fitted model. The greater the area between the lift curve and the baseline, the better the model.

- In the Training Lift chart, if 100 records were selected as belonging to the success class and the fitted model was used to pick the members most likely to be successes, the lift curve indicates that 60 of them would be correct. Conversely, if 100 random records were selected and the fitted model was not used to identify the members most likely to be successes, the baseline indicates that about 30 would be correct.

- In the Validation Lift chart, if 100 records were selected as belonging to the success class and the fitted model was used to pick the members most likely to be successes, the lift curve indicates that about 40 classifications would be correct. Conversely, if 100 random records were selected and the fitted model was not used to identify the members most likely to be successes, the baseline indicates that about 25 would be correct.

Decile-wise Lift Chart

The decilewise lift curve is drawn as the decile number versus the cumulative actual output variable value divided by the decile's mean output variable value. This bars in this chart indicate the factor by which the model outperforms a random assignment, one decile at a time.

- Refer to the decile-wise lift chart for the training dataset. In the first decile, the predictive performance of the model is almost 3 times better than simply assigning a random predicted value.

- Refer to the validation graph above. In the first decile, the predictive performance of the model is a little over 2 times better than simply assigning a random predicted value.

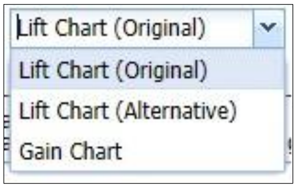
ROC Curve

The Regression ROC curve was updated in V2017. This new chart compares the performance of the regressor (Fitted Predictor) with an Optimum Predictor Curve and a Random Classifier curve. The Optimum Predictor Curve plots a hypothetical model that would provide perfect classification results. The best possible classification performance is denoted by a point at the top left of the graph at the intersection of the x and y axis. This point is sometimes referred to as the "perfect classification". The closer the AUC is to 1, the better the performance of the model.

- In the Training partition, AUC = .998 which suggests that this fitted model is a good fit to the data.
- In the Validation partition, AUC = .803 which suggests that this fitted model is also a good fit to the data.

Alternative Lift Chart and Gain Chart

In V2017, two new charts were introduced: a new Lift Chart and the Gain Chart. To display these new charts, click the down arrow next to Lift Chart (Original), in the Original Lift Chart, then select the desired chart.

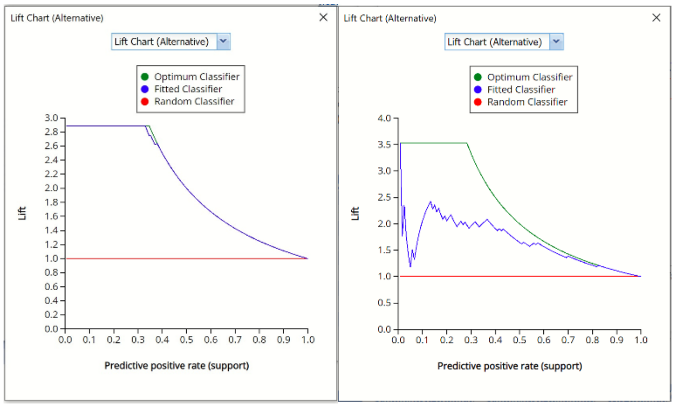


Select Lift Chart (Alternative) to display Analytic Solver Data Science's alternative Lift Chart. Each of these charts consists of an Optimum Predictor curve, a Fitted Predictor curve, and a Random Predictor curve.

The Optimum Predictor curve plots a hypothetical model that would provide perfect classification for our data. The Fitted Predictor curve plots the fitted model and the Random Predictor curve plots the results from using no model or by using a random guess (i.e. for x% of selected observations, x% of the total number of positive observations are expected to be correctly classified).

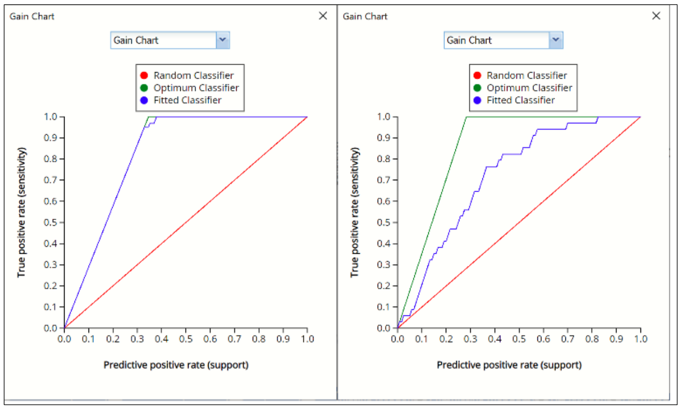
The Alternative Lift Chart plots Lift against the Predictive Positive Rate or Support.

Lift Charts (Alternative) for Training and Validation Partitions



Click the down arrow and select Gain Chart from the menu. In this chart, the True Positive Rate or Sensitivity is plotted against the Predictive Positive Rate or Support.

Gain Charts for Training and Validation Partitions

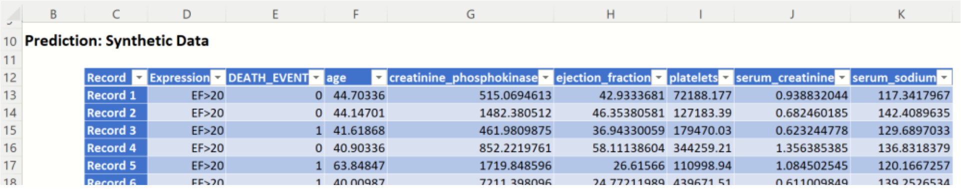


CFBM_Simulation

As discussed above, Analytic Solver Data Science generates a new output worksheet, CFBM_Simulation, when *Simulate Response Prediction* is selected on the Simulation tab of the Find Best Model dialog.

This report contains the synthetic data, the predicted values for the training partition (using the fitted model) and the Excel – calculated Expression column, if populated, in the dialog. Users can switch between charts displaying the Predicted synthetic data, the Predicted training data or the Expression, or a combination of two, as long as they are of the same type.

Synthetic Data



Prediction: Synthetic Data

Record	Expression	DEATH_EVENT	age	creatinine_phosphokinase	ejection_fraction	platelets	serum_creatinine	serum_sodium
Record 1	EF>20	0	44.70336	515.0694613	42.9333681	72188.177	0.938832044	117.3417967
Record 2	EF>20	0	44.14701	1482.380512	46.35380581	127183.39	0.682460185	142.4089635
Record 3	EF>20	1	41.61868	461.9809875	36.94330059	179470.03	0.623244778	129.6897033
Record 4	EF>20	0	40.90336	852.2219761	58.11138604	344259.21	1.356385385	136.8318379
Record 5	EF>20	1	63.84847	1719.848596	26.61566	110998.94	1.084502545	120.1667257
Record 6	EF>20	1	40.00987	7211.398096	24.77211989	439671.51	0.611009849	139.2526534

Note the first column in the output, Expression. This column was inserted into the Synthetic Data results because Calculate Expression was selected and an Excel function was entered into the Expression field, on the Simulation tab of the Find Best Model dialog.

IF([@ejection_fraction]<=20, [@DEATH_EVENT], "EF > 20")

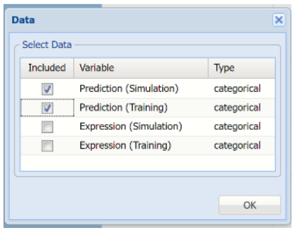
The results in this column are either 0, 1, or EF > 20.

- DEATH_EVENT = 0 indicates that the patient has an ejection fraction <= 20 but *did not* suffer catastrophic heart failure.
- DEATH_EVENT = 1 in this column indicates that the patient has an ejection fraction <= 20 and *did* suffer catastrophic heart failure.
- EF > 20 indicates that the patient's ejection fraction was over 20.

The remainder of the data in this report is synthetic data, generated using the Generate Data feature described in the chapter with the same name, that appears earlier in this guide.

Click Prediction (Simulation), in the upper right hand corner of the chart, and select Prediction (Training) in the Data dialog to add the predicted data for the training partition into the chart.

Simulation Frequency Chart Data Dialog

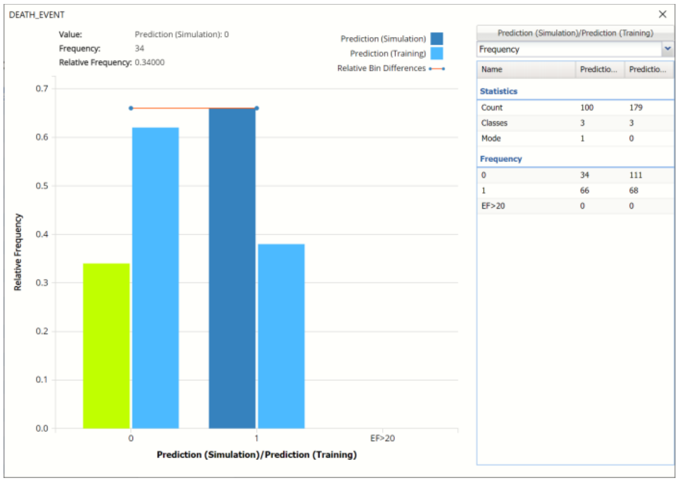


Included	Variable	Type
☑	Prediction (Simulation)	categorical
☑	Prediction (Training)	categorical
□	Expression (Simulation)	categorical
□	Expression (Training)	categorical

The chart that is displayed once this tab is selected, contains frequency information pertaining to the output variable's predicted values in the synthetic data (darker shaded columns) and the training partition (lighter shaded columns) In the synthetic data, 34 patients are predicted to survive and 66 are not. In the training data, 111 patients were predicted to survive and 68 were not.

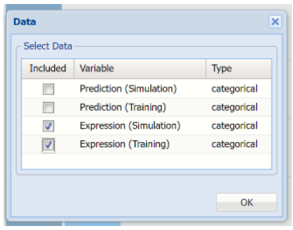
Note the flat Relative Bin Differences curve indicates the Absolute Difference in the bins is equal. (Click the down arrow next to Frequency and select Bin Details to find the bin metrics.)

Frequency Chart for CFBM_Simulation output



Name	Predictio...	Predictio...
Statistics		
Count	100	179
Classes	3	3
Mode	1	0
Frequency		
0	34	111
1	66	68
EF>20	0	0

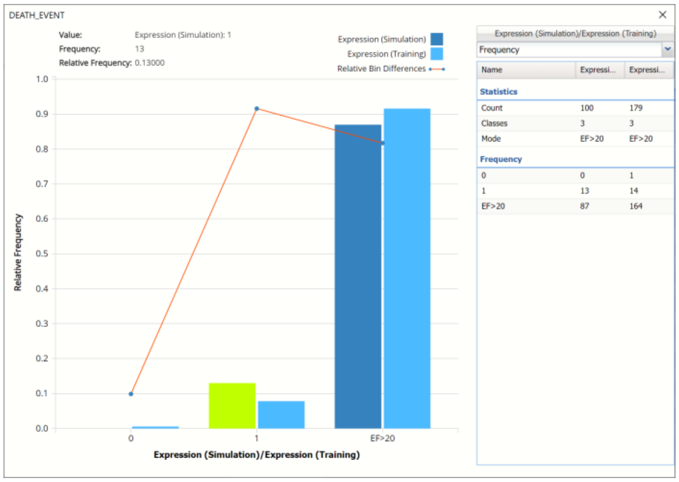
Click *Prediction(Simulation)/ Prediction (Training)* to change the chart view to *Expression (Simulation)/Expression (Training)* by selecting **Expression (Simulation)** and **Expression (Training)** in the Data dialog.



Included	Variable	Type
□	Prediction (Simulation)	categorical
□	Prediction (Training)	categorical
☑	Expression (Simulation)	categorical
☑	Expression (Training)	categorical

The chart below shows that in the synthetic, or simulated data, no "patients" with an ejection fraction <= 20 are expected to survive. However, in the training data, 1 patient with an ejection fraction <= 20 was predicted to survive. The 3rd column, EF>20 displays the remaining predictions for patients with ejection fraction > 20 in both the simulation and training data.

Frequency Chart for Expression



Value: Expression (Simulation): 1
Frequency: 13
Relative Frequency: 0.13000

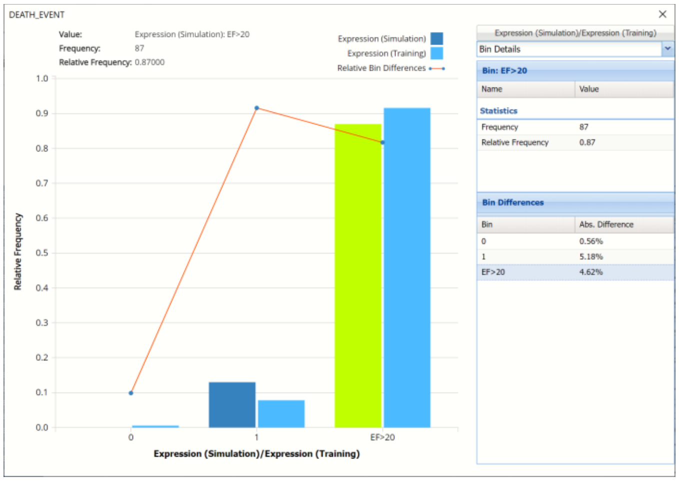
Expression (Simulation)/Expression (Training)

Frequency

Name	Expressi...	Expressi...
Statistics		
Count	100	179
Classes	3	3
Mode	EF>20	EF>20
Frequency		
0	0	1
1	13	14
EF>20	87	164

Click the down arrow next to Frequency and select Bin Details to see the Absolute Difference for each bin.

Frequency Chart for Expression, Bin Details view



Value: Expression (Simulation): EF>20
Frequency: 87
Relative Frequency: 0.87000

Expression (Simulation)/Expression (Training)

Bin Details

Bin: EF>20

Name	Value
Statistics	
Frequency	87
Relative Frequency	0.87

Bin Differences

Bin	Abs. Difference
0	0.56%
1	5.18%
EF>20	4.62%

Notice that red lines which connect the relative Bin Differences for each bin. Bin Differences are computed based on the frequencies of records which predictions fall into each bin. For example, consider the highlighted bin in the screenshot above [Expression]. There are 87 Simulation records and 164 Training records in this bin. The relative frequency of the Simulation data is 87/100 = 87% and the relative frequency of the Training data is 164/179 = 91.6%. Hence the Absolute Difference (in frequencies) is = 91.6 – 87 = 4.6%

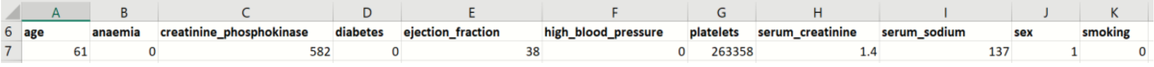
Click the down arrow next to Frequency to change the chart view to Relative Frequency or to change the look by clicking Chart Options. Statistics on the right of the chart dialog are discussed earlier in this section. For more information on the generated synthetic data, see the Generate Data chapter that appears earlier in this guide.

For information on Stored Model Sheets, in this example CFBM_*Stored*, please refer to the “Scoring New Data” chapter within the Analytic Solver Data Science User Guide.

Scoring New Data

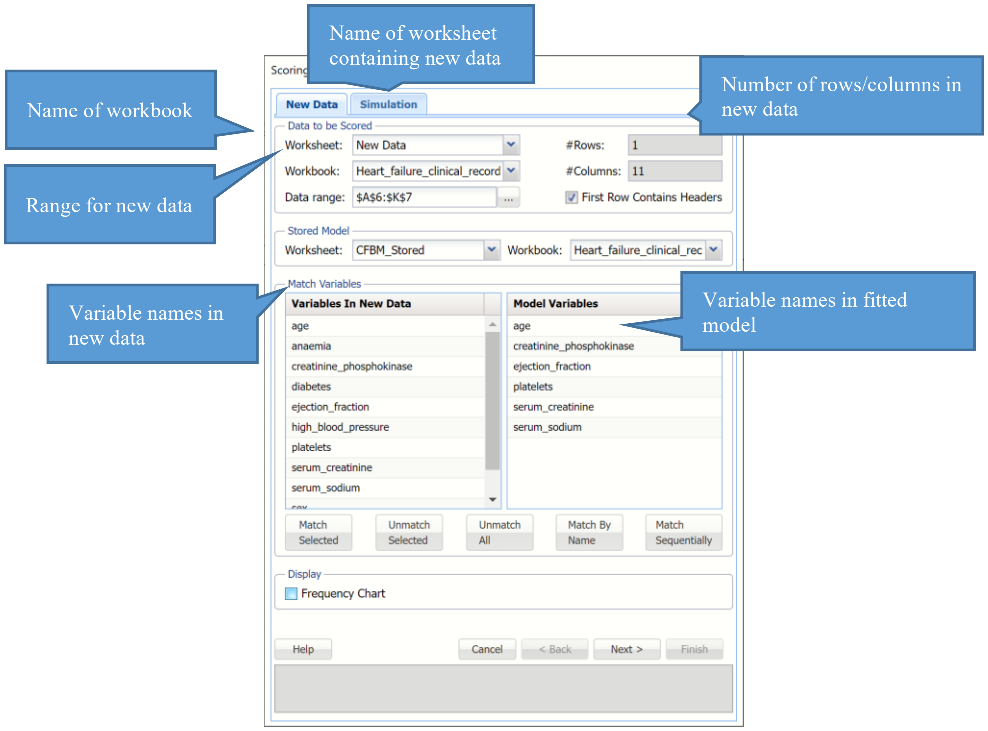
Now that the model has been fit to the data, this fitted model will be used to score new patient data, found below.

New Data



	A	B	C	D	E	F	G	H	I	J	K
6	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking
7	61	0	582	0	38	0	263358	1.4	137	1	0

To score new data, click the New Data worksheet tab and then click the Score icon on the Analytic Solver Data Science ribbon.



Name of worksheet containing new data

Name of workbook

Number of rows/columns in new data

Range for new data

Variable names in new data

Variable names in fitted model

Data to be Scored

Worksheet: New Data

Workbook: Heart_failure_clinical_record

Data range: $A$6:$K$7

#Rows: 1

#Columns: 11

☑ First Row Contains Headers

Stored Model

Worksheet: CFBM_Stored

Workbook: Heart_failure_clinical_rec

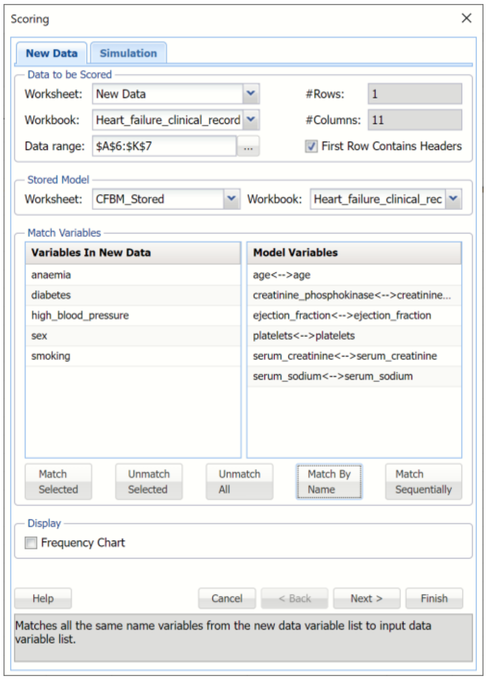
Variables In New Data	Model Variables
age	age
anaemia	creatinine_phosphokinase
creatinine_phosphokinase	ejection_fraction
diabetes	platelets
ejection_fraction	serum_creatinine
high_blood_pressure	serum_sodium
platelets	
serum_creatinine	
serum_sodium	
sex	

Display

□ Frequency Chart

Click "Match By Name" to match each variable in the new data with the same variable in the fitted model, i.e. age with age, anaemia with anaemia, etc.

Scoring tab



Scoring

New Data | Simulation

Data to be Scored

Worksheet:	New Data	#Rows:	1
Workbook:	Heart_failure_clinical_record	#Columns:	11
Data range:	$A$6:$K$7	☑ First Row Contains Headers	

Stored Model

Worksheet: CFBM_Stored Workbook: Heart_failure_clinical_rec

Match Variables

Variables In New Data	Model Variables
anaemia	age<-->age
diabetes	creatinine_phosphokinase<-->creatinine...
high_blood_pressure	ejection_fraction<-->ejection_fraction
sex	platelets<-->platelets
smoking	serum_creatinine<-->serum_creatinine
	serum_sodium<-->serum_sodium

Display

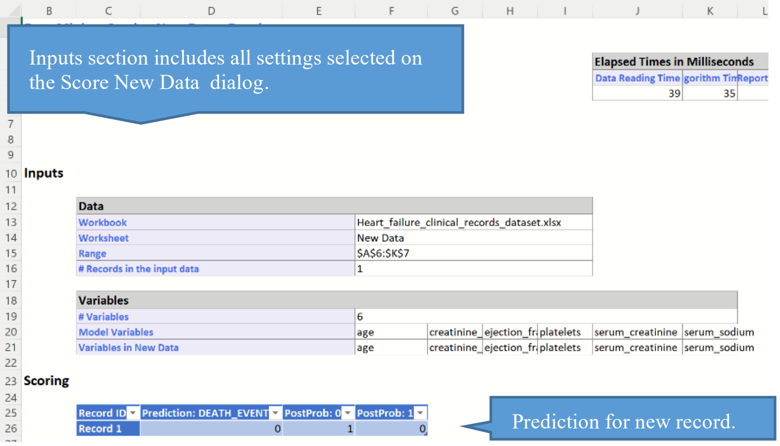
☐ Frequency Chart

Matches all the same name variables from the new data variable list to input data variable list.

Click OK to score the new data record and determine if mitigation measures are needed to keep the patient healthy.

A new worksheet, Scoring_Bagging is inserted to the right.

Scoring Results



Inputs section includes all settings selected on the Score New Data dialog.

Elapsed Times in Milliseconds		
Data Reading Time	gorithm Tin	Report
39	35	

Inputs

Data	
Workbook	Heart_failure_clinical_records_dataset.xlsx
Worksheet	New Data
Range	$A$6:$K$7
# Records in the input data	1

Variables						
# Variables	6					
Model Variables	age	creatinine_	ejection_fr	platelets	serum_creatinine	serum_sodium
Variables in New Data	age	creatinine_	ejection_fr	platelets	serum_creatinine	serum_sodium

Scoring

Record ID	Prediction: DEATH_EVENT	PostProb: 0	PostProb: 1
Record 1	0	1	0

Prediction for new record.

Notice that the predicted Death_Event is 0, or that this patient will survive without any further mitigations at this point in time. Scoring should be repeated when changes in the patient's data is observed.

Please see the “Scoring New Data” chapter within the Analytic Solver Data Science User Guide for information on scoring new data.

Find Best Model Regression Model

This example illustrates how to utilize the Find Best Model for Regression included in Analytic Solver Data Science for Desktop Excel or Excel Online by using the Wine dataset[7]. This dataset contains 13 different features describing three wine varieties obtained from three different vineyards, all located in the same vicinity.

Find Best Model fits a model to all selected regression methods in order to observe which method provides the best fit to the data. The goal of this example is to fit the best model to the dataset, then use this fitted model to determine the alcohol content in a new sample of wine.

A list of each variable contained in the dataset appears in the table below.

VARIABLE
Alcohol
Malic_Acid
Ash
Ash_Alcalinity
Magnesium
Total_Phenols
Flavanoids
Nonflavanoid_Phenols
Proanthocyanins
Color_Intensity
Hue
OD280_OD315
Proline

All supervised algorithms include a new Simulation tab. This tab uses the functionality from the Generate Data feature (described in the What's New section of this guide and then more in depth in the Analytic Solver Data Science Reference Guide) to generate synthetic data based on the training partition, and uses the fitted model to produce predictions for the synthetic data. The resulting report, PFBM_Simulation, will contain the synthetic data, the predicted values and the Excel-calculated Expression column, if present. In addition, frequency charts containing the Predicted, Training, and Expression (if present) sources or a combination of any pair may be viewed, if the charts are of the same type. Since this new functionality does not support categorical variables, these types of variables will not be present in the model, only continuous variables.

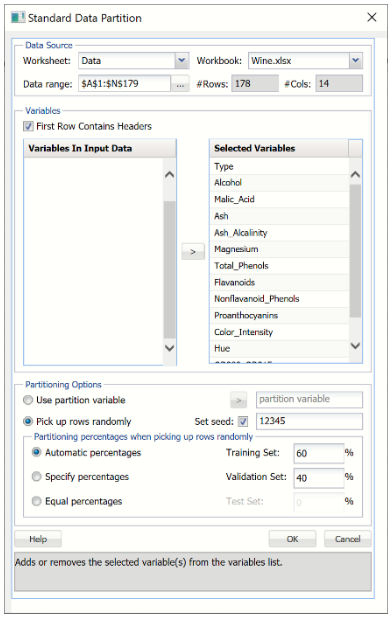
Opening the Dataset

Open wine.xlsx by clicking Help – Example Models – Forecasting/Data Science Examples.

[7] This data set can be found in the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLSummary.html or ftp://ftp.ics.uci.edu/pub/machine-learning-databases/wine/)

Partitioning the Dataset

Standard Data Partition Dialog



Standard Data Partition

Data Source

Worksheet: Data

Workbook: Wine.xlsx

Data range: $A$1:$N$179

#Rows: 178

#Cols: 14

Variables

☑ First Row Contains Headers

Variables In Input Data

Selected Variables: Type, Alcohol, Malic_Acid, Ash, Ash_Alcalinity, Magnesium, Total_Phenols, Flavanoids, Nonflavanoid_Phenols, Proanthocyanins, Color_Intensity, Hue

Partitioning Options

○ Use partition variable

● Pick up rows randomly

Set seed: ☑ 12345

Partitioning percentages when picking up rows randomly

● Automatic percentages

○ Specify percentages

○ Equal percentages

Training Set: 60 %

Validation Set: 40 %

Test Set: 0 %

Adds or removes the selected variable(s) from the variables list.

Partition the dataset by clicking **Partition – Standard Partition**.

- **Move all features under Variables In Input Data to Selected Variables.**
- Click **OK** to accept the partitioning defaults and create the partitions.

A new worksheet, STDPartition, is inserted directly to the right of the dataset. Click the new tab to open the worksheet. Find Best Model will be performed on both the Training and Validation partitions.

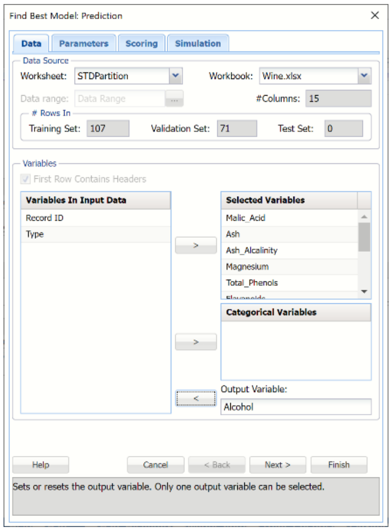
For more information on partitioning a dataset, see the Partitioning chapter within the Data Science Reference Guide.

Running Find Best Model

With STDPartition worksheet selected, click **Predict – Find Best Model** to open the Find Best Model Data tab.

Data Tab

Find Best Model: Prediction Data Tab



Find Best Model: Prediction

Data Source

Worksheet: STDPartition

Workbook: Wine.xlsx

Data range: Data Range

#Columns: 15

Rows In

Training Set: 107

Validation Set: 71

Test Set: 0

Variables

☑ First Row Contains Headers

Variables In Input Data: Record ID, Type

Selected Variables: Malic_Acid, Ash, Ash_Alcalinity, Magnesium, Total_Phenols

Categorical Variables

Output Variable: Alcohol

Sets or resets the output variable. Only one output variable can be selected.

The continuous variables are selected on the Data tab.

- **Select Malic_Acid, Ash, Ash_Alcalinity, Magnesium, Total_Phenols, Flavanoids, Nonflavanoid_Phenols, Proanthocyanins, Color_Intensity, Hue, OD280_OD315, Proline for Selected Variables.**
- **Select Alcohol for the Output Variable.**
- Click **Next** to move to the Parameters tab.

Parameters tab

By default all eligible regression learners are automatically enabled based on the presence of categorical features or binary/multiclass classification. Optionally, all possible parameters for each algorithm may be defined using the Parameters button to the right of each learner.

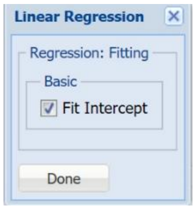
The Partition Data button is disabled because the original dataset was partitioned before Find Best Model was initiated.

Rescale Data

To rescale the data, click Rescale Data and select the Rescale Data option at the top of the Rescaling dialog.

Use Rescaling to normalize one or more features in your data during the data preprocessing stage. Analytic Solver Data Science provides the following methods for feature scaling: Standardization, Normalization, Adjusted Normalization and Unit-norm.

Linear Regression Parameter Dialog



Linear Regression ☒

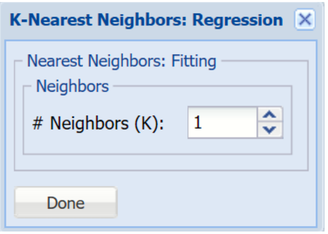
Regression: Fitting

Basic

☑ Fit Intercept

Done

K-Nearest Neighbors Regression Parameter Dialog



K-Nearest Neighbors: Regression ☒

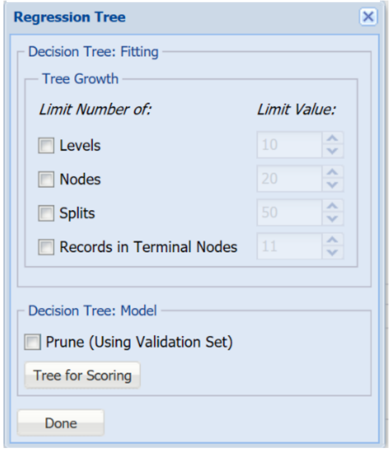
Nearest Neighbors: Fitting

Neighbors

Neighbors (K): 1

Done

Decision Trees Parameter Dialog



Regression Tree ☒

Decision Tree: Fitting

Tree Growth

Limit Number of: *Limit Value:*

☐ Levels 10

☐ Nodes 20

☐ Splits 50

☐ Records in Terminal Nodes 11

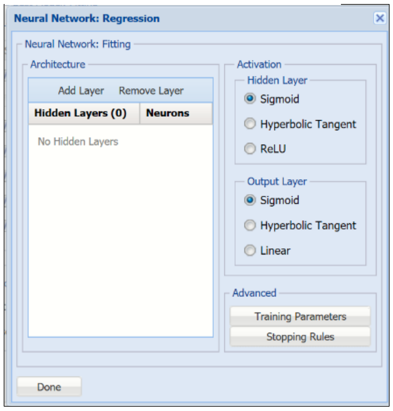
Decision Tree: Model

☐ Prune (Using Validation Set)

Tree for Scoring

Done

Neural Network Parameter Dialog



Neural Network: Regression ☒

Neural Network: Fitting

Architecture

Add Layer Remove Layer

Hidden Layers (0)	Neurons
No Hidden Layers	

Activation

Hidden Layer

● Sigmoid

○ Hyperbolic Tangent

○ ReLU

Output Layer

● Sigmoid

○ Hyperbolic Tangent

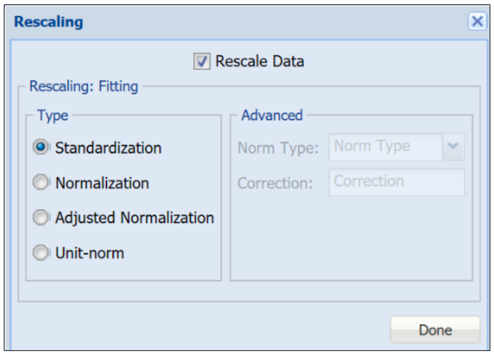
○ Linear

Advanced

Training Parameters

Stopping Rules

Done



Rescaling ☒

☑ Rescale Data

Rescaling: Fitting

Type

● Standardization

○ Normalization

○ Adjusted Normalization

○ Unit-norm

Advanced

Norm Type: Norm Type

Correction: Correction

Done

This example does not use rescaling to rescale the dataset data. Uncheck the Rescale Data option at the top of the dialog and click Done.

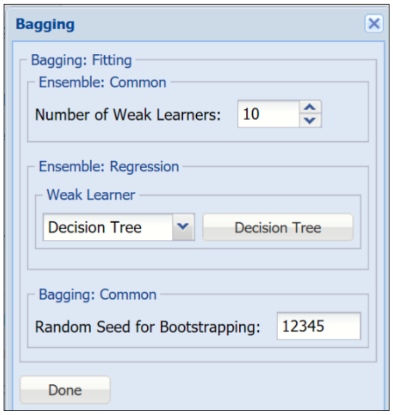
Find Best Model: Fitting

To set a parameter for each selected learner, click the Parameters button to the right.

Linear Regression	For more information on each parameter, see the Linear Regression Method chapter within the Data Science Reference Guide.
Fit Intercept	If this option is selected, a constant term will be included in the model. Otherwise, a constant term will not be included in the equation. This option is selected by default.
K-Nearest Neighbors	For more information on each parameter, see the K-Nearest Neighbors Regression Method chapter within the Data Science Reference Guide.
# Neighbors (k)	Enter a value for the parameter K in the Nearest Neighbor algorithm.
Regression Tree	For more information on each parameter, see the Regression Tree Method chapter within the Data Science Reference Guide.
Tree Growth Levels, Nodes, Splits, Tree Records in Terminal Nodes	In the *Tree Growth* section, select Levels, Nodes, Splits, and Records in Terminal Nodes. Values entered for these options limit tree growth, i.e. if 10 is entered for Levels, the tree will be limited to 10 levels.
Prune	If a validation partition exists, this option is enabled. When this option is selected, Analytic Solver Data Science will prune the tree using the validation set. Pruning the tree using the validation set reduces the error from over-fitting the tree to the training data. Click Tree for Scoring to click the Tree type used for scoring: Fully Grown, Best Pruned, Minimum Error, User Specified or Number of Decision Nodes.
Neural Network	For more information on each parameter, see the Neural Network Regression Method chapter within the Data Science Reference Guide.

Architecture	Click *Add Layer* to add a hidden layer. To delete a layer, click *Remove Layer*. Once the layer is added, enter the desired Neurons.
Hidden Layer	Nodes in the hidden layer receive input from the input layer. The output of the hidden nodes is a weighted sum of the input values. This weighted sum is computed with weights that are initially set at random values. As the network "learns", these weights are adjusted. This weighted sum is used to compute the hidden node's output using a *transfer function*. The default selection is *Sigmoid*.
Output Layer	As in the hidden layer output calculation (explained in the above paragraph), the output layer is also computed using the same transfer function as described for *Activation: Hidden Layer*. The default selection is *Sigmoid*.
Training Parameters	Click Training Parameters to open the Training Parameters dialog to specify parameters related to the training of the Neural Network algorithm.
Stopping Rules	Click Stopping Rules to open the Stopping Rules dialog. Here users can specify a comprehensive set of rules for stopping the algorithm early plus cross-validation on the training error.
Bagging Ensemble Method	For more information on each parameter, see the Ensemble Methods chapter within the Data Science Reference Guide.
Number of Weak Learners	This option controls the number of "weak" regression models that will be created. The ensemble method will stop when the number of regression models created reaches the value set for this option. The algorithm will then compute the weighted sum of votes for each class and assign the "winning" value to each record.
Weak Learner	Under Ensemble: Common click the down arrow beneath Weak Leaner to select one of the four featured classifiers: Linear Regression, k-NN, Neural Networks or Decision Tree. The command button to the right will be enabled. Click this command button to control various option settings for the weak leaner.
Random Seed for Bootstrapping	Enter an integer value to specify the seed for random resampling of the training data for each weak learner. Setting the random number seed to a nonzero value (any number of your choice is OK) ensures that the same sequence of random numbers is used each time the dataset is chosen for the classifier. The default value is "12345". If left blank, the random number generator is initialized from the system clock, so the sequence of random numbers will be different in each calculation. If you need the results from

Ensemble Method Bagging Parameter Dialog

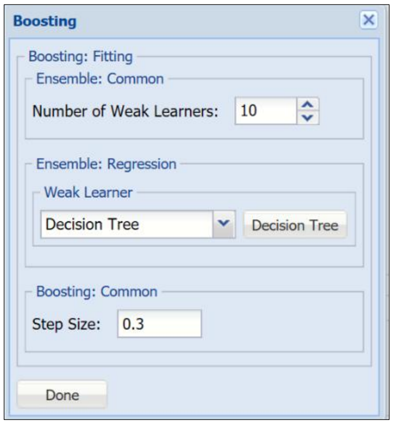


Number of Weak Learners: 10

Weak Learner: Decision Tree

Random Seed for Bootstrapping: 12345

Ensemble Method Boosting Parameter Dialog



Boosting

Boosting: Fitting

Ensemble: Common

Number of Weak Learners: 10

Ensemble: Regression

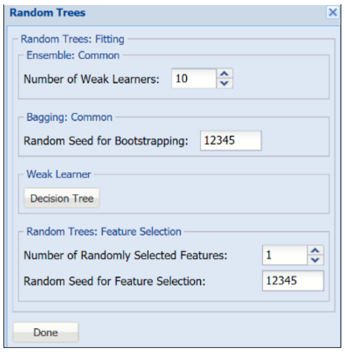
Weak Learner

Decision Tree

Step Size: 0.3

	successive runs of the algorithm to another to be strictly comparable, you should set the seed. To do this, type the desired number you want into the box. This option accepts positive integers with up to 9 digits.
Boosting Ensemble Method	For more information on each parameter, see the Boosting Regression Ensemble Method chapter within the Data Science Reference Guide.
Number of Weak Learners Weak Learner	See description above.
Step Size	The Adaboost algorithm minimizes a loss function using the gradient descent method. The Step size option is used to ensure that the algorithm does not descend too far when moving to the next step. It is recommended to leave this option at the default of 0.3, but any number between 0 and 1 is acceptable. A Step size setting closer to 0 results in the algorithm taking smaller steps to the next point, while a setting closer to 1 results in the algorithm taking larger steps towards the next point.
Random Trees Ensemble Method	For more information on each parameter, see the Boosting Classification Ensemble Method chapter within the Data Science Reference Guide.
Number of Weak Learners Random Seed for Bootstrapping Weak Learner	See description above.
Number of Randomly Selected Features	The Random Trees ensemble method works by training multiple "weak" classification trees using a fixed number of randomly selected features then taking the mode of each class to create a "strong" classifier. The option *Number of randomly selected features* controls the fixed number of randomly selected features in the algorithm. The default setting is **3**.
Feature Selection Random Seed	If an integer value appears for *Feature Selection Random seed*, Analytic Solver Data Science will use this value to set the feature selection random number seed. Setting the random number seed to a nonzero value (any number of your choice is OK) ensures that the same sequence of random numbers is used each time the dataset is chosen for the classifier. The default value is "12345". If left blank, the random number generator is initialized from the system clock, so the sequence of random numbers will be different in each calculation. If you need the results from successive runs of the algorithm to another to be strictly comparable, you should set the seed. To do this, type the desired number you want into the box. This option accepts positive integers with up to 9 digits.

Ensemble Method Random Trees Parameter Dialog



Random Trees

Random Trees: Fitting

Ensemble: Common

Number of Weak Learners: 10

Bagging: Common

Random Seed for Bootstrapping: 12345

Weak Learner

Random Trees: Feature Selection

Number of Randomly Selected Features: 1

Random Seed for Feature Selection: 12345

Find Best Model: Scoring

The two parameters at the bottom of the dialog under Find Best Model: Scoring, determine how well each regression method fits the data.

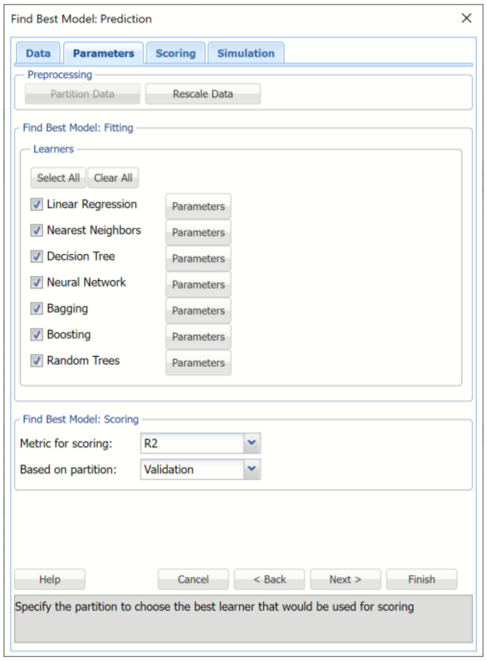
The Metric for Scoring may be changed to R2, SSE, MSE, RMSE or MAD. See the table below for a brief description of each statistic.

- For this example, **leave R2 selected**.
- Select **Validation** for Based on partition.

Statistic	Description
R2	Coefficient of Determination - Examines how differences in one variable can be explained by the difference in a second variable, when predicting the outcome of a given event. RMSE = $\frac{SSR}{SST} = \frac{\sum_i(\hat{y}_i-\bar{y})^2}{\sum_i(y_i-\bar{y})^2}$ where $\hat{y}_i$ is the predicted value for obs i y_i is the actual value for obs i $\bar{y}$ is mean of the y values
SSE	Sum of Squared Error – The sum of the squares of the differences between the actual and predicted values. SSE = $\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ $\hat{y}_i$ is the predicted value for obs i y_i is the actual value for obs i
MSE	Mean Squared Error – The average of the squared differences between the actual and predicted values. MSE = $\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ $\hat{y}_i$ is the predicted value for obs i y_i is the actual value for obs i
RMSE	Root Mean Squared Error – The standard deviation of the residuals. RSE = $\sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i-y_i)^2}{n}}$ $\hat{y}_i$ is the predicted value for obs i y_i is the actual value for obs i
MAD	Mean Absolute Deviation - Average distance between each data value and the sample mean; describes variation in a data set.

	MAD = $\frac{1}{n}\sum_{i=1}^{n}\lvert x_i - \bar{x}\rvert$ where x_i is the ith obs in the sample where $\bar{x}$ is the sample mean

Find Best Model dialog, Parameters tab



☑ Linear Regression
☑ Nearest Neighbors
☑ Decision Tree
☑ Neural Network
☑ Bagging
☑ Boosting
☑ Random Trees

Metric for scoring: R2

Based on partition: Validation

Specify the partition to choose the best learner that would be used for scoring

Scoring tab

Output options are selected on the Scoring tab. By default, the PFBM_Output worksheet will be inserted directly to the right of the STDPartition worksheet.

In this example select all four options: Detailed Report, Summary Report, Lift Charts and Frequency Chart under both Score Training Data and Score Validation Data. Then click Finish to run Find Best Model.

- PFBM_Output contains a listing of all model inputs such a input/output variables and parameter settings for all Learners, as well as Model Performance tables containing evaluations for every available metric, every learner on all available partitions. The Learner identified internally as performing the best, is highlighted in red. (Recall that the statistic used for determining which Learner performs best on the dataset was selected on the Parameters tab.)
- PFBM_Stored contains the PMML (Predictive Model Markup Language) model which can be utilized to score new data. For more information on scoring, see the Scoring chapter that appears later in this guide.

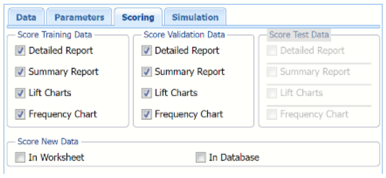
Selecting *Detailed Report* produces PFBM_TrainingScore and PFBM_ValidationScore.

- Both reports contain detailed scoring information on both the Training and Validation partitions using the "best" learner.

Summary Report is selected by default. This option produces a summary report at the top of both PFBM_TrainingScore and PFBM_ValidationScore worksheets.

- *Summary Report* contains a listing of 5 metrics: SSE, MSE, RMSE, MAD and R2.

Find Best Model dialog, Scoring tab



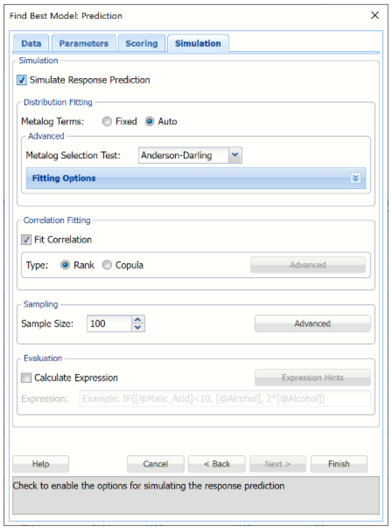
Score Training Data
☑ Detailed Report
☑ Summary Report
☑ Lift Charts
☑ Frequency Chart

Score Validation Data
☑ Detailed Report
☑ Summary Report
☑ Lift Charts
☑ Frequency Chart

Score Test Data
□ Detailed Report
□ Summary Report
□ Lift Charts
□ Frequency Chart

Score New Data
□ In Worksheet
□ In Database

Find Best Model dialog, Simulation tab



Find Best Model: Prediction

Simulation
☑ Simulate Response Prediction

Distribution Fitting
Metalog Terms: ○ Fixed ● Auto
Advanced
Metalog Selection Test: Anderson-Darling

Correlation Fitting
☑ Fit Correlation
Type: ● Rank ○ Copula

Sampling
Sample Size: 100

Evaluation
□ Calculate Expression
Expression: Example: IF([@Malic_Acid]<10, [@Alcohol], 2*[@Alcohol])

Check to enable the options for simulating the response prediction

Selecting *Frequency Chart* produces a frequency graph of the records in the partitions.

- When Frequency Chart is selected, a frequency chart will be displayed when the PFBM_TrainingScore and PFBM_ValidationScore worksheets are selected. This chart will display an interactive application similar to the Analyze Data feature, explained in detail in the Analyze Data chapter that appears earlier in this guide. This chart will include frequency distributions of the actual and predicted responses individually, or side-by-side, depending on the user's preference, as well as basic and advanced statistics for variables, percentiles, six sigma indices.

Selecting *Lift Charts* generates Lift Charts, RROC Curves and Decil-Wise Lift Charts.

See the Scoring chapter that appears later in this guide for more information on the Score New Data section of the Scoring tab.

Simulation Tab

Click **Next** to advance to the Simulation tab.

Select Simulation Response Prediction to enable all options on the Simulation tab of the Find Best Model Prediction dialog.

Simulation tab: All supervised algorithms include a new Simulation tab. This tab uses the functionality from the Generate Data feature (described earlier in this guide) to generate synthetic data based on the training partition, and uses the fitted model to produce predictions for the synthetic data. The resulting report, PFBM_Simulation, will contain the synthetic data, the predicted values and the Excel-calculated Expression column, if present. In addition, frequency charts containing the Predicted, Training, and Expression (if present) sources or a combination of any pair may be viewed, if the charts are of the same type.

Evaluation: If Calculate Expression is selected, Analytic Solver amends an Expression column onto the frequency chart displayed on the PFBM_Simulation output tab. Expression can be any valid Excel formula that references a variable and the response as [@COLUMN_NAME]. Click the *Expression Hints* button for more information on entering an expression.

For the purposes of this example, leave this checkbox *unchecked*. See the Find Best Model classification example above to see an example of this option in use.

For more information on the remaining options shown on this dialog in the Distribution Fitting, Correlation Fitting and Sampling sections, see the Generate Data chapter that appears earlier in this guide.

Click Finish to run Find Best Model on the example dataset.

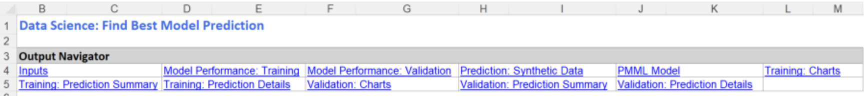
Interpreting the Results

All output sheets are inserted to the right of STDPartition.

PFBM_Output

The PFBM_Output worksheet is inserted directly to the right of the STDPartition worksheet. This report lists all input variables and all parameter settings for each learner, along with the Model Performance of each Learner on all partitions. This example utilizes two partitions, training and validation.

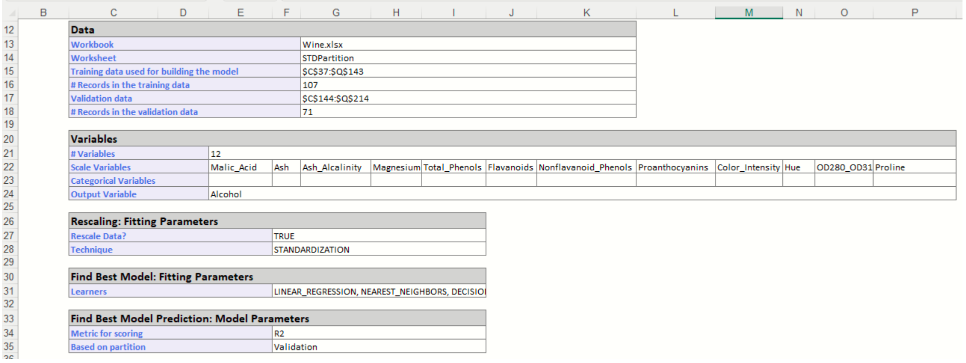
The Output Navigator appears at the very top of all output worksheets. Click the links to easily move between each section of the output. The Output Navigator is listed at the top of each worksheet included in the output.



Data Science: Find Best Model Prediction

Output Navigator					
Inputs	Model Performance: Training	Model Performance: Validation	Prediction: Synthetic Data	PMML Model	Training: Charts
Training: Prediction Summary	Training: Prediction Details	Validation: Charts	Validation: Prediction Summary	Validation: Prediction Details	

The Inputs section includes information pertaining to the dataset, the input variables and parameter settings.



Data	
Workbook	Wine.xlsx
Worksheet	STDPartition
Training data used for building the model	$C$37:$Q$143
# Records in the training data	107
Validation data	$C$144:$Q$214
# Records in the validation data	71

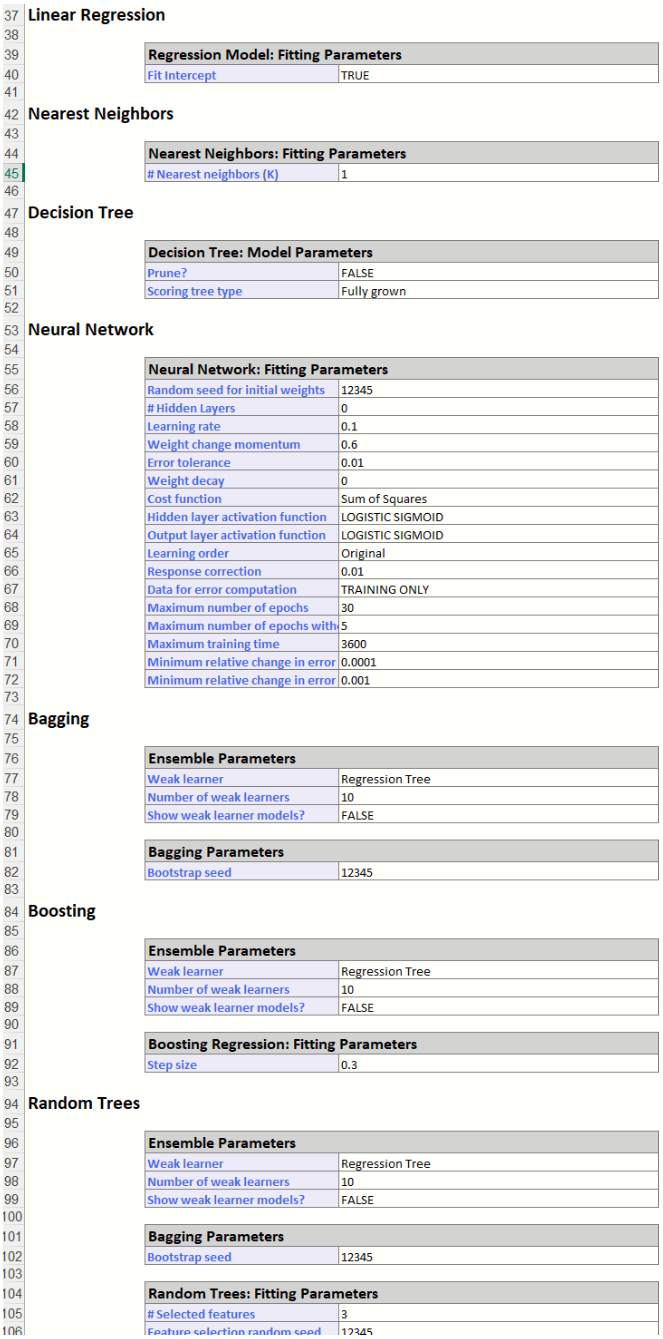
Variables												
# Variables	12											
Scale Variables	Malic_Acid	Ash	Ash_Alcalinity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280_OD31	Proline
Categorical Variables												
Output Variable	Alcohol											

Rescaling: Fitting Parameters	
Rescale Data?	TRUE
Technique	STANDARDIZATION

Find Best Model: Fitting Parameters	
Learners	LINEAR_REGRESSION, NEAREST_NEIGHBORS, DECISIOI

Find Best Model Prediction: Model Parameters	
Metric for scoring	R2
Based on partition	Validation

Further down within Inputs, the parameter selections for each Learner are listed.



Linear Regression

Regression Model: Fitting Parameters	
Fit Intercept	TRUE

Nearest Neighbors

Nearest Neighbors: Fitting Parameters	
# Nearest neighbors (K)	1

Decision Tree

Decision Tree: Model Parameters	
Prune?	FALSE
Scoring tree type	Fully grown

Neural Network

Neural Network: Fitting Parameters	
Random seed for initial weights	12345
# Hidden Layers	0
Learning rate	0.1
Weight change momentum	0.6
Error tolerance	0.01
Weight decay	0
Cost function	Sum of Squares
Hidden layer activation function	LOGISTIC SIGMOID
Output layer activation function	LOGISTIC SIGMOID
Learning order	Original
Response correction	0.01
Data for error computation	TRAINING ONLY
Maximum number of epochs	30
Maximum number of epochs with	5
Maximum training time	3600
Minimum relative change in error	0.0001
Minimum relative change in error	0.001

Bagging

Ensemble Parameters	
Weak learner	Regression Tree
Number of weak learners	10
Show weak learner models?	FALSE

Bagging Parameters	
Bootstrap seed	12345

Boosting

Ensemble Parameters	
Weak learner	Regression Tree
Number of weak learners	10
Show weak learner models?	FALSE

Boosting Regression: Fitting Parameters	
Step size	0.3

Random Trees

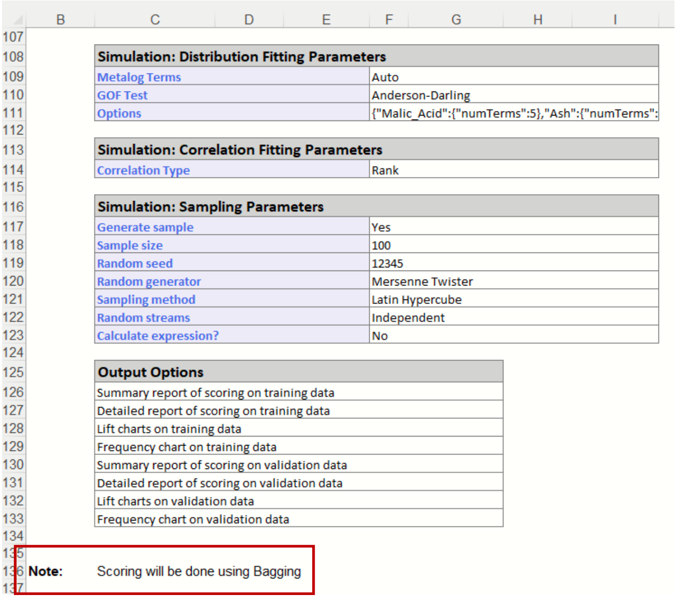
Ensemble Parameters	
Weak learner	Regression Tree
Number of weak learners	10
Show weak learner models?	FALSE

Bagging Parameters	
Bootstrap seed	12345

Random Trees: Fitting Parameters	
# Selected features	3
Feature selection random seed	12345

Scroll down to view Simulation tab option settings and any generated messages from the Find Best Model feature.

Further down, the Model Performance tables display how each prediction method performed on the dataset.



Simulation: Distribution Fitting Parameters	
Metalog Terms	Auto
GOF Test	Anderson-Darling
Options	{"Malic_Acid":{"numTerms":5},"Ash":{"numTerms":

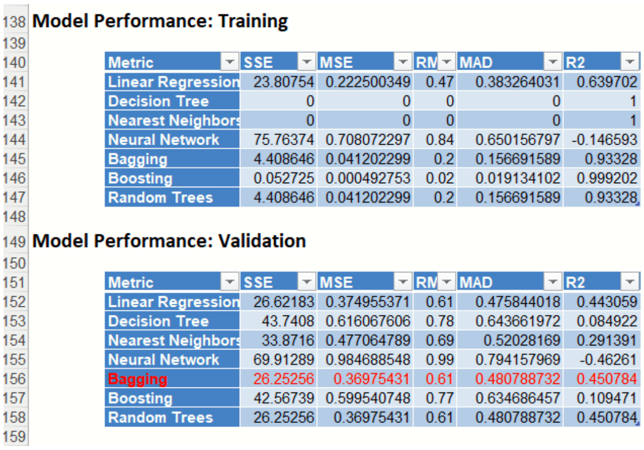
Simulation: Correlation Fitting Parameters	
Correlation Type	Rank

Simulation: Sampling Parameters	
Generate sample	Yes
Sample size	100
Random seed	12345
Random generator	Mersenne Twister
Sampling method	Latin Hypercube
Random streams	Independent
Calculate expression?	No

Output Options
Summary report of scoring on training data
Detailed report of scoring on training data
Lift charts on training data
Frequency chart on training data
Summary report of scoring on validation data
Detailed report of scoring on validation data
Lift charts on validation data
Frequency chart on validation data

Note: Scoring will be done using Bagging

The Messages portion of the report indicates that Scoring will be performed using the Bagging Ensemble Method the Learner selected as the "best" choice according to the selection for Find Best Model: Scoring parameters on the Parameters tab: Validation Partition R2 Metric.



Model Performance: Training

Metric	SSE	MSE	RM	MAD	R2
Linear Regression	23.80754	0.222500349	0.47	0.383264031	0.639702
Decision Tree	0	0	0	0	1
Nearest Neighbors	0	0	0	0	1
Neural Network	75.76374	0.708072297	0.84	0.650156797	-0.146593
Bagging	4.408646	0.041202299	0.2	0.156691589	0.93328
Boosting	0.052725	0.000492753	0.02	0.019134102	0.999202
Random Trees	4.408646	0.041202299	0.2	0.156691589	0.93328

Model Performance: Validation

Metric	SSE	MSE	RM	MAD	R2
Linear Regression	26.62183	0.374955371	0.61	0.475844018	0.443059
Decision Tree	43.7408	0.616067606	0.78	0.643661972	0.084922
Nearest Neighbors	33.8716	0.477064789	0.69	0.52028169	0.291391
Neural Network	69.91289	0.984688548	0.99	0.794157969	-0.46261
Bagging	26.25256	0.36975431	0.61	0.480788732	0.450784
Boosting	42.56739	0.599540748	0.77	0.634686457	0.109471
Random Trees	26.25256	0.36975431	0.61	0.480788732	0.450784

Since the Bagging Ensemble R2 metric for the Validation Partition has the highest score, that is the Learner that will be used for scoring.

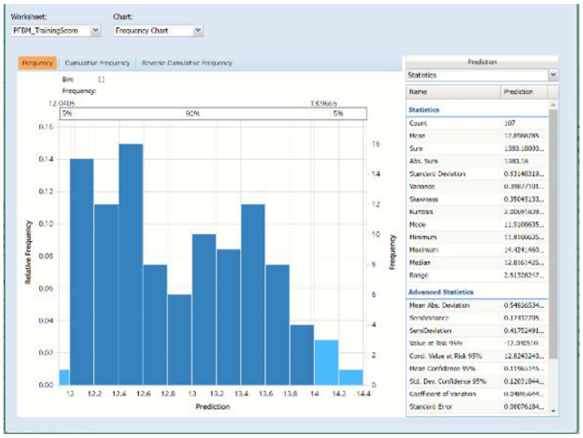
PFBM_TrainingScore and PFBM_ValidationScore

PFBM_TrainingScore contains the Prediction Summary and the Prediction Details reports for the training partition. PFBM_Validation contains the same reports for the validation partition.   Both reports have been generated using the Bagging Ensemble Method, as discussed above.

PFBM_TrainingScore

Click the *PFBM_TrainingScore* tab to view the newly added Output Variable frequency chart, the Training: Prediction Summary and the Training: Prediction Details report. All calculations, charts and predictions on this worksheet apply to the Training data.

> Note: To view charts in the Cloud app, click the Charts icon on the Ribbon, select a worksheet under Worksheet and a chart under Chart.



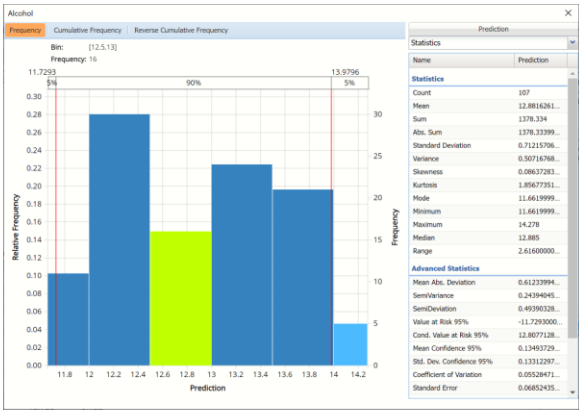
Prediction

Statistics

Name	Prediction
Statistics	
Count	107
Mean	12.8998785
Sum	1380.18000...
Abs. Sum	1380.18
Standard Deviation	0.63148319...
Variance	0.39877101...
Skewness	0.35045133...
Kurtosis	2.80691839...
Mode	11.9108635...
Minimum	11.9108635...
Maximum	14.4241460...
Median	12.8161425...
Range	2.51328247...
Advanced Statistics	
Mean Abs. Deviation	0.54926534...
SemiVariance	0.17432705
SemiDeviation	0.41752491...
Value at Risk 95%	-12.040510
Cond. Value at Risk 95%	12.8243240...
Mean Confidence 95%	0.11965145...
Std. Dev. Confidence 95%	0.12031844...
Coefficient of Variation	0.04895644...
Standard Error	0.06076184...

Tabs: The Analyze Data dialog contains three tabs: Frequency, Cumulative Frequency, and Reverse Cumulative Frequency. Each tab displays different information about the distribution of variable values.

Analyze Data Dialog



Prediction

Statistics

Name	Prediction
Statistics	
Count	107
Mean	12.8816261...
Sum	1378.334
Abs. Sum	1378.33399...
Standard Deviation	0.71215706...
Variance	0.50716768...
Skewness	0.08637283...
Kurtosis	1.85677351...
Mode	11.6619999...
Minimum	11.6619999...
Maximum	14.278
Median	12.885
Range	2.61600000...
Advanced Statistics	
Mean Abs. Deviation	0.61233994...
SemiVariance	0.24394045...
SemiDeviation	0.49390328...
Value at Risk 95%	-11.7293000...
Cond. Value at Risk 95%	12.8077128...
Mean Confidence 95%	0.13493729...
Std. Dev. Confidence 95%	0.13312297...
Coefficient of Variation	0.05528471...
Standard Error	0.06852435...

Hovering over a bar in either of the three charts will populate the Bin and Frequency headings at the top of the chart. In the Frequency chart above, the bar for the [12.5, 13] Bin is selected. This bar has a frequency of 16 and a relative frequency of about 15%.

By default, red vertical lines will appear at the 5% and 95% percentile values in all three charts, effectively displaying the 90th confidence interval. The middle percentage is the percentage of all the variable values that lie within the 'included' area, i.e. the darker shaded area. The two percentages on each end are the percentage of all variable values that lie outside of the 'included' area or the "tails". i.e. the lighter shaded area. Percentile values can be altered by moving either red vertical line to the left or right.

Click the "X" in the upper right corner of the detailed chart dialog to close the dialog. To re-open the chart, click a new tab, say the Data tab in this example, and then click PFBM_TrainingScore.

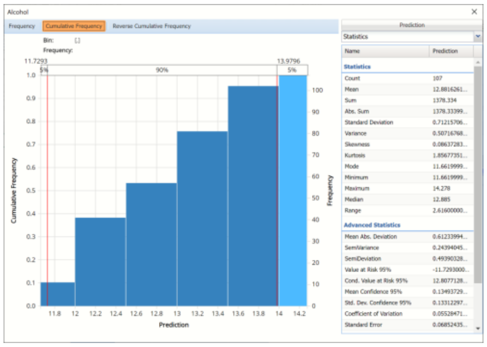
Frequency Tab: When the Analyze Data dialog is first displayed, the Frequency tab is selected by default. This tab displays a histogram of the variable's values.

Bins containing the range of values for the variable appear on the horizontal axis, the relative frequency of occurrence of the bin values appears on the left vertical axis while the actual frequency of the bin values appear on the right vertical axis.

Cumulative Frequency / Reverse Cumulative Frequency

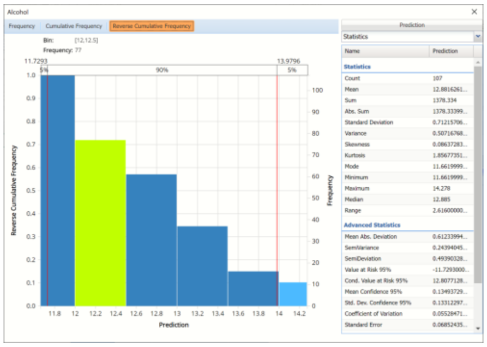
The Cumulative Frequency tab displays a chart of the cumulative form of the frequency chart, as shown below. Hover over each bar to populate the Bin and Frequency headings at the top of the chart. In this screenshot below, the bar for the [12.5, 13.0] Bin is selected in the Cumulative Frequency Chart. This bar has a frequency of 57 and a relative frequency of about 52%.

Cumulative Frequency Chart



Name	Prediction
Statistics	
Count	107
Mean	12.8816261...
Sum	1378.334
Abs. Sum	1378.33399...
Standard Deviation	0.71215706...
Variance	0.50716768...
Skewness	0.08637283...
Kurtosis	1.85677351...
Mode	11.6619999...
Minimum	11.6619999...
Maximum	14.278
Median	12.885
Range	2.61600000...
Advanced Statistics	
Mean Abs. Deviation	0.61233994...
SemiVariance	0.24394045...
SemiDeviation	0.49390328...
Value at Risk 95%	-11.7293000...
Cond. Value at Risk 95%	12.8077128...
Mean Confidence 95%	0.13493729...
Std. Dev. Confidence 95%	0.13312297...
Coefficient of Variation	0.05528471...
Standard Error	0.06852435...

Reverse Cumulative Frequency Chart



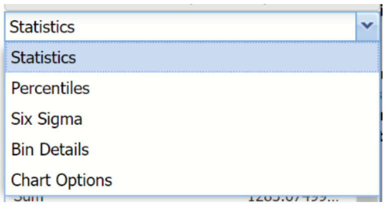
Name	Prediction
Statistics	
Count	107
Mean	12.8816261...
Sum	1378.334
Abs. Sum	1378.33399...
Standard Deviation	0.71215706...
Variance	0.50716768...
Skewness	0.08637283...
Kurtosis	1.85677351...
Mode	11.6619999...
Minimum	11.6619999...
Maximum	14.278
Median	12.885
Range	2.61600000...
Advanced Statistics	
Mean Abs. Deviation	0.61233994...
SemiVariance	0.24394045...
SemiDeviation	0.49390328...
Value at Risk 95%	-11.7293000...
Cond. Value at Risk 95%	12.8077128...
Mean Confidence 95%	0.13493729...
Std. Dev. Confidence 95%	0.13312297...
Coefficient of Variation	0.05528471...
Standard Error	0.06852435...

Cumulative Frequency Chart: Bins containing the range of values for the variable appear on the horizontal axis, the cumulative frequency of occurrence of the bin values appear on the left vertical axis while the actual cumulative frequency of the bin values appear on the right vertical axis.

Reverse Cumulative Frequency Chart: Bins containing the range of values for the variable appear on the horizontal axis, similar to the Cumulative Frequency Chart. The reverse cumulative frequency of occurrence of the bin values appear on the left vertical axis while the actual reverse cumulative frequency of the bin values appear on the right vertical axis.

Click the drop down menu on the upper right of the dialog to display additional panes: Statistics, Six Sigma and Percentiles.

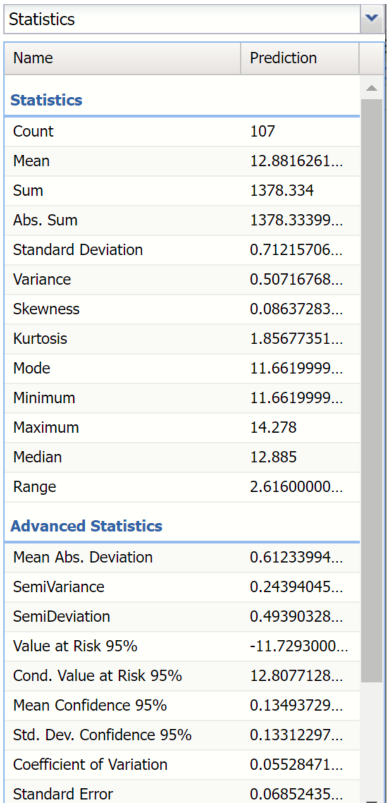
Drop down menu



Statistics View

The Statistics tab displays numeric values for several summary statistics, computed from all values for the specified variable. The statistics shown on the pane below were computed for the predicted variable, in this case Alcohol.

Statistics Pane



Statistics

Name	Prediction
Statistics	
Count	107
Mean	12.8816261...
Sum	1378.334
Abs. Sum	1378.33399...
Standard Deviation	0.71215706...
Variance	0.50716768...
Skewness	0.08637283...
Kurtosis	1.85677351...
Mode	11.6619999...
Minimum	11.6619999...
Maximum	14.278
Median	12.885
Range	2.61600000...
Advanced Statistics	
Mean Abs. Deviation	0.61233994...
SemiVariance	0.24394045...
SemiDeviation	0.49390328...
Value at Risk 95%	-11.7293000...
Cond. Value at Risk 95%	12.8077128...
Mean Confidence 95%	0.13493729...
Std. Dev. Confidence 95%	0.13312297...
Coefficient of Variation	0.05528471...
Standard Error	0.06852435...

All statistics appearing on the Statistics pane are briefly described below.

Statistics

- **Count**, the total number of records included in the data.
- **Mean**, the average of all the values.
- **Sum**, the sum of all the values.
- **Abs. Sum**, the absolute sum of all the values.
- **Standard Deviation**, the square root of variance.
- **Variance,** describes the spread of the distribution of values.
- **Skewness**, which describes the *asymmetry* of the distribution of values.
- **Kurtosis**, which describes the *peakedness* of the distribution of values.
- **Mode**, the most frequently occurring single value.
- **Minimum**, the minimum value attained.
- **Maximum**, the maximum value attained.
- **Median,** the median value.
- **Range**, the difference between the maximum and minimum values.

Advanced Statistics

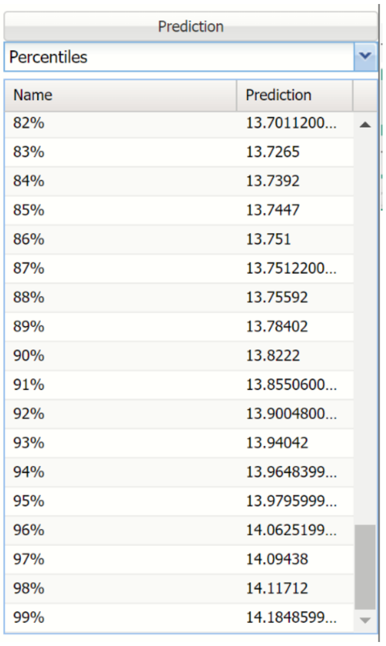
- **Mean Abs. Deviation**, returns the average of the absolute deviations.
- **SemiVariance**, measure of the dispersion of values.

- **SemiDeviation**, *one-sided* measure of dispersion of values.
- **Value at Risk 95%**, the maximum loss that can occur at a given confidence level.
- **Cond. Value at Risk**, is defined as the *expected value* of a loss *given that* a loss at the specified percentile occurs.
- **Mean Confidence 95%**, returns the confidence “half-interval” for the estimated mean value (returned by the PsiMean() function).
- **Std. Dev. Confidence 95%**, returns the confidence ‘half-interval’ for the estimated standard deviation of the simulation trials (returned by the PsiStdDev() function).
- **Coefficient of Variation**, is defined as the ratio of the standard deviation to the mean.
- **Standard Error**, defined as the standard deviation of the sample mean.
- **Expected Loss**, returns the average of all negative data multiplied by the percentrank of 0 among all data.
- **Expected Loss Ratio**, returns the expected loss ratio.
- **Expected Gain** returns the average of all positive data multiplied by 1 - percentrank of 0 among all data.
- **Expected Gain Ratio**, returns the expected gain ratio**.**
- **Expected Value Margin**, returns the expected value margin.

Percentiles View

Selecting Percentiles from the menu displays numeric percentile values (from 1% to 99%) computed using all values for the variable. The percentiles shown below were computed using the values for the Alcohol variable.

Percentiles Pane



Name	Prediction
82%	13.7011200...
83%	13.7265
84%	13.7392
85%	13.7447
86%	13.751
87%	13.7512200...
88%	13.75592
89%	13.78402
90%	13.8222
91%	13.8550600...
92%	13.9004800...
93%	13.94042
94%	13.9648399...
95%	13.9795999...
96%	14.0625199...
97%	14.09438
98%	14.11712
99%	14.1848599...

The values displayed here represent 99 equally spaced points on the Cumulative

Frequency chart: In the Percentile column, the numbers rise smoothly on the vertical axis, from 0 to 1.0, and in the Value column, the corresponding values from the horizontal axis are shown. For example, the 75th Percentile value is a number such that three-quarters of the values occurring in the last simulation are less than or equal to this value.

Six Sigma View

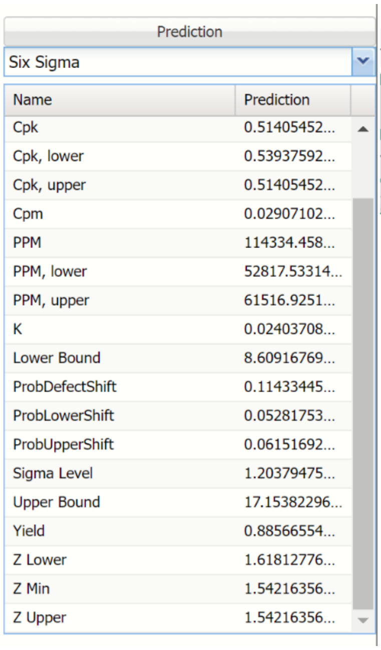
Selecting Six Sigma from the menu displays various computed Six Sigma measures. In this display, the red vertical lines on the chart are the Lower Specification Limit (LSL) and the Upper Specification Limit (USL) which are initially set equal to the 5th and 95th percentile values, respectively.

These functions compute values related to the Six Sigma indices used in manufacturing and process control. For more information on these functions, see the Appendix located at the end of the Data Science Reference Guide.

- **SigmaCP** calculates the Process Capability.
- **SigmaCPK** calculates the Process Capability Index.
- **SigmaCPKLower** calculates the one-sided Process Capability Index based on the Lower Specification Limit.
- **SigmaCPKUpper** calculates the one-sided Process Capability Index based on the Upper Specification Limit.
- **SigmaCPM** calculates the Taguchi Capability Index.
- **SigmaDefectPPM** calculates the Defect Parts per Million statistic.
- **SigmaDefectShiftPPM** calculates the Defective Parts per Million statistic with a Shift.
- **SigmaDefectShiftPPMLower** calculates the Defective Parts per Million statistic with a Shift below the Lower Specification Limit.
- **SigmaDefectShiftPPMUpper** calculates the Defective Parts per Million statistic with a Shift above the Upper Specification Limit.
- **SigmaK** calculates the Measure of Process Center.
- **SigmaLowerBound** calculates the Lower Bound as a specific number of standard deviations below the mean.
- **SigmaProbDefectShift** calculates the Probability of Defect with a Shift outside the limits.
- **SigmaProbDefectShiftLower** calculates the Probability of Defect with a Shift below the lower limit.
- **SigmaProbDefectShiftUpper** calculates the Probability of Defect with a Shift above the upper limit.
- **SigmaSigmaLevel** calculates the Process Sigma Level with a Shift.
- **SigmaUpperBound** calculates the Upper Bound as a specific number of standard deviations above the mean.
- **SigmaYield** calculates the Six Sigma Yield with a shift, i.e. the fraction of the process that is free of defects.
- **SigmaZLower** calculates the number of standard deviations of the process that the lower limit is below the mean of the process.

- **SigmaZMin** calculates the minimum of ZLower and ZUpper.
- **SigmaZUpper** calculates the number of standard deviations of the process that the upper limit is above the mean of the process.

Six Sigma Pane



Prediction

Six Sigma

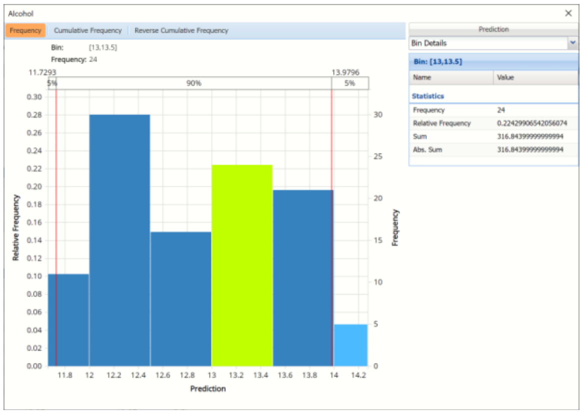
Name	Prediction
Cpk	0.51405452...
Cpk, lower	0.53937592...
Cpk, upper	0.51405452...
Cpm	0.02907102...
PPM	114334.458...
PPM, lower	52817.53314...
PPM, upper	61516.9251...
K	0.02403708...
Lower Bound	8.60916769...
ProbDefectShift	0.11433445...
ProbLowerShift	0.05281753...
ProbUpperShift	0.06151692...
Sigma Level	1.20379475...
Upper Bound	17.15382296...
Yield	0.88566554...
Z Lower	1.61812776...
Z Min	1.54216356...
Z Upper	1.54216356...

Bin Details View

Click the down arrow next to Statistics to view Bin Details for each bin in the chart.

Bin: If viewing the chart with only the Predicted or simulate data, only one grid will be displayed on the Bin Details pane. This grid displays important bin statistics such as frequency, relative frequency, sum and absolute sum.

Bin Details View with continuous (scale) variables



Prediction

Bin Details

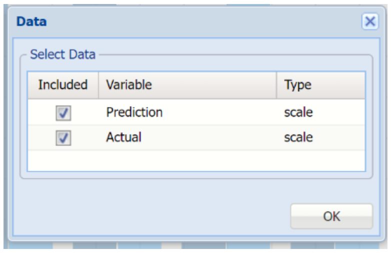
Bin: [13,13.5]

Name	Value
Statistics	
Frequency	24
Relative Frequency	0.22429906542056074
Sum	316.84399999999994
Abs. Sum	316.84399999999994

- Frequency is the number of observations assigned to the bin.
- Relative Frequency is the number of observations assigned to the bin divided by the total number of observations.
- Sum is the sum of all observations assigned to the bin.

- Absolute Sum is the sum of the absolute value of all observations assigned to the bin, i.e. |observation 1| + |observation 2| + |observation 3| + …

Bin Differences: Click *Prediction* to open the Data dialog. Select both Prediction and Actual to add the predicted values for the training partition to the chart.

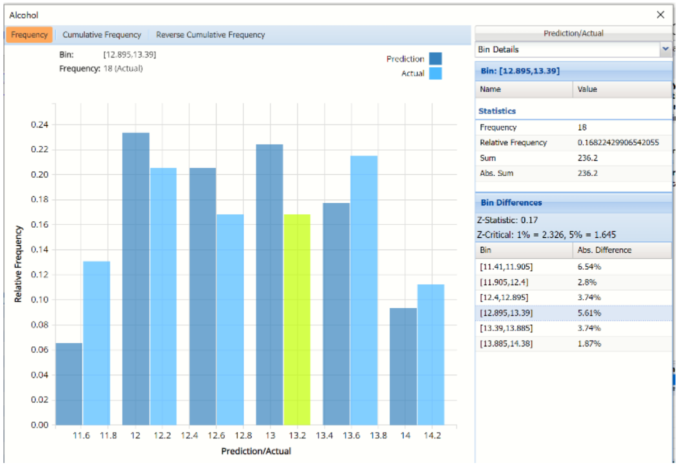


Data

Select Data

Included	Variable	Type
☑	Prediction	scale
☑	Actual	scale

Two grids are displayed, Bin and Bin Differences. Bin Differences displays the differences between the relative frequencies of each bin for the two histograms, sorted in the same order as the bins listed in the chart. The computed Z-Statistic as well as the critical values, are displayed in the title of the grid.



Bin: [12.895,13.39]
Frequency: 18 (Actual)

Bin: [12.895,13.39]

Name	Value
Statistics	
Frequency	18
Relative Frequency	0.16822429906542055
Sum	236.2
Abs. Sum	236.2

Bin Differences
Z-Statistic: 0.17
Z-Critical: 1% = 2.326, 5% = 1.645

Bin	Abs. Difference
[11.41,11.905]	6.54%
[11.905,12.4]	2.8%
[12.4,12.895]	3.74%
[12.895,13.39]	5.61%
[13.39,13.885]	3.74%
[13.885,14.38]	1.87%

For more information on Bin Details, see the *Generate Data* chapter within the **Data Science Reference Guide**.

Chart Settings View

The Chart Options view contains controls that allow you to customize the appearance of the charts that appear in the dialog. When you change option selections or type numerical values in these controls, the chart area is instantly updated.

Chart Options Pane

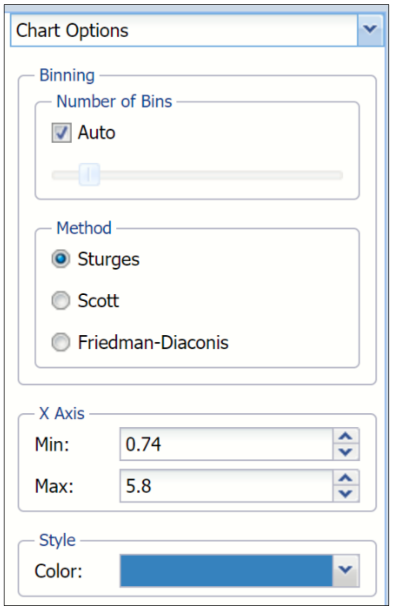


Chart Options

Binning

Number of Bins

☑ Auto

Method

● Sturges

○ Scott

○ Friedman-Diaconis

X Axis

Min: 0.74

Max: 5.8

Style

Color:

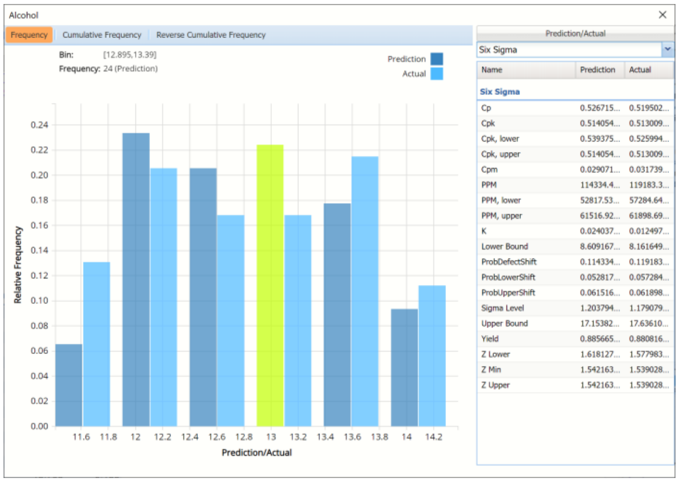
The controls are divided into three groups: Binning, Method and Style.

- **Binning**: Applies to the number of bins in the chart.
  - **Auto:** Select Auto to allow Analytic Solver to automatically select the appropriate number of bins to be included in the frequency charts. See Method below for information on how to change the bin generator used by Analytic Solver when this option is selected.
  - **Manually select # of Bins:** To manually select the number of bins used in the frequency charts, uncheck "Auto" and drag the slider to the right to increase the number of bins or to the left to decrease the number of bins.
- **Method:** Three generators are included in the Analyze Data application to generate the "optimal" number of bins displayed in the chart. All three generators implicitly assume a normal distribution. Sturges is the default setting. The Scott generator should be used with random samples of normally distributed data. The Freedman-Diaconis' generator is less sensitive than the standard deviation to outliers in the data.
- **X Axis:** Analytic Solver allows users to manually set the Min and Max values for the X Axis. Simply type the desired value into the appropriate text box.
- **Style**:
  - **Color**: Select a color, to apply to the entire variable graph, by clicking the down arrow next to Color and then selecting the desired hue.

Notice in the screenshot below that both the Prediction and Actual data appear in the chart together, and statistics, percentiles or Six Sigma indices for both data appear on the right.

To remove either the Predicted or the Actual data from the chart, click Prediction/Actual in the top right and then uncheck the data type to be removed.

Frequency Chart shown with Prediction and Training data, Six Sigma displayed.



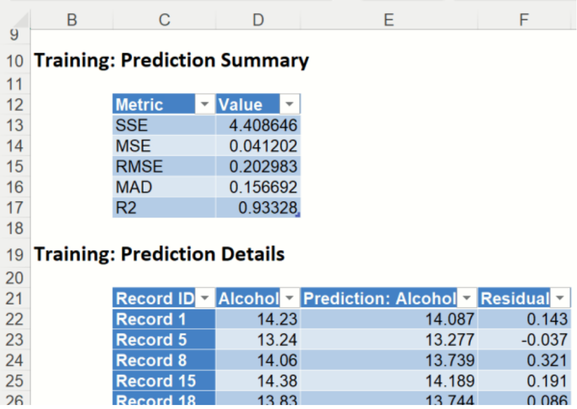
Bin: [12.895,13.39]
Frequency: 24 (Prediction)

Name	Prediction	Actual
Six Sigma		
Cp	0.526715...	0.519502...
Cpk	0.514054...	0.513009...
Cpk, lower	0.539375...	0.525994...
Cpk, upper	0.514054...	0.513009...
Cpm	0.029071...	0.031739...
PPM	114334.4...	119183.3...
PPM, lower	52817.53...	57284.64...
PPM, upper	61516.92...	61898.69...
K	0.024037...	0.012497...
Lower Bound	8.609167...	8.161649...
ProbDefectShift	0.114334...	0.119183...
ProbLowerShift	0.052817...	0.057284...
ProbUpperShift	0.061516...	0.061898...
Sigma Level	1.203794...	1.179079...
Upper Bound	17.15382...	17.63610...
Yield	0.885665...	0.880816...
Z Lower	1.618127...	1.577983...
Z Min	1.542163...	1.539028...
Z Upper	1.542163...	1.539028...

Click the down arrow next to Statistics to view Percentiles or Six Sigma indices for each type of data.

Training: Prediction Summary and Prediction Details

The Prediction Summary for the Training Partition lists the following metrics: SSE, MSE, RMSE, MAD and R2. See definitions above.



Training: Prediction Summary

Metric	Value
SSE	4.408646
MSE	0.041202
RMSE	0.202983
MAD	0.156692
R2	0.93328

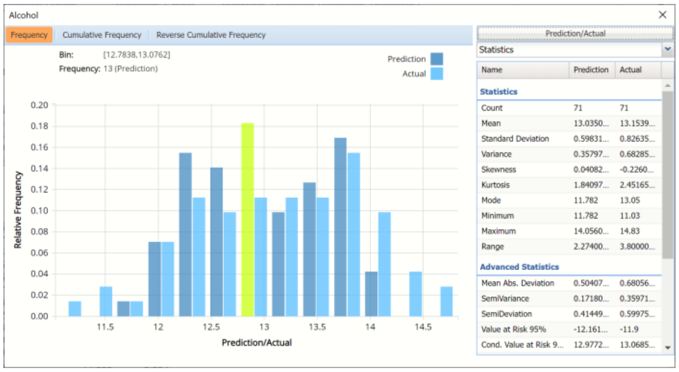
Training: Prediction Details

Record ID	Alcohol	Prediction: Alcohol	Residual
Record 1	14.23	14.087	0.143
Record 5	13.24	13.277	-0.037
Record 8	14.06	13.739	0.321
Record 15	14.38	14.189	0.191
Record 18	13.83	13.744	0.086

Individual records and their predictions are shown beneath Training: Prediction Details.

PFBM_ValidationScore

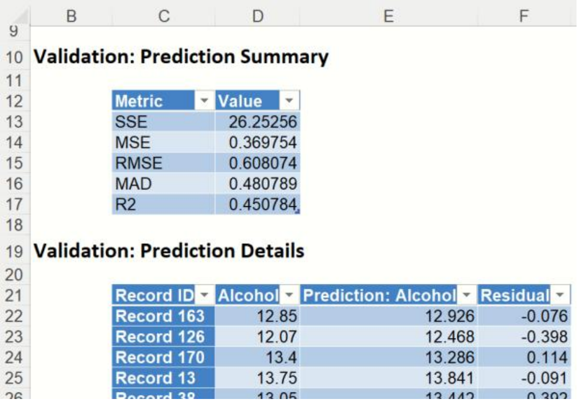
Frequency Chart: PFBM_ValidationScore also displays a frequency chart once the tab is selected. See above for an explanation of this chart.



Bin: [12.7838,13.0762]
Frequency: 13 (Prediction)

Name	Prediction	Actual
Statistics		
Count	71	71
Mean	13.0350...	13.1539...
Standard Deviation	0.59831...	0.82635...
Variance	0.35797...	0.68285...
Skewness	0.04082...	-0.2260...
Kurtosis	1.84097...	2.45165...
Mode	11.782	13.05
Minimum	11.782	11.03
Maximum	14.0560...	14.83
Range	2.27400...	3.80000...
Advanced Statistics		
Mean Abs. Deviation	0.50407...	0.68056...
SemiVariance	0.17180...	0.35971...
SemiDeviation	0.41449...	0.59975...
Value at Risk 95%	-12.161...	-11.9
Cond. Value at Risk 9...	12.9772...	13.0685...

The Prediction Summary for the Validation Partition is shown below.



Validation: Prediction Summary

Metric	Value
SSE	26.25256
MSE	0.369754
RMSE	0.608074
MAD	0.480789
R2	0.450784

Validation: Prediction Details

Record ID	Alcohol	Prediction: Alcohol	Residual
Record 163	12.85	12.926	-0.076
Record 126	12.07	12.468	-0.398
Record 170	13.4	13.286	0.114
Record 13	13.75	13.841	-0.091
Record 38	13.05	13.442	0.392

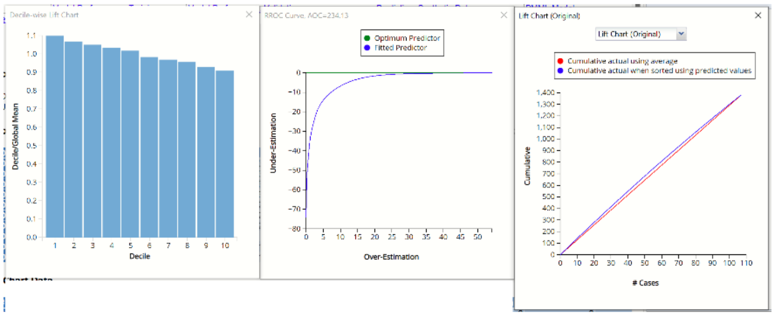
Individual records and their predictions are shown beneath Validation: Prediction Details.

PFBM_TrainingLiftChart and PFBM_ValidationLiftChart

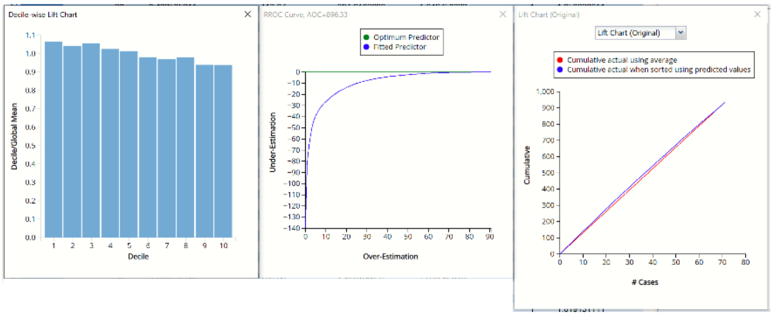
Lift charts and Regression ROC Curves (RROC) are visual aids for measuring model performance. Lift Charts consist of a lift curve and a baseline. The greater the area between the lift curve and the baseline, the better the model. R\] (regression receiver operating characteristic) curves plot the performance of regressors by graphing over-estimations (or predicted values that are too high) versus underestimations (or predicted values that are too low.) The closer the curve is to the top left corner of the graph (in other words, the smaller the area above the curve), the better the performance of the model.

Note: To view these charts in the Cloud app, click the Charts icon on the Ribbon, select PFBM_TrainingLiftChart or PFBM_ValidationLiftChart for Worksheet and Decile Chart, RROC Chart or Gain Chart for Chart.

Decile-wise Lift Chart, RROC Curve, and Lift Charts for Training Partition



Decile-wise Lift Chart, RROC Curve, and Lift Charts for Valid. Partition



Original Lift Chart

After the model is built using the training data set, the model is used to score on the training data set and the validation data set (if one exists). Then the data set(s) are sorted using the predicted output variable value. After sorting, the actual outcome values of the output variable are cumulated and the lift curve is drawn as the number of cases versus the cumulated value. The baseline (red line connecting the origin to the end point of the blue line) is drawn as the number of cases versus the average of actual output variable values multiplied by the number of cases.

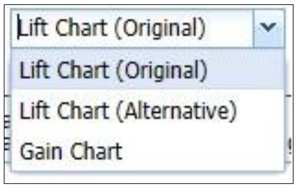
Decile-Wise Lift Chart

The decile-wise lift curve is drawn as the decile number versus the cumulative actual output variable value divided by the decile's mean output variable value. This bars in this chart indicate the factor by which the Linear Regression model outperforms a random assignment, one decile at a time. Typically, this graph will have a "stairstep" appearance - the bars will descend in order from left to right as shown in the decile-wise charts for both partitions.

RRoc Curve

The Regression ROC curve (RROC) was updated in V2017. This new chart compares the performance of the regressor (Fitted Predictor) with an Optimum Predictor Curve. The Optimum Predictor Curve plots a hypothetical model that would provide perfect prediction results. The best possible prediction performance is denoted by a point at the top left of the graph at the intersection of the x and y axis. Area Over the Curve (AOC) is the space in the graph that appears above the RROC curve and is calculated using the formula: sigma^2 * n^2/2 where n is the number of records The smaller the AOC, the better the performance of the model.

In V2017, two new charts were introduced: a new Lift Chart and the Gain Chart. To display these new charts, click the down arrow next to Lift Chart (Original), in the Original Lift Chart, then select the desired chart.

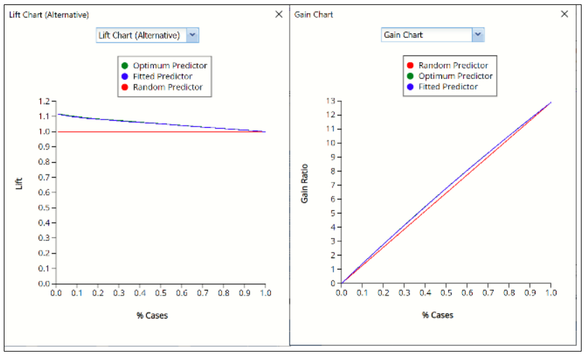


Lift Chart Alternative and Gain Chart

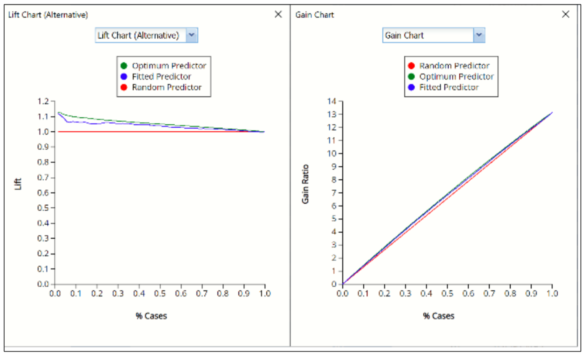
Select Lift Chart (Alternative) to display Analytic Solver Data Science's alternative Lift Chart. Each of these charts consists of an Optimum Predictor curve, a Fitted Predictor curve, and a Random Predictor curve. The Optimum Predictor curve plots a hypothetical model that would provide perfect classification for our data. The Fitted Predictor curve plots the fitted model and the Random Predictor curve plots the results from using no model or by using a random guess (i.e. for x% of selected observations, x% of the total number of positive observations are expected to be correctly classified).

Click the down arrow and select Gain Chart from the menu. In this chart, the Gain Ratio is plotted against the % Cases.

Lift Chart (Alternative) and Gain Chart for Training Partition



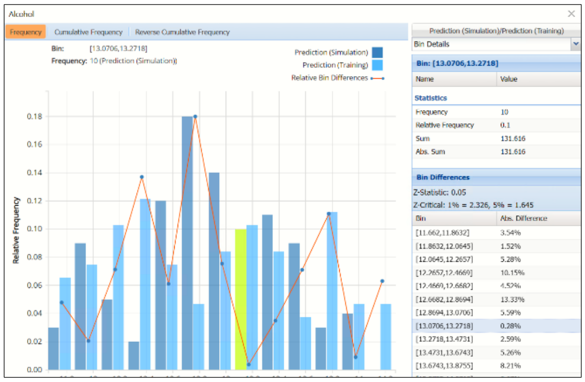
Lift Chart (Alternative) and Gain Chart for Validation Partition



PFBM_Simulation

As discussed above, Analytic Solver Data Science generates a new output worksheet, PFBM_Simulation, when *Simulate Response Prediction* is selected on the Simulation tab of the Find Best Model dialog.

This report contains the prediction data for the synthetic data, the training data (using the fitted model) and the Excel – calculated Expression column, if populated in the dialog. Users may switch between between the Predicted, Training, and Expression data or a combination of two, as long as they are of the same type. Recall that Expression was not used in this example. For more information on this chart, see above.



Bin: [13.0706,13.2718]
Frequency: 10 (Prediction (Simulation))

Bin: [13.0706,13.2718]

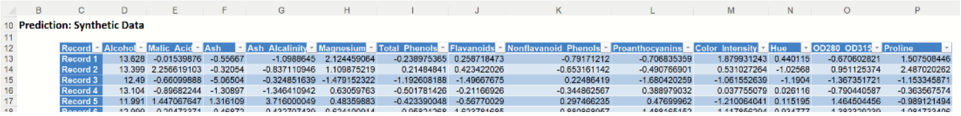
Name	Value
Statistics	
Frequency	10
Relative Frequency	0.1
Sum	131.616
Abs. Sum	131.616

Bin Differences
Z-Statistic: 0.05
Z-Critical: 1% = 2.326, 5% = 1.645

Bin	Abs. Difference
[11.662,11.8632]	3.54%
[11.8632,12.0645]	1.52%
[12.0645,12.2657]	5.28%
[12.2657,12.4669]	10.15%
[12.4669,12.6682]	4.52%
[12.6682,12.8694]	13.33%
[12.8694,13.0706]	5.59%
[13.0706,13.2718]	0.28%
[13.2718,13.4731]	2.59%
[13.4731,13.6743]	5.26%
[13.6743,13.8755]	8.21%

Notice that red lines which connect the relative Bin Differences for each bin. Bin Differences are computed based on the frequencies of records which predictions fall into each bin. For example, consider the highlighted bin in the screenshot above [x0, x1] = [13.0706, 13.2718]. There are 10 Simulation records and 11 Training records in this bin. The relative frequency of the Simulation data is 10/100 = 10% and the relative frequency of the Training data is 11/107 = 10.28%. Hence the Absolute Difference (in frequencies) is = |10 – 10.3| = .28%.

The generated synthetic data is included in the Prediction: Synthetic Data report.

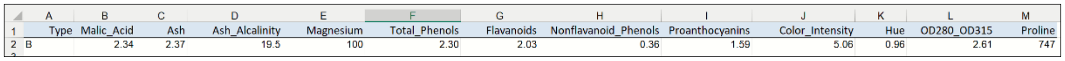


Prediction: Synthetic Data

Record	Alcohol	Malic_Acid	Ash	Ash_Alcalinity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280_OD315	Proline
Record 1	13.628	-0.01539876	-0.55667	-1.0988645	2.124459064	-0.238975365	0.258718473	-0.79171212	-0.706835359	1.879931243	0.440115	-0.670602821	1.507508446
Record 2	13.399	2.256619103	-0.32054	-0.837110946	1.109875219	0.21484841	0.423422026	-0.653161142	-0.490766901	0.531027264	-1.02568	0.951125374	2.487020262
Record 3	12.49	-0.66099888	-5.06504	-0.324851639	-1.479152322	-1.192608188	-1.49667675	0.22486419	-1.680420259	-1.061552639	-1.1904	-1.367351721	-1.153345871
Record 4	13.104	-0.89682244	-1.30897	-1.346410942	0.63059763	-0.501781426	-0.21166926	-0.344862567	0.388979032	0.037755079	0.026116	-0.790440587	-0.363567574
Record 5	11.991	1.447067647	1.316109	3.716000049	0.48359883	-0.423390048	-0.56770029	0.297466235	0.47699962	-1.210064041	0.115195	1.464504456	-0.989121494

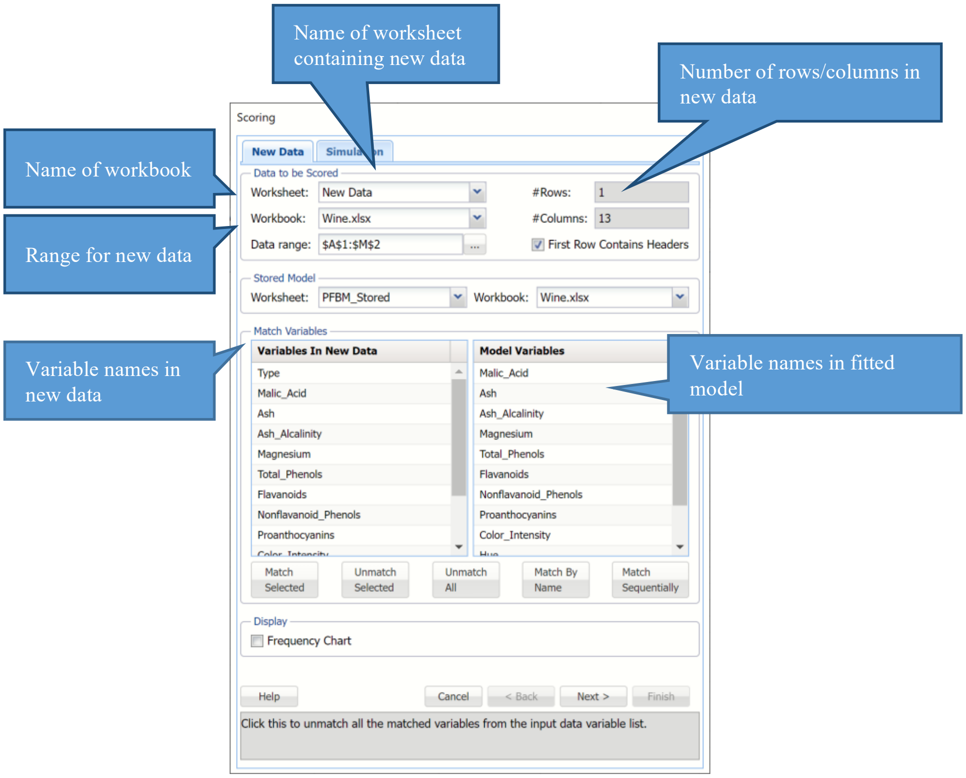
Scoring New Data

Now that the model has been fit to the data, this fitted model will be used to score new patient data, found below. Enter the following new data into a new tab in the workbook.



Type	Malic_Acid	Ash	Ash_Alcalinity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280_OD315	Proline
B	2.34	2.37	19.5	100	2.30	2.03	0.36	1.59	5.06	0.96	2.61	747

Click the New Data tab and then click the Score icon on the Analytic Solver Data Science ribbon.



Name of worksheet containing new data

Number of rows/columns in new data

Name of workbook

Range for new data

Variable names in new data

Variable names in fitted model

Scoring

Data to be Scored

Worksheet: New Data

Workbook: Wine.xlsx

Data range: $A$1:$M$2

#Rows: 1

#Columns: 13

☑ First Row Contains Headers

Stored Model

Worksheet: PFBM_Stored

Workbook: Wine.xlsx

Match Variables

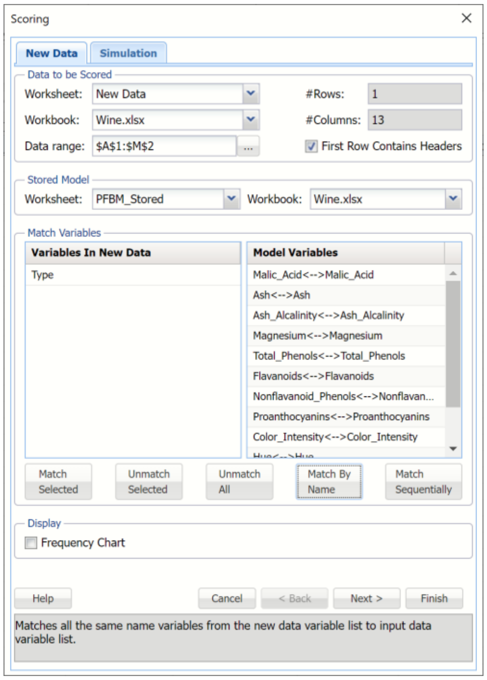
Variables In New Data	Model Variables
Type	Malic_Acid
Malic_Acid	Ash
Ash	Ash_Alcalinity
Ash_Alcalinity	Magnesium
Magnesium	Total_Phenols
Total_Phenols	Flavanoids
Flavanoids	Nonflavanoid_Phenols
Nonflavanoid_Phenols	Proanthocyanins
Proanthocyanins	Color_Intensity
Color_Intensity	Hue

Display

☐ Frequency Chart

Click this to unmatch all the matched variables from the input data variable list.

Click "Match By Name" to match each variable in the new data with the same variable in the fitted model, i.e. Malic_Acid with Malic_Acid, Ash with Ash, etc.



Scoring

Data to be Scored

Worksheet: New Data

Workbook: Wine.xlsx

Data range: $A$1:$M$2

#Rows: 1

#Columns: 13

☑ First Row Contains Headers

Stored Model

Worksheet: PFBM_Stored

Workbook: Wine.xlsx

Match Variables

Variables In New Data	Model Variables
Type	Malic_Acid<-->Malic_Acid
	Ash<-->Ash
	Ash_Alcalinity<-->Ash_Alcalinity
	Magnesium<-->Magnesium
	Total_Phenols<-->Total_Phenols
	Flavanoids<-->Flavanoids
	Nonflavanoid_Phenols<-->Nonflavan...
	Proanthocyanins<-->Proanthocyanins
	Color_Intensity<-->Color_Intensity
	Hue<-->Hue

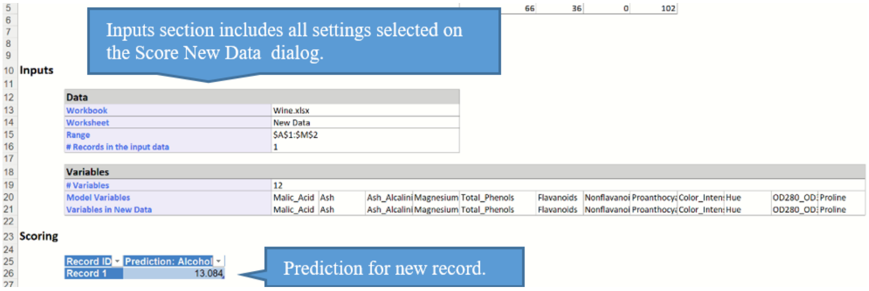
Display

☐ Frequency Chart

Matches all the same name variables from the new data variable list to input data variable list.

Click OK to score the new data record and predict the alcohol content of the new wine sample.

A new worksheet, Scoring_Bagging is inserted to the right.



Inputs section includes all settings selected on the Score New Data dialog.

Inputs

Data	
Workbook	Wine.xlsx
Worksheet	New Data
Range	$A$1:$M$2
# Records in the input data	1

Variables												
# Variables	12											
Model Variables	Malic_Acid	Ash	Ash_Alcalini	Magnesium	Total_Phenols	Flavanoids	Nonflavanoi	Proanthocya	Color_Inten	Hue	OD280_OD	Proline
Variables in New Data	Malic_Acid	Ash	Ash_Alcalini	Magnesium	Total_Phenols	Flavanoids	Nonflavanoi	Proanthocya	Color_Inten	Hue	OD280_OD	Proline

Scoring

Record ID	Prediction: Alcohol
Record 1	13.084

Prediction for new record.

Notice that the predicted alcohol content for this sample is 13.084.

Please see the "Scoring New Data" chapter within the Analytic Solver Data Science User Guide for information on scoring new data.

Classifying the Iris Dataset

Introduction

Analytic Solver Data Science's chart wizard allows you to visualize the contents of your dataset by allowing you to create up to 8 different types of charts quickly and easily: bar chart, histogram, line chart, parallel coordinates, scatterplot, scatterplot matrix, boxplot, and variable. Create a bar chart to compare individual statistics (i.e. mean, count, etc.) across a group of variables or a box plot to illustrate the shape of a distribution, it's central value and range of data. Use a histogram to depict the range and scale of your observations at variable intervals, a line chart to describe a time series dataset, or a parallel coordinates plot to create a multivariate profile. A scatterplot can be created to compare the relationship between two variables while a scatterplot matrix combines several scatterplots into one panel allowing the user to see pairwise relationships between variables. Finally, use the variable graph to plot each variable's distribution. Two additional options, Export to PowerBI and Export to Tableau, may be used to export your data to these 3rd party applications. For more information, see the Analytic Solver Data Science Reference Guide.

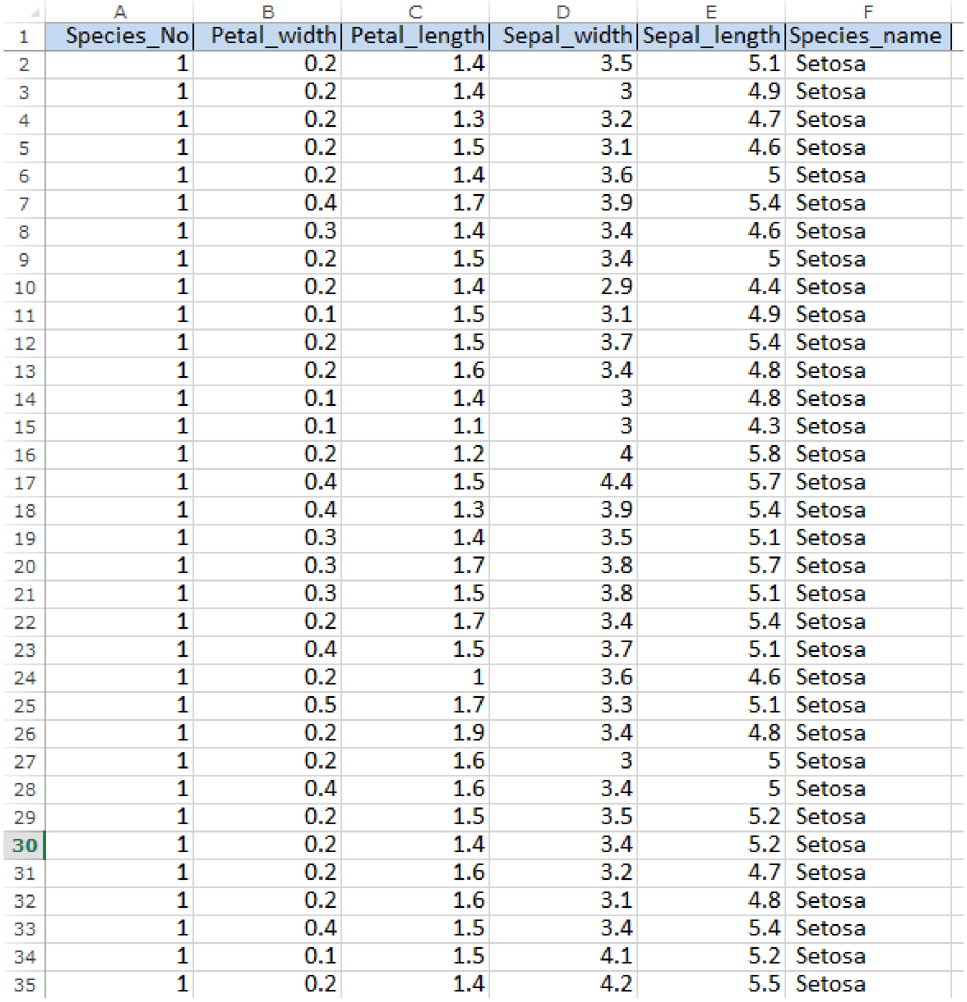
The following example will walk you through a quick tutorial, using both Analytic Solver Data Science Cloud and Desktop apps, which create several different types of charts to allow a quick and thorough visualization of the well-known iris dataset.

Creating the Classification Model

In this example, we will use the well-known iris dataset to craft a classification model. The iris dataset is a multivariate data set introduced by Sir Ronald Fisher in 1936 as an example of discriminant analysis. This data set contains 50 observations from each of three Iris species: Iris setosa, Iris virginica and Iris versicolor. Four characterisitics were recorded from each sample: sepal length and width and petal length and width. Our goal is to build a model that will assign new data points to the correct iris species.

To open this dataset, click **Help – Example Models – Forecasting/Data Science Examples**, then click the *Iris* link.

A portion of the dataset is shown below.

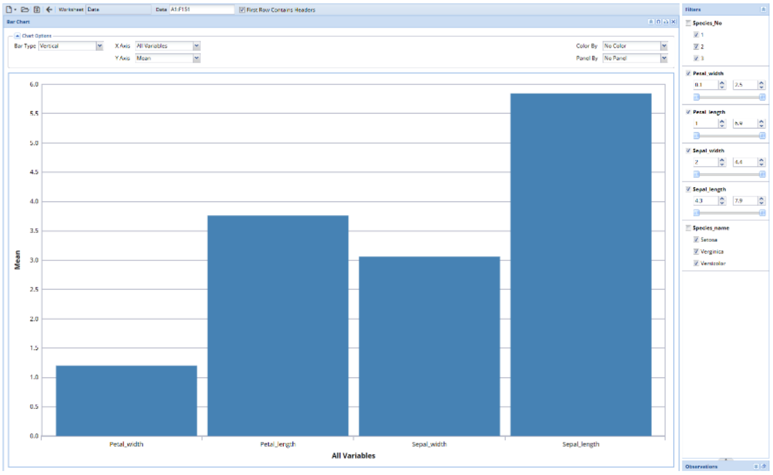


	A	B	C	D	E	F
1	Species_No	Petal_width	Petal_length	Sepal_width	Sepal_length	Species_name
2	1	0.2	1.4	3.5	5.1	Setosa
3	1	0.2	1.4	3	4.9	Setosa
4	1	0.2	1.3	3.2	4.7	Setosa
5	1	0.2	1.5	3.1	4.6	Setosa
6	1	0.2	1.4	3.6	5	Setosa
7	1	0.4	1.7	3.9	5.4	Setosa
8	1	0.3	1.4	3.4	4.6	Setosa
9	1	0.2	1.5	3.4	5	Setosa
10	1	0.2	1.4	2.9	4.4	Setosa
11	1	0.1	1.5	3.1	4.9	Setosa
12	1	0.2	1.5	3.7	5.4	Setosa
13	1	0.2	1.6	3.4	4.8	Setosa
14	1	0.1	1.4	3	4.8	Setosa
15	1	0.1	1.1	3	4.3	Setosa
16	1	0.2	1.2	4	5.8	Setosa
17	1	0.4	1.5	4.4	5.7	Setosa
18	1	0.4	1.3	3.9	5.4	Setosa
19	1	0.3	1.4	3.5	5.1	Setosa
20	1	0.3	1.7	3.8	5.7	Setosa
21	1	0.3	1.5	3.8	5.1	Setosa
22	1	0.2	1.7	3.4	5.4	Setosa
23	1	0.4	1.5	3.7	5.1	Setosa
24	1	0.2	1	3.6	4.6	Setosa
25	1	0.5	1.7	3.3	5.1	Setosa
26	1	0.2	1.9	3.4	4.8	Setosa
27	1	0.2	1.6	3	5	Setosa
28	1	0.4	1.6	3.4	5	Setosa
29	1	0.2	1.5	3.5	5.2	Setosa
30	1	0.2	1.4	3.4	5.2	Setosa
31	1	0.2	1.6	3.2	4.7	Setosa
32	1	0.2	1.6	3.1	4.8	Setosa
33	1	0.4	1.5	3.4	5.4	Setosa
34	1	0.1	1.5	4.1	5.2	Setosa
35	1	0.2	1.4	4.2	5.5	Setosa

First, we will explore our dataset with graphical analysis. This is often a beginning step when creating a classification or prediction model. Visualizations can help identify patterns within each variable or in relationships between variables. If a variable is numerical, charts such as histograms or boxplots may be used to study the distribution of values and to identify outliers. If the dataset is a time series, a line chart may be used to detect a trend in the data, for example, the time of year when ski/snowboard sales spike. Bar charts are typically used when a variable is categorical, such as the number of ski's, snowboards, and poles that are purchased during a winter sale. The iris dataset includes both categorical and numerical data. Let's start with a simple bar chart to examine the maximum, minimum and mean values of each numerical variable, petal_length, petal_width, sepal_length and sepal_width.

- To create a chart, select a cell within the dataset, say cell A2, then click **Explore – Chart Wizard** to open the Chart Wizard. Select **Bar Chart**.
- Uncheck Species_No and Species_name from the Filters pane to remove both from the chart.

The resultant chart displays the mean values for all variables.



Worksheet Data Data A1:F151 ☑ First Row Contains Headers

Bar Chart

Chart Options

Bar Type Vertical X Axis All Variables Color By No Color

Y Axis Mean Panel By No Panel

Filters

□ Species_No ☑ 1 ☑ 2 ☑ 3

☑ Petal_width 0.1 2.5

☑ Petal_length 1 6.9

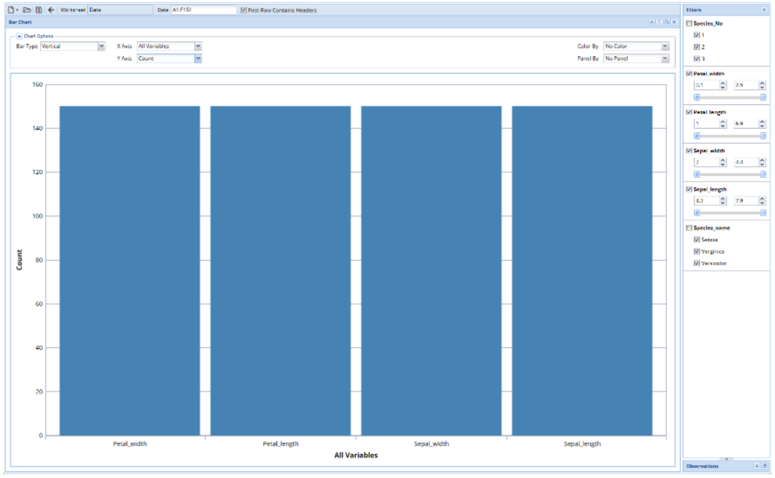
☑ Sepal_width 2 4.4

☑ Sepal_length 4.3 7.9

□ Species_name ☑ Setosa ☑ Verginica ☑ Versicolor

Observations

To confirm that all variables have the same data counts, select Count for the Y Axis.



Worksheet Data Data A1:F151 ☑ First Row Contains Headers

Bar Chart

Chart Options

Bar Type Vertical X Axis All Variables Color By No Color

Y Axis Count Panel By No Panel

Filters

□ Species_No ☑ 1 ☑ 2 ☑ 3

☑ Petal_width 0.1 2.5

☑ Petal_length 1 6.9

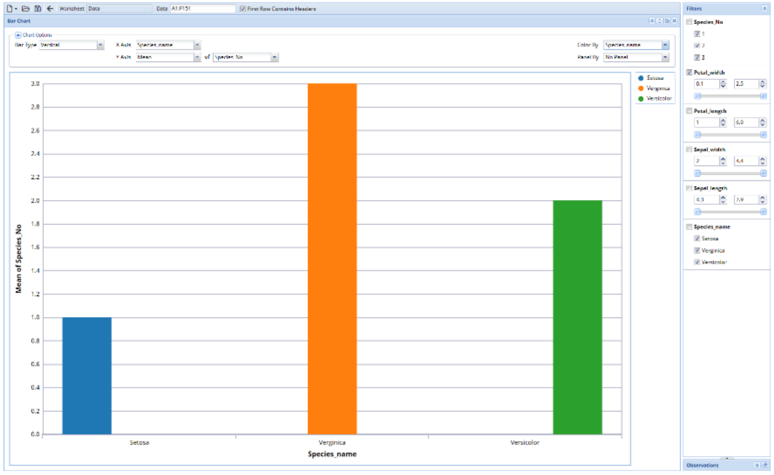
☑ Sepal_width 2 4.4

☑ Sepal_length 4.3 7.9

□ Species_name ☑ Setosa ☑ Verginica ☑ Versicolor

Observations

Select Species_Name for X Axis and Mean of Species_No for Y Axis. Then set Color By to Species_name. Then unselect all but Petal_width in the Filters pane.



Worksheet Data Data A1:F151 ☑ First Row Contains Headers

Bar Chart

Chart Options

Bar Type Vertical X Axis Species_name Color By Species_name

Y Axis Mean of Species_No Panel By No Panel

Filters

□ Species_No ☑ 1 ☑ 2 ☑ 3

☑ Petal_width 0.1 2.5

□ Petal_length 1 6.9

□ Sepal_width 2 4.4

□ Sepal_length 4.3 7.9

□ Species_name ☑ Setosa ☑ Verginica ☑ Versicolor

Observations

This chart becomes a bit more interesting. In this chart, we see that the Setosa species typcially has a small petal width, while the Verginica species has a very large petal_width, in relation to the Setosa species. The petal width of

Versicolor is about midway between the two. This chart supports the idea of including petal_width as a possible predictor in our classification model.

We could use similar steps to examine the mean values by species of the reamining numerical variables: petal_length, sepal_width and sepal_length.

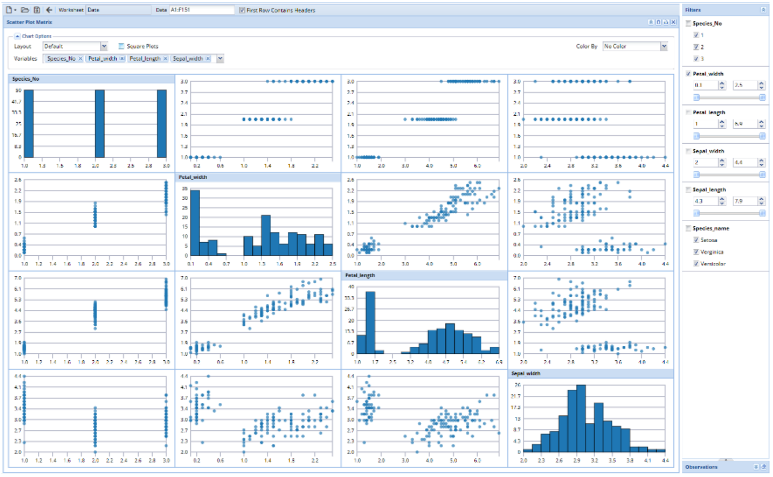
Now let's create a scatterplot matrix to identify any trends in the relationships *between* variables.

Click the back arrow. You will be asked to Save or Discard the chart. To save, enter a name such as Box Whisker Petal Length in the Name field, then click Save. To discard, click the trash can icon that corresponds to the chart name. To open a previously saved chart, click Explore – Existing Charts.

Select the Scatterpot Matrix.

Uncheck Square Plots at the top to extend the matrix to the width of the chart area.

On the diagonal of the Scatterplot Matrix, we find histograms of each variable. Each histogram displays the frequency of values for each variable.



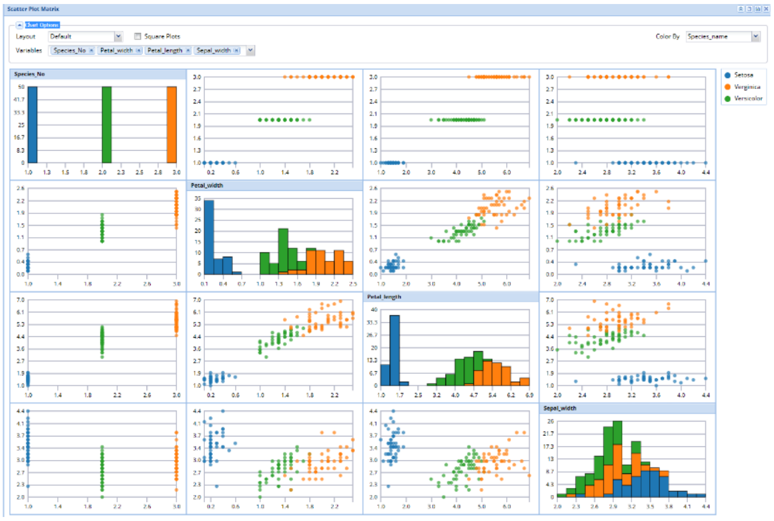
Worksheet Data Data A1:F151 ☑ First Row Contains Headers

Layout Default □ Square Plots Color By No Color

Variables Species_No × Petal_width × Petal_length × Sepal_width ×

Now we can clearly see that Setosa iris' have short, narrow petals and short, wide sepals. While the Verginica species has long wide petals and long, medium width sepals.

Set Color By to Species Name to distinguish each data point by species.



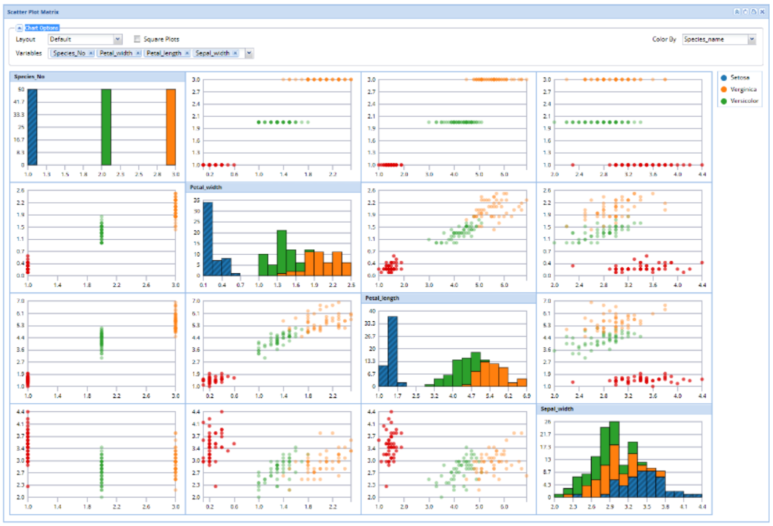
Layout Default □ Square Plots Color By Species_name

Variables Species_No × Petal_width × Petal_length × Sepal_width ×

Moving off to the right of the Petal_width histogram, we find a scatterplot matrix with Petal_width on the y axis and petal_length on the x axis. As you can see there is a distinct cluster of observations with narrow petal_widths and short petal_lengths.

Let’s take a closer look by using our mouse to draw a square around the cluster of points in the lower left of this scatterplot in each app.

Immediately, the observations that are in this cluster all turn red, if using the Desktop app, or blue, if using the Cloud app, in each histogram and scatterplot matrix. As you can see, not only do these observations have narrow petal widths and short petal lengths but they also have short sepal lengths with slightly wider sepal widths.



Scatter Plot Matrix

Chart Options

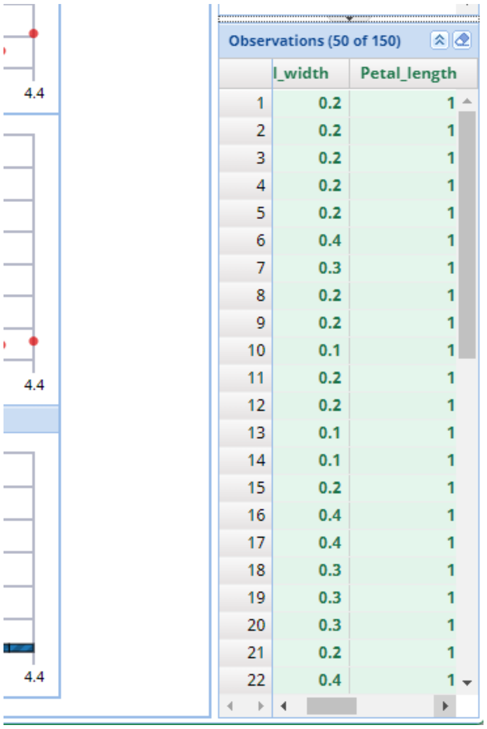
Layout: Default

□ Square Plots

Color By: Species_name

Variables: Species_No, Petal_width, Petal_length, Sepal_width

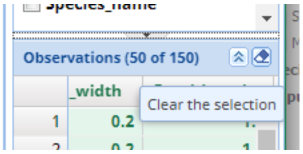
The records for each point in this cluster can be found under Observations on the right of the Chart Wizard. The first thing we see is that this cluster is made up of 50 points out of a total of 150. If we take a moment to scroll through these 50 records using the right and left arrows, one common feature starts to emerge – each of these records belongs to the Setosa species.



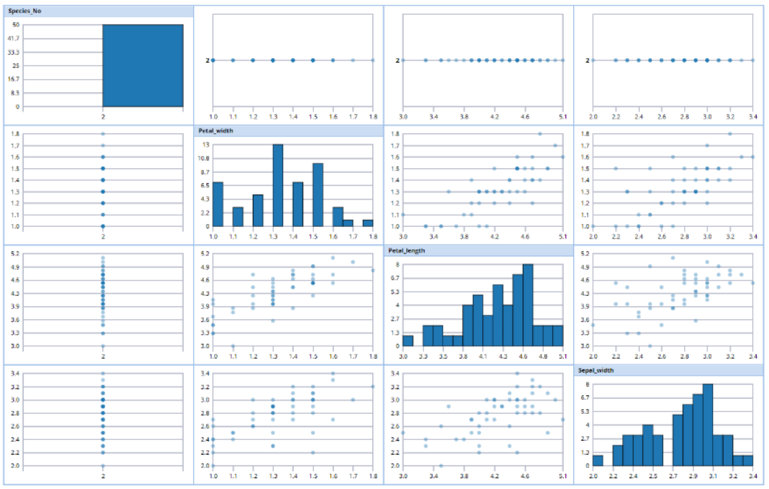
Observations (50 of 150)

	l_width	Petal_length
1	0.2	1
2	0.2	1
3	0.2	1
4	0.2	1
5	0.2	1
6	0.4	1
7	0.3	1
8	0.2	1
9	0.2	1
10	0.1	1
11	0.2	1
12	0.2	1
13	0.1	1
14	0.1	1
15	0.2	1
16	0.4	1
17	0.4	1
18	0.3	1
19	0.3	1
20	0.3	1
21	0.2	1
22	0.4	1

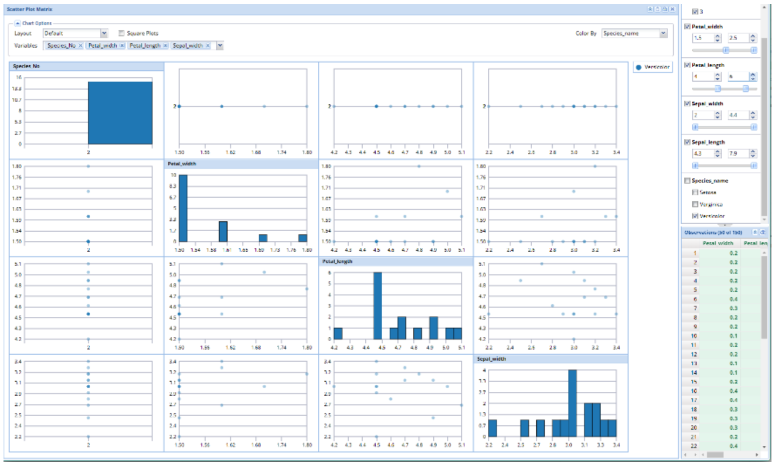
To remove the selection, click the eraser icon in the top, right of the Observations tab.



From here we can start to customize a little deeper in the Destop app. For example, we can look only at the Versicolor species simply by unchecking both Setosa and Verginica under Species_name.

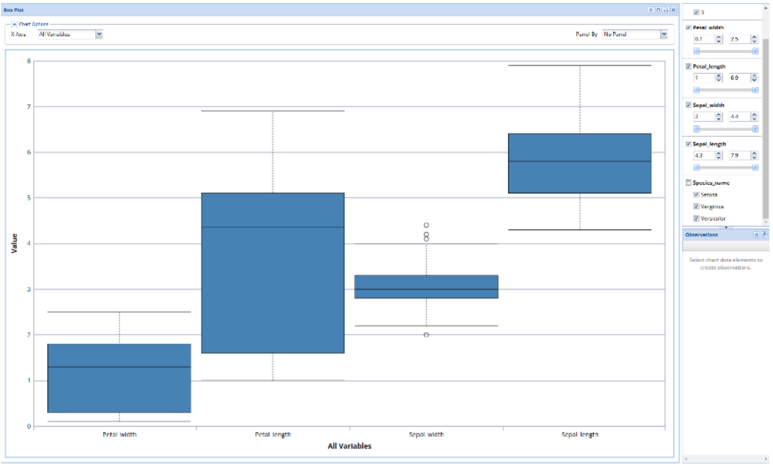


We can further filter our graph by only looking at observations where petal_width is between 1.5 and 2.5 and petal_length is between 4 and 6, by either moving the sliders left and right or using the spinner buttons, appropriately.



Add the two species back into the chart and then restore all the filters back to their defaults by simply moving all sliders to their extremes. Now let's create a boxplot, as an alternative to the histograms in the scatterplot matrix. A boxplot is similar to a histogram in that it graphically displays a variable's distribution of values, but a boxplot also includes several additional useful statistics. In a boxplot, also known as a box and whisker plot, the whiskers denote the minimum and maximum values and a "box" is used to designate the 25th and 75th percentiles. The top of the box denotes the 75th percentile and the bottom of the box denotes the 25th percentile. Inside the box is a solid line indicating the variable's median value. Looking at the three boxplots together, we can see that while petal_widths for Verginica and Versacolor species overlap, while the petal_widths of the Setosa iris do not.

Data Science Cloud



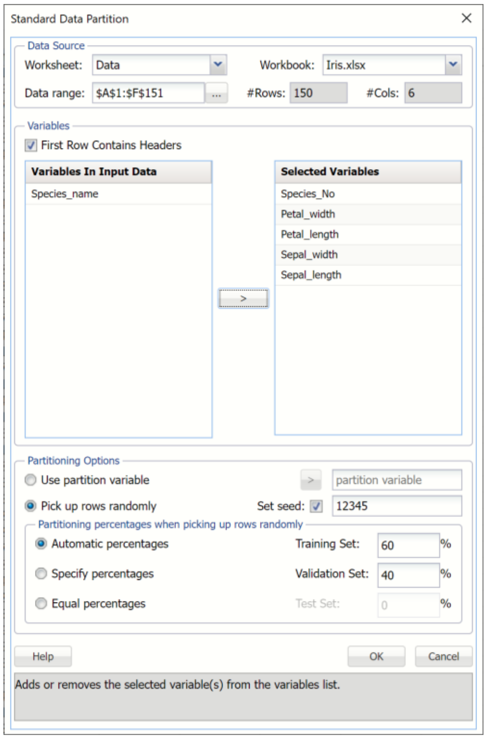
Click the X in the title of the Chart Wizard dialog to close the Chart Wizard. For more information on the chart wizard, including walk throughs on how to create each chart in the wizard, see the Data Science Reference Guide.

In the data science field, datasets with large amounts of variables are routinely encountered. In most cases, the size of the dataset can be reduced by removing highly correlated or superfluous variables. The accuracy and reliability of a classification or prediction model produced from this resultant dataset will be improved by the removal of these redundant and unnecessary variables. In addition, superfluous variables increase the data-collection and data-processing costs of deploying a model on a large database. As a result, one of the first steps in data science should be finding ways to reduce the number of independent or input variables used in the model (otherwise known as dimensionality) without sacrificing accuracy.

Dimensionality Reduction is the process of reducing the amount of variables to be used as input to a prediction or classification model. One way to reduce the number of dimensions in a classification model is to fit a model using a classification tree which splits on the variables that result in the best model. Variables that are not included in the classification tree, can be removed.

One very important issue when fitting a model is how well the newly created model will behave when applied to new data. To address this issue, the dataset is divided into multiple partitions: a training partition which is used to create or "train" the model and a validation partition to test the performance of the model. (For more information on partitioning, see the Analytic Solver Data Science Reference Guide.) For this particular example, we will partition the dataset according to the Standard Partition defaults of 60% of records assigned to the Training set and 40% of records assigned to the Validation set.

Click **Partition – Standard Partition** to open the Standard Data Partition dialog. Select Species_No, Petal_width, Petal_length, Sepal_width, and Sepal_length under Variables In Input data and click > to select and include in the partition. Click OK to accept the defaults and create the partitions.



Standard Data Partition ×

Data Source

Worksheet: Data   Workbook: Iris.xlsx

Data range: $A$1:$F$151 ...   #Rows: 150   #Cols: 6

Variables

☑ First Row Contains Headers

Variables In Input Data	Selected Variables
Species_name	Species_No
	Petal_width
	Petal_length
	Sepal_width
	Sepal_length

Partitioning Options

○ Use partition variable   >   partition variable

● Pick up rows randomly   Set seed: ☑ 12345

Partitioning percentages when picking up rows randomly

● Automatic percentages   Training Set: 60 %

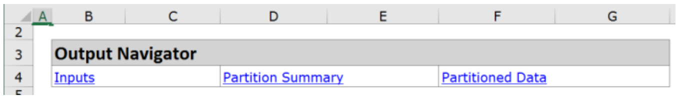
○ Specify percentages   Validation Set: 40 %

○ Equal percentages   Test Set: 0 %

Help   OK   Cancel

Adds or removes the selected variable(s) from the variables list.

STDPartition is inserted in the Model tab of the Analytic Solver task pane under Data Science – Results – Text Mining. Click the Partition Summary link in the Output Navigator to highlight the records selected for the Training Dataset and click the Validation Data link to highlight the records selected for the Validation Dataset.

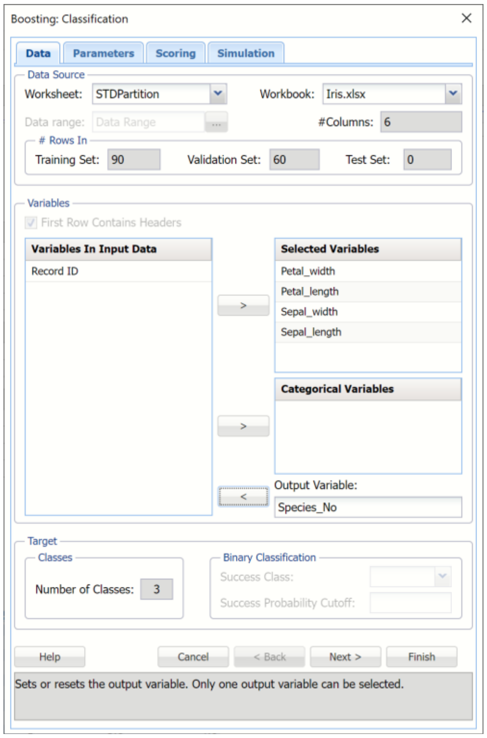


Output Navigator

Inputs	Partition Summary	Partitioned Data

Analytic Solver Data Science offers three powerful ensemble methods for use with Classification trees: bagging (bootstrap aggregating), boosting, and random trees. The Classification Tree Algorithm on its own can be used to find one model that will result in a good classification of the new data. We can view the statistics and confusion matrices of the current classifier to see if our model is a good fit to the data, but how would we know if there is a better classifier? The answer is – we don't. Ensemble methods, however, allow us to combine multiple "weak" classification tree models which, when taken together form a new, more accurate "strong" classification tree model. These methods work by creating multiple diverse classification models, by taking different samples of the original dataset, and then combining their outputs. Outputs may be combined by several techniques, for example, majority vote for classification and averaging for regression. This combination of models effectively reduces the variance in the "strong" model. The three different types of ensemble methods offered in Analytic Solver (bagging, boosting, and random trees) differ on three items: 1.The selection of training data for each classifier or "weak" model, 2.How the "weak" models are generated and 3. How the outputs are combined. In all three methods, each "weak" model is trained on the entire training dataset to become proficient in some portion of the dataset.

Click Classify – Ensemble – Boosting to open the Boosting – Data tab. Select Petal_width, Petal_length, Sepal_width, and Sepal_length as Selected Variables and Species_No as the Output Variable.



Boosting: Classification

Data Source

Worksheet: STDPartition

Workbook: Iris.xlsx

Data range: Data Range

#Columns: 6

Rows In

Training Set: 90

Validation Set: 60

Test Set: 0

Variables

☑ First Row Contains Headers

Variables In Input Data
Record ID

Selected Variables
Petal_width
Petal_length
Sepal_width
Sepal_length

Categorical Variables

Output Variable: Species_No

Target

Classes

Number of Classes: 3

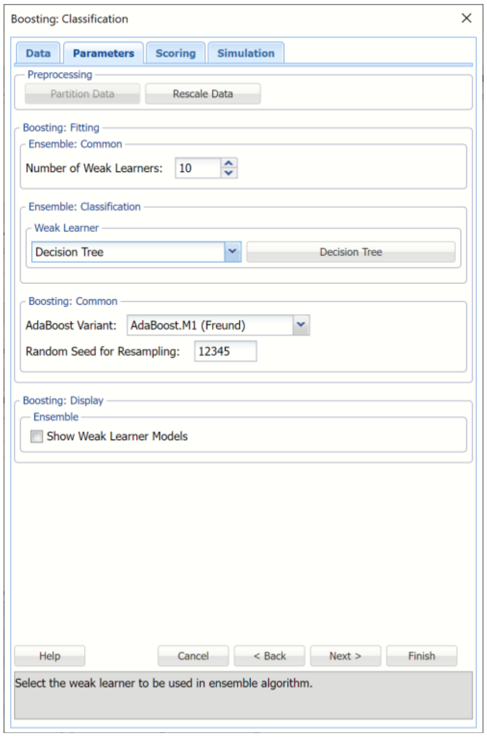
Binary Classification

Success Class:

Success Probability Cutoff:

Sets or resets the output variable. Only one output variable can be selected.

Click Next to advance to the Classification Tree Boosting – Parameters tab. Click the down arrow beneath Weak Learner and select Decision Tree from the menu. Note: If we had not previously partitioned the dataset, we could have performed the partitioning from this dialog by clicking Partition Data.



Number of Weak Learners: 10

Weak Learner: Decision Tree

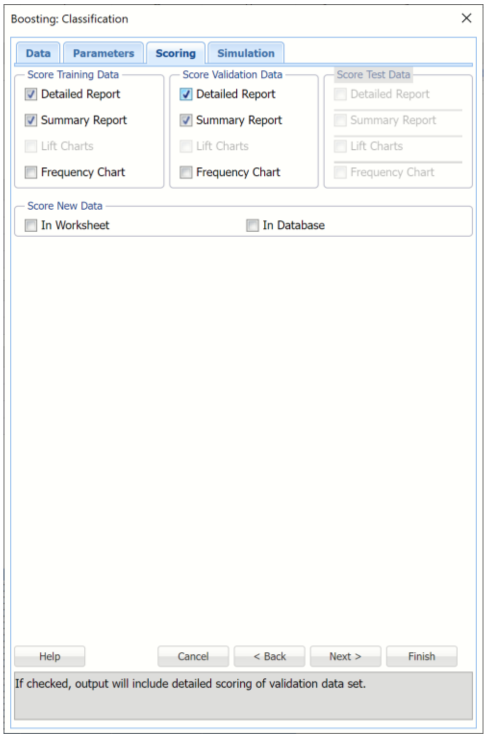
AdaBoost Variant: AdaBoost.M1 (Freund)

Random Seed for Resampling: 12345

□ Show Weak Learner Models

Select the weak learner to be used in ensemble algorithm.

Click Next to accept all default option settings and advance to the Classification Tree Boosting – Scoring tab. Select Detailed Report under both Scoring Training Data and Score Validation Data.



Score Training Data

☑ Detailed Report

☑ Summary Report

□ Lift Charts

□ Frequency Chart

Score Validation Data

☑ Detailed Report

☑ Summary Report

□ Lift Charts

□ Frequency Chart

Score Test Data

□ Detailed Report

□ Summary Report

□ Lift Charts

□ Frequency Chart

Score New Data

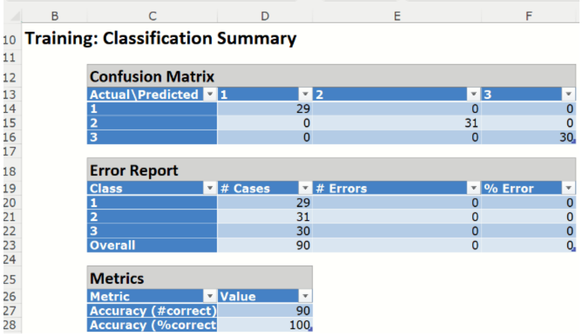
□ In Worksheet □ In Database

If checked, output will include detailed scoring of validation data set.

For more information on the Simulation tab and Frequency Chart on the Simulation tab, see the Finding Best Model or the Predicting Housing Prices chapters.

Click Finish to accept all default option settings and run the Boosting algorithm. Output sheets containing results of the algorithm will be inserted to the right of the Data worksheet. Click the Training: Classification Summary and Validation: Classification Summary links in the Output Navigator.

The algorithm performed perfectly on the training data partition as evidenced by the overall error of 0.



Training: Classification Summary

Confusion Matrix

Actual\Predicted	1	2	3
1	29	0	0
2	0	31	0
3	0	0	30

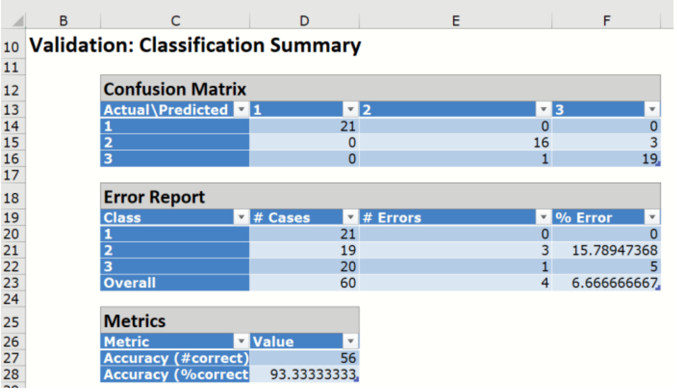
Error Report

Class	# Cases	# Errors	% Error
1	29	0	0
2	31	0	0
3	30	0	0
Overall	90	0	0

Metrics

Metric	Value
Accuracy (#correct)	90
Accuracy (%correct	100

In the validiation set, four records were assigned incorrectly resulting in a 6.67% misclassification error.



Validation: Classification Summary

Confusion Matrix

Actual\Predicted	1	2	3
1	21	0	0
2	0	16	3
3	0	1	19

Error Report

Class	# Cases	# Errors	% Error
1	21	0	0
2	19	3	15.78947368
3	20	1	5
Overall	60	4	6.666666667

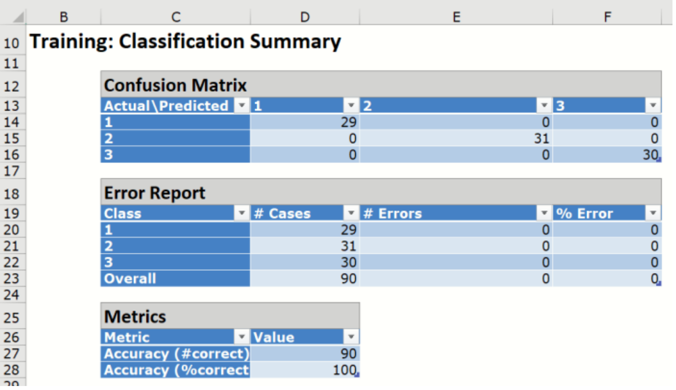
Metrics

Metric	Value
Accuracy (#correct)	56
Accuracy (%correct	93.33333333

If we repeat the same steps using the two remaining Ensemble methods, Bagging (using Decision Trees as the Weak Leaner) and Random Trees, we will get the following results.

Each ensemble method performed well, by evidenced by the overall errors in the validation sets, but the random trees method performed best.

Bagging



Training: Classification Summary

Confusion Matrix

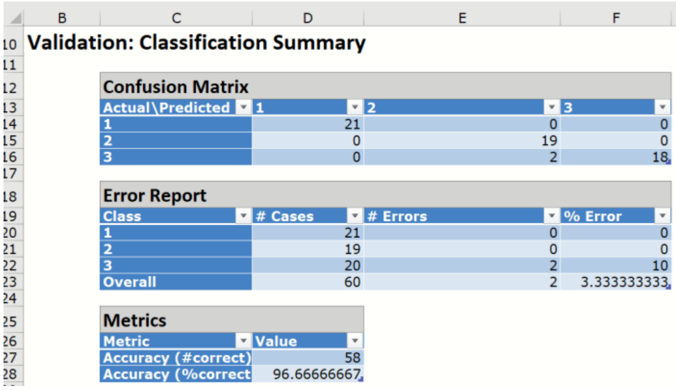
Actual\Predicted	1	2	3
1	29	0	0
2	0	31	0
3	0	0	30

Error Report

Class	# Cases	# Errors	% Error
1	29	0	0
2	31	0	0
3	30	0	0
Overall	90	0	0

Metrics

Metric	Value
Accuracy (#correct)	90
Accuracy (%correct	100



Validation: Classification Summary

Confusion Matrix

Actual\Predicted	1	2	3
1	21	0	0
2	0	19	0
3	0	2	18

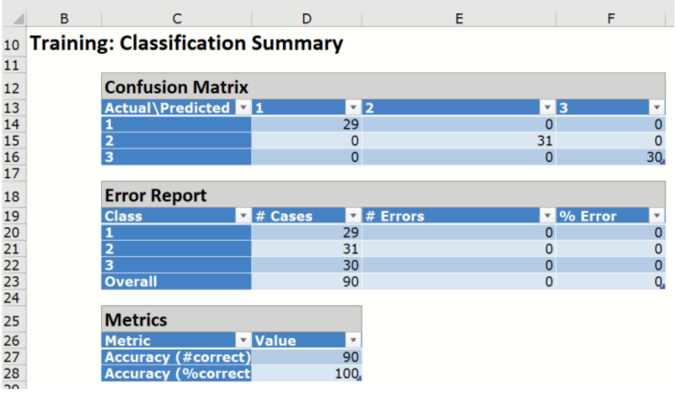
Error Report

Class	# Cases	# Errors	% Error
1	21	0	0
2	19	0	0
3	20	2	10
Overall	60	2	3.333333333

Metrics

Metric	Value
Accuracy (#correct)	58
Accuracy (%correct	96.66666667

Random Trees



Training: Classification Summary

Confusion Matrix

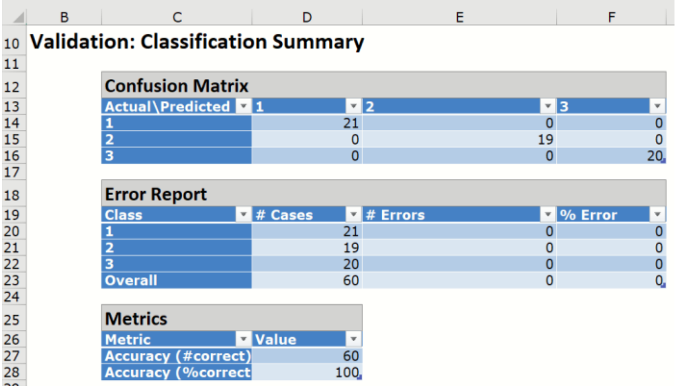
Actual\Predicted	1	2	3
1	29	0	0
2	0	31	0
3	0	0	30

Error Report

Class	# Cases	# Errors	% Error
1	29	0	0
2	31	0	0
3	30	0	0
Overall	90	0	0

Metrics

Metric	Value
Accuracy (#correct)	90
Accuracy (%correct	100



Validation: Classification Summary

Confusion Matrix

Actual\Predicted	1	2	3
1	21	0	0
2	0	19	0
3	0	0	20

Error Report

Class	# Cases	# Errors	% Error
1	21	0	0
2	19	0	0
3	20	0	0
Overall	60	0	0

Metrics

Metric	Value
Accuracy (#correct)	60
Accuracy (%correct	100

Predicting Housing Prices using Multiple Linear Regression

Introduction

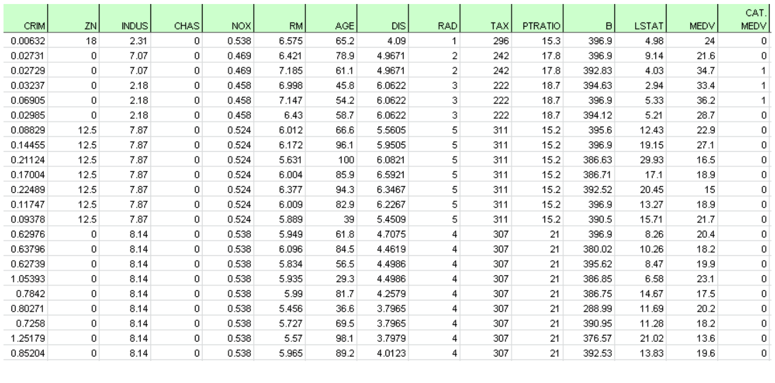
Prediction algorithms are supervised learning methods which aim to estimate, or forecast, a continuous output variable. For example, the predicted price of a house that has just come on the market. Analytic Solver Data Science includes 4 different prediction algorithms: Multiple Linear Regression, k-Nearest Neighbors, Regression Tree, and Neural Networks.

This example uses Multiple Linear Regression to fit a prediction model using the Boston_Housing dataset. The information in this dataset was gathered by the US Census Service from census tracts within the Boston area. Each of the 14 features (or variables) describes a characteristic impacting the selling price of a house. (A description of each variable is given in the example workbook.) In addition to these variables, the data set also contains an additional variable, which has been created by categorizing median value (MEDV) into two categories – high (MEDV > 30) and low (MEDV < 30).

Multiple Linear Regression Example

Input

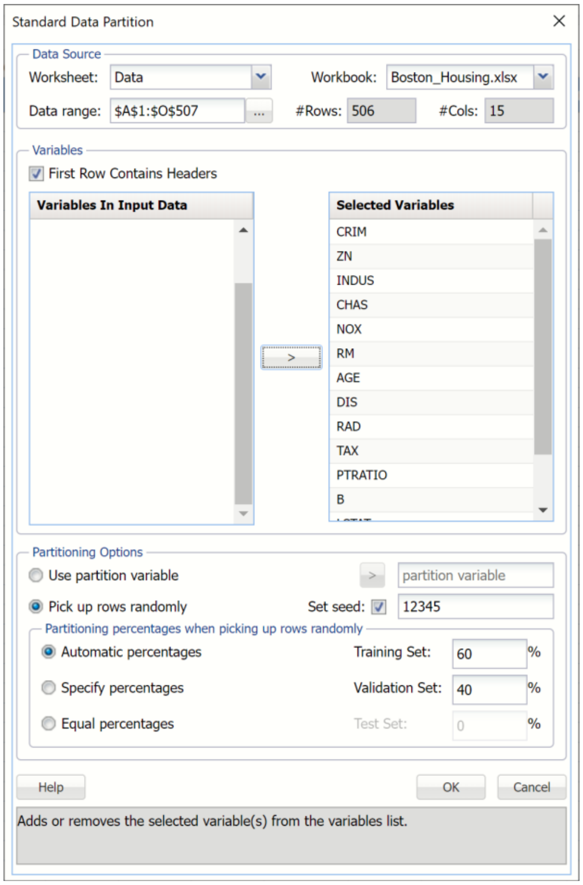
1. Open or upload the Boston_Housing.xlsx example dataset. A portion of the dataset is shown below. The last variable, CAT.MEDV, is a discrete classification of the MEDV variable and will not be used in this example.



CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV	CAT. MEDV
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24	0
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6	0
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7	1
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4	1
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2	1
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7	0
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9	0
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1	0
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5	0
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9	0
0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15	0
0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9	0
0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7	0
0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4	0
0.63796	0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21	380.02	10.26	18.2	0
0.62739	0	8.14	0	0.538	5.834	56.5	4.4986	4	307	21	395.62	8.47	19.9	0
1.05393	0	8.14	0	0.538	5.935	29.3	4.4986	4	307	21	386.85	6.58	23.1	0
0.7842	0	8.14	0	0.538	5.99	81.7	4.2579	4	307	21	386.75	14.67	17.5	0
0.80271	0	8.14	0	0.538	5.456	36.6	3.7965	4	307	21	288.99	11.69	20.2	0
0.7258	0	8.14	0	0.538	5.727	69.5	3.7965	4	307	21	390.95	11.28	18.2	0
1.25179	0	8.14	0	0.538	5.57	98.1	3.7979	4	307	21	376.57	21.02	13.6	0
0.85204	0	8.14	0	0.538	5.965	89.2	4.0123	4	307	21	392.53	13.83	19.6	0

2. First, we partition the data into training and validation sets using the Standard Data Partition defaults with percentages of 60% of the data randomly allocated to the Training Set and 40% of the data randomly allocated to the Validation Set. For more information on partitioning a dataset, see the *Data Science Partitioning* chapter within the Analytic Solver Data Science Reference Guide.

Standard Data Partition Dialog



Standard Data Partition

Data Source

Worksheet: Data

Workbook: Boston_Housing.xlsx

Data range: $A$1:$O$507

#Rows: 506

#Cols: 15

Variables

☑ First Row Contains Headers

Variables In Input Data	Selected Variables
	CRIM
	ZN
	INDUS
	CHAS
	NOX
	RM
	AGE
	DIS
	RAD
	TAX
	PTRATIO
	B

Partitioning Options

○ Use partition variable

● Pick up rows randomly

Set seed: ☑ 12345

Partitioning percentages when picking up rows randomly

● Automatic percentages

○ Specify percentages

○ Equal percentages

Training Set: 60 %

Validation Set: 40 %

Test Set: 0 %

Adds or removes the selected variable(s) from the variables list.

3. With the STDPartition worksheet selected, click **Predict – Linear Regression** to open the Linear Regression Data tab.

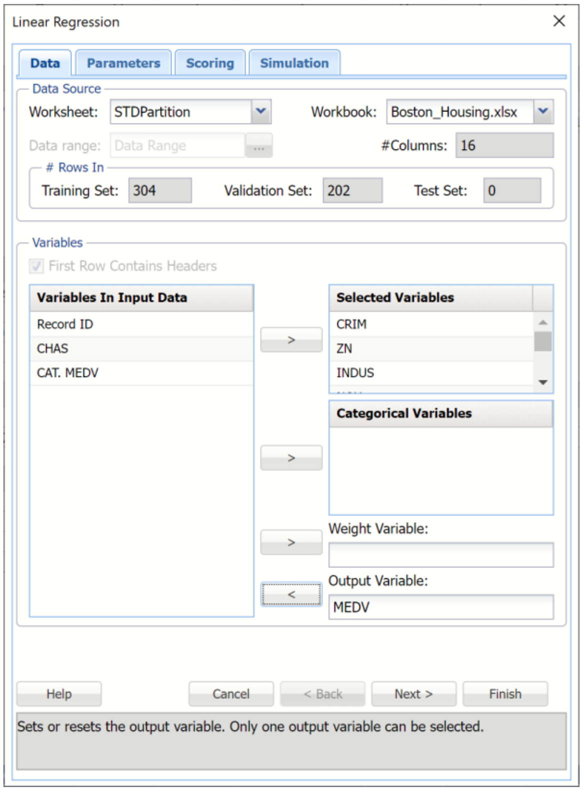
4. Select **MEDV** as the *Output Variable,* **CHAS** as a *Categorical Variable* and all remaining variables (except CAT. MEDV) as *Selected variables.*

Note:

All supervised algorithms include a new Simulation tab. This tab uses the functionality from the Generate Data feature (described in the What's New section of this guide and then more in depth in the Analytic Solver Data Science Reference Guide) to generate synthetic data based on the training partition, and uses the fitted model to produce predictions for the synthetic data. The resulting report, LinReq_Simulation, will contain the synthetic data, the predicted values and the Excel-calculated Expression column, if presentSince this new functionality does not support categorical variables, these types of variables will not be present in the model, only continuous variables.

If the new Simulation feature is not invoked, the categorical variable, CHAS may be added to the model as a categorical variable. However, in this specific instance, since the CHAS variable is the only Categorical Variable selected and it's values are binary (0/1), the output results will be the same as if you selected this variable as a Selected (or scale) Variable. If either 1. there were more categorical variables, 2. the CHAS variable was non-binary, or 3, another algorithm besides Mulitple Linear Regression or Logistic Regression was selected, this would not be true.

Linear Regression Dialog, Data tab



Data Source

Worksheet: STDPartition

Workbook: Boston_Housing.xlsx

Data range: Data Range

#Columns: 16

Rows In

Training Set: 304

Validation Set: 202

Test Set: 0

Variables

☑ First Row Contains Headers

Variables In Input Data
Record ID
CHAS
CAT. MEDV

Selected Variables
CRIM
ZN
INDUS

Categorical Variables

Weight Variable:

Output Variable: MEDV

Sets or resets the output variable. Only one output variable can be selected.

5. Click **Next** to advance to the *Parameters* tab.

6. If *Fit Intercept* is selected, the intercept term will be fitted, otherwise there will be no constant term in the equation. Leave this option selected for this example.

7. Under *Regression: Display*, select all 6 display options to display each in the output.

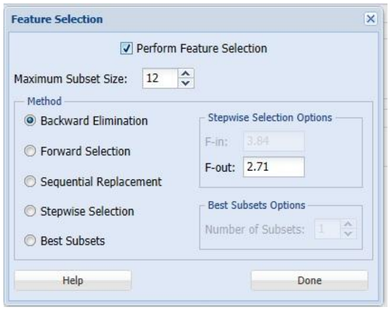
   Under Statistics, select:

   - ANOVA
   - Variance-Covariance Matrix
   - Multicollinearity Diagnostics

   Under Advanced, select:

   - Analysis of Coefficients
   - Analysis of Residuals
   - Influence Diagnostics
   - Confidence/Prediction Intervals

8. When you have a large number of predictors and you would like to limit the model to only the significant variables, click **Feature Selection** to open the *Feature Selection* dialog and select **Perform Feature Selection** at the top of the dialog. The default setting for *Maximum Subset Size* is **12**. This option can take on values of 1 up to N where N is the number of Selected Variables. The default setting is N.



Feature Selection

☑ Perform Feature Selection

Maximum Subset Size: 12

Method

● Backward Elimination

○ Forward Selection

○ Sequential Replacement

○ Stepwise Selection

○ Best Subsets

Stepwise Selection Options

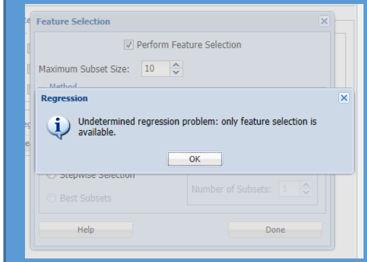
F-in: 3.84

F-out: 2.71

Best Subsets Options

Number of Subsets: 1

If the number of rows in the data is less than the number of variables selected as Input variables, the following message box is displayed. Select OK to be directed back to the Feature Selection dialog.



Feature Selection

☑ Perform Feature Selection

Maximum Subset Size: 10

Regression

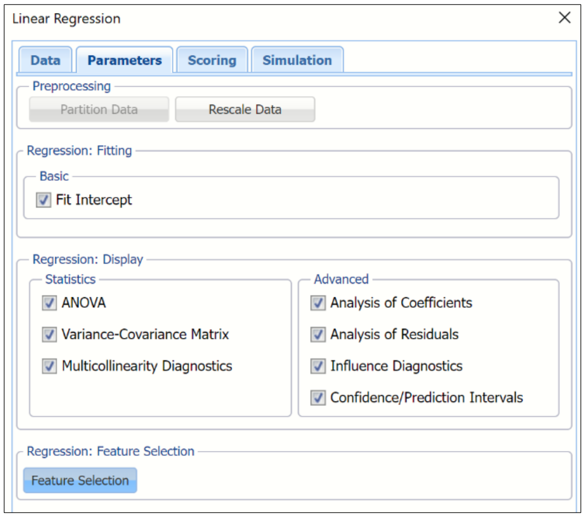
Undetermined regression problem: only feature selection is available.

Analytic Solver Data Science offers five different selection procedures for selecting the best subset of variables.

- *Backward Elimination* in which variables are eliminated one at a time, starting with the least significant. If this procedure is selected, FOUT is enabled. A statistic is calculated when variables are eliminated. For a variable to leave the regression, the statistic's value must be less than the value of FOUT (default = 2.71).
- *Forward Selection* in which variables are added one at a time, starting with the most significant. If this procedure is selected, FIN is enabled. On each iteration of the Forward Selection procedure, each variable is examined for the eligibility to enter the model. The significance of variables is measured as a partial F-statistic. Given a model at a current iteration, we perform an F Test, testing the null hypothesis stating that the regression coefficient would be zero if added to the existing set if variables and an alternative hypothesis stating otherwise. Each variable is examined to find the one with the largest partial F-Statistic. The decision rule for adding this variable into a model is: Reject the null hypothesis if the F-Statistic for this variable exceeds the critical value chosen as a threshold for the F Test (FIN value), or Accept the null hypothesis if the F-Statistic for this variable is less than a threshold. If the null hypothesis is rejected, the variable is added to the model and selection continues in the same fashion, otherwise the procedure is terminated.
- *Sequential Replacement* in which variables are sequentially replaced and replacements that improve performance are retained.
- *Stepwise selection* is similar to Forward selection except that at each stage, Analytic Solver Data Science considers dropping variables that are not statistically significant. When this procedure is selected, the Stepwise selection options FIN and FOUT are enabled. In the stepwise selection procedure a statistic is calculated when variables are added or eliminated. For a variable to come into the regression, the statistic's value must be greater than the value for FIN (default = 3.84). For a variable to leave the regression, the statistic's value must be less than the value of FOUT (default = 2.71). The value for FIN must be greater than the value for FOUT.
- *Best Subsets* where searches of all combinations of variables are performed to observe which combination has the best fit. (This option can become quite time consuming depending on the number of input variables.) If this procedure is selected, Number of best subsets is enabled.

9. Click **Done** to accept the default choice, Backward Elimimination with a Maximum Subset Size of 3 and an F-out setting of 2.71, and return to the Parameters tab, then click **Next** to advance to the *Scoring* tab.

Linear Regression Dialog, Parameters tab



Regression: Fitting

Basic

☑ Fit Intercept

Regression: Display

Statistics

☑ ANOVA

☑ Variance-Covariance Matrix

☑ Multicollinearity Diagnostics

Advanced

☑ Analysis of Coefficients

☑ Analysis of Residuals

☑ Influence Diagnostics

☑ Confidence/Prediction Intervals

Regression: Feature Selection

10. Click Next to proceed to the Scoring tab.

11. Select all four options for **Score Training/Validation data**.

When *Detailed report* is selected, Analytic Solver Data Science will create a detailed report of the Discriminant Analysis output.

When *Summary report* is selected (the default), Analytic Solver Data Science will create a report summarizing the Discriminant Analysis output.

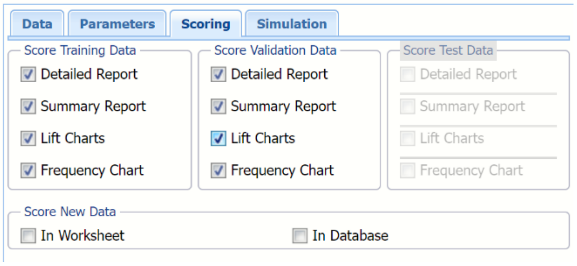
When *Lift Charts* is selected, Analytic Solver Data Science will include Lift Chart and ROC Curve plots in the output.

When Frequency Chart is selected, a frequency chart will be displayed when the LinReg_TrainingScore and LinReg_ValidationScore worksheets are selected. This chart will display an interactive application similar to the Analyze Data feature, explained in detail in the Analyze Data chapter that appears earlier in this guide. This chart will include frequency distributions of the actual and predicted responses individually, or side-by-side, depending on the user's preference, as well as basic and advanced statistics for variables, percentiles, six sigma indices.

Since we did not create a test partition, the options for Score test data are disabled. See the chapter "Data Science Partitioning" for information on how to create a test partition.

See the *Scoring New Data* chapter within the Analytic Solver Data Science User Guide for more information on *Score New Data in* options.

LinReg dialog, Scoring tab



Score Training Data: ☑ Detailed Report ☑ Summary Report ☑ Lift Charts ☑ Frequency Chart

Score Validation Data: ☑ Detailed Report ☑ Summary Report ☑ Lift Charts ☑ Frequency Chart

Score Test Data: □ Detailed Report □ Summary Report □ Lift Charts □ Frequency Chart

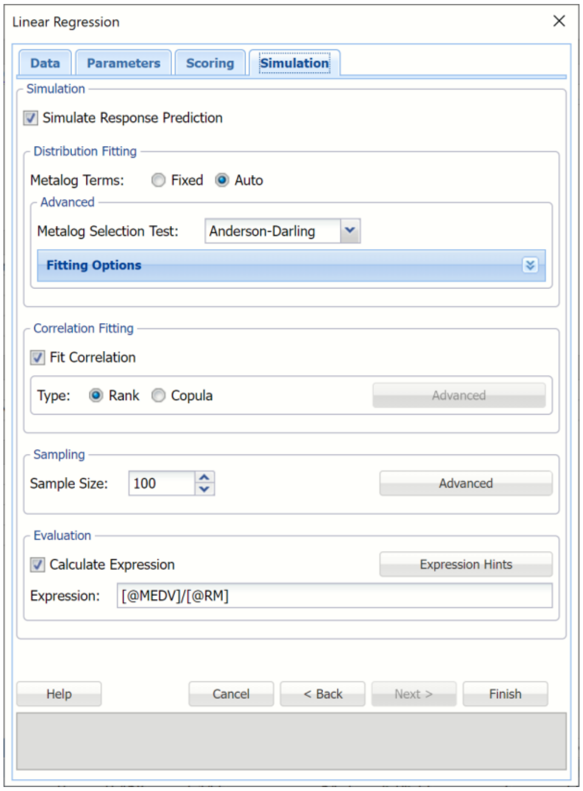
Score New Data: □ In Worksheet □ In Database

12. Click **Next** to advance to the Simulation tab.

13. Select Simulation Response Prediction to enable all options on the Simulation tab of the Linear Regression dialog.

    Simulation tab: All supervised algorithms include a new Simulation tab. This tab uses the functionality from the Generate Data feature (described earlier in this guide) to generate synthetic data based on the training partition, and uses the fitted model to produce predictions for the synthetic data. The resulting report, LinReg_Simulation, will contain the synthetic data, the predicted values and the Excel-calculated Expression column, if present. In addition, frequency charts containing the Predicted, Training, and Expression (if present) sources or a combination of any pair may be viewed, if the charts are of the same type.

    Linear Regression dialog, Simulation tab



Simulation: ☑ Simulate Response Prediction

Distribution Fitting: Metalog Terms: ○ Fixed ● Auto

Advanced: Metalog Selection Test: Anderson-Darling

Correlation Fitting: ☑ Fit Correlation

Type: ● Rank ○ Copula

Sampling: Sample Size: 100

Evaluation: ☑ Calculate Expression

Expression: [@MEDV]/[@RM]

Evaluation: Select Calculate Expression to amend an Expression column onto the frequency chart displayed on the LinReg_Simulation output sheet. Expression can be any valid Excel formula that references a variable and the response as [@COLUMN_NAME]. Click the *Expression Hints* button for more information on entering an expression.

For the purposes of this example, leave all options at their defaults in the Distribution Fitting, Correlation Fitting and Sampling sections of the dialog. For Expression, enter the following formula to display the price per room of each record.

Note that variable names are case sensitive.

For more information on the remaining options shown on this dialog in the Distribution Fitting, Correlation Fitting and Sampling sections, see the Generate Data chapter that appears earlier in this guide.

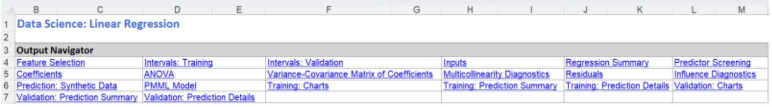
14. Click Finish to run Discriminant Analysis on the example dataset.

Output

This result worksheet includes 8 segments: Output Navigator, Inputs, Regression Summary, Predictor Screening, Coefficients, ANOVA, Variance-Covariance Matrix of Coefficients and Multicollinearity Diagnostics.

- **Output Navigator:** The Output Navigator appears at the top of all result worksheets. Use this feature to quickly navigate to all reports included in the output.

  LinReg_Output: Output Navigator

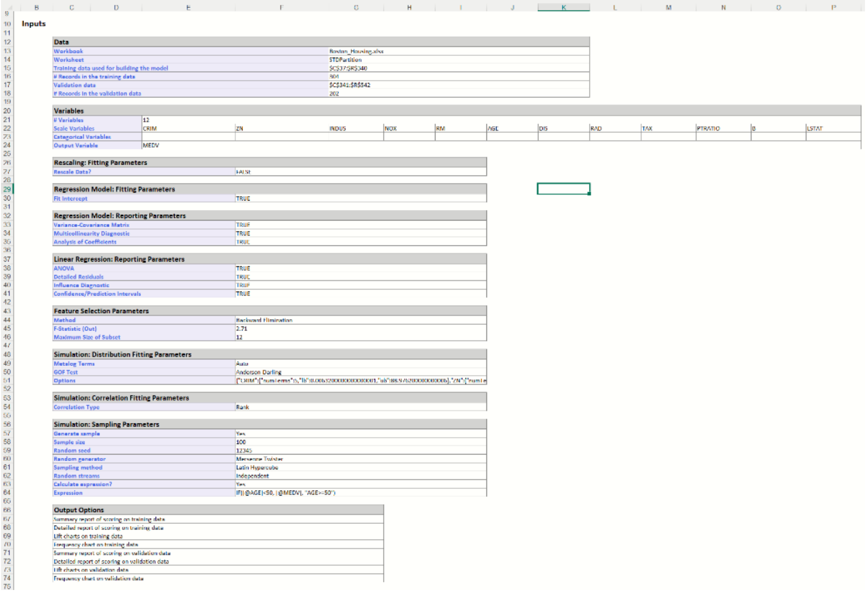


Data Science: Linear Regression

Output Navigator

Feature Selection	Intervals: Training	Intervals: Validation	Inputs	Regression Summary	Predictor Screening
Coefficients	ANOVA	Variance-Covariance Matrix of Coefficients	Multicollinearity Diagnostics	Residuals	Influence Diagnostics
Prediction: Synthetic Data	PMML Model	Training: Charts	Training: Prediction Summary	Training: Prediction Details	Validation: Charts
Validation: Prediction Summary	Validation: Prediction Details				

- **Inputs:** Scroll down to the Inputs section to find all inputs entered or selected on all tabs of the Discriminant Analysis dialog.



Inputs

Data	
Workbook	Boston_Housing.xlsx
Worksheet	STDPartition
Training data used for building the model	$C$37:$R$340
# Records in the training data	304
Validation data	$C$341:$R$542
# Records in the validation data	202

Variables												
# Variables	12											
Scale Variables	CRIM	ZN	INDUS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
Categorical Variables												
Output Variable	MEDV											

Rescaling: Fitting Parameters	
Rescale Data?	FALSE

Regression Model: Fitting Parameters	
Fit Intercept	TRUE

Regression Model: Reporting Parameters	
Variance-Covariance Matrix	TRUE
Multicollinearity Diagnostic	TRUE
Analysis of Coefficients	TRUE

Linear Regression: Reporting Parameters	
ANOVA	TRUE
Detailed Residuals	TRUE
Influence Diagnostic	TRUE
Confidence/Prediction Intervals	TRUE

Feature Selection Parameters	
Method	Backward Elimination
F-Statistic (Out)	2.71
Maximum Size of Subset	12

Simulation: Distribution Fitting Parameters	
Metalog Terms	Auto
GOF Test	Anderson Darling
Options	{"CRIM":{"numTerms":5,"lb":0.00632000000000001,"ub":88.9762000000006},"ZN":{"numTe

Simulation: Correlation Fitting Parameters	
Correlation Type	Rank

Simulation: Sampling Parameters	
Generate sample	Yes
Sample size	100
Random seed	12345
Random generator	Mersenne Twister
Sampling method	Latin Hypercube
Random streams	Independent
Calculate expression?	Yes
Expression	IF([@AGE]<50, [@MEDV], "AGE>=50")

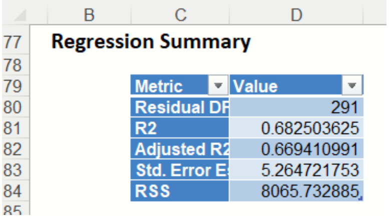
Output Options
Summary report of scoring on training data
Detailed report of scoring on training data
Lift charts on training data
Frequency chart on training data
Summary report of scoring on validation data
Detailed report of scoring on validation data
Lift charts on validation data
Frequency chart on validation data

- **Regression Summary:** Summary Statistics, in the *Regression Summary* table, show the residual degrees of freedom (#observations - #predictors), the R-squared value, a standard deviation type measure for the model (which typically has a chi-square distribution), and the Residual Sum of Squares error.

The R-squared value shown here is the r-squared value for a logistic regression model , defined as -

$R^2 = (D_0\text{-}D)/D_0$,

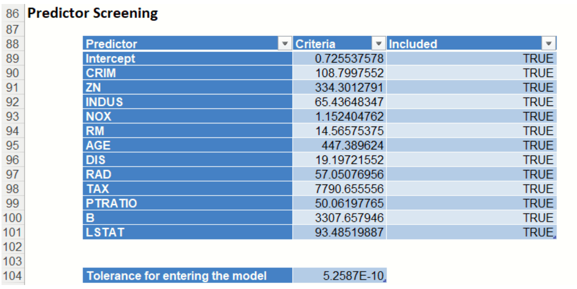
where D is the Deviance based on the fitted model and D_0 is the deviance based on the null model. The null model is defined as the model containing no predictor variables apart from the constant.



Regression Summary

Metric	Value
Residual DF	291
R2	0.682503625
Adjusted R2	0.669410991
Std. Error E	5.264721753
RSS	8065.732885

- **Predictor Screening:** In Analytic Solver Data Science, a new preprocessing feature selection step was added in V2015 to take advantage of automatic variable screening and elimination, using Rank-Revealing QR Decomposition. This allows Analytic Solver Data Science to identify the variables causing multicollinearity, rank deficiencies, and other problems that would otherwise cause the algorithm to fail. Information about "bad" variables is used in Variable Selection and Multicollinearity Diagnostics, and in computing other reported statistics. Included and excluded predictors are shown in the Model Predictors table. In this model, all predictors were included in the model; all predictors were eligible to enter the model passing the tolerance threshold. This denotes a tolerance beyond which a variance – covariance matrix is not exactly singular to within machine precision. The test is based on the diagonal elements of the triangular factor R resulting from Rank-Revealing QR Decomposition. Predictors that do not pass the test are excluded.



Predictor Screening

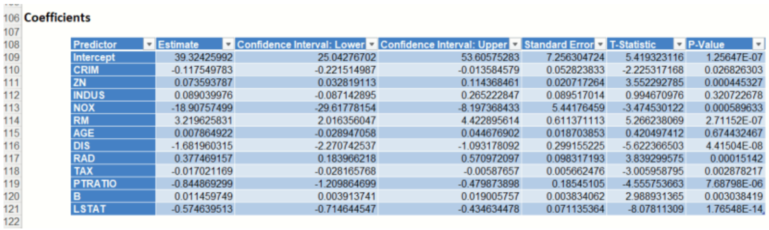
Predictor	Criteria	Included
Intercept	0.725537578	TRUE
CRIM	108.7997552	TRUE
ZN	334.3012791	TRUE
INDUS	65.43648347	TRUE
NOX	1.152404762	TRUE
RM	14.56575375	TRUE
AGE	447.389624	TRUE
DIS	19.19721552	TRUE
RAD	57.05076956	TRUE
TAX	7790.655556	TRUE
PTRATIO	50.06197765	TRUE
B	3307.657946	TRUE
LSTAT	93.48519887	TRUE

Tolerance for entering the model	5.2587E-10

Note: If a predictor is excluded, the corresponding coefficient estimates will be 0 in the regression model and the variable – covariance matrix would contain all zeros in the rows and columns that correspond to the excluded predictor. Multicollinearity diagnostics, variable selection and other remaining output will be calculated for the reduced model.

The design matrix may be rank-deficient for several reasons. The most common cause of an ill-conditioned regression problem is the presence of feature(s) that can be exactly or approximately represented by a linear combination of other feature(s). For example, assume that among predictors you have 3 input variables X, Y, and Z where Z = a * X + b * Y and a and b are constants. This will cause the design matrix to not have a full rank. Therefore, one of these 3 variables will not pass the threshold for entrance and will be excluded from the final regression model.

- **Coefficients:** The Regression Model table contains the estimate, the standard error of the coefficient, the p-value and confidence intervals for each variable included in the model.



Coefficients

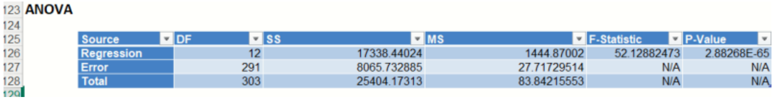
Predictor	Estimate	Confidence Interval: Lower	Confidence Interval: Upper	Standard Error	T-Statistic	P-Value
Intercept	39.32425992	25.04276702	53.60575283	7.256304724	5.419323116	1.25647E-07
CRIM	-0.117549783	-0.221514987	-0.013584579	0.052823833	-2.225317168	0.026826303
ZN	0.073593787	0.032819113	0.114368461	0.020717264	3.552292785	0.000445327
INDUS	0.089039976	-0.087142895	0.265222847	0.089517014	0.994670976	0.320722678
NOX	-18.90757499	-29.61778154	-8.197368433	5.44176459	-3.474530122	0.000589633
RM	3.219625831	2.016356047	4.422895614	0.611371113	5.266238069	2.71152E-07
AGE	0.007864922	-0.028947058	0.044676902	0.018703853	0.420497412	0.674432467
DIS	-1.681960315	-2.270742537	-1.093178092	0.299155225	-5.622366503	4.41504E-08
RAD	0.377469157	0.183966218	0.570972097	0.098317193	3.839299575	0.00015142
TAX	-0.017021169	-0.028165768	-0.00587657	0.005662476	-3.005958795	0.002878217
PTRATIO	-0.844869299	-1.209864699	-0.479873898	0.18545105	-4.555753663	7.68798E-06
B	0.011459749	0.003913741	0.019005757	0.003834062	2.988931365	0.003038419
LSTAT	-0.574639513	-0.714644547	-0.434634478	0.071135364	-8.07811309	1.76548E-14

Note: If a variable has been eliminated by Rank-Revealing QR Decomposition, the variable will appear in red in the Coefficients table with a 0 for Estimate, CI Lower, CI Upper, Standard Error and N/A for T-Statistic and P-Value.

The Standard Error is calculated as each variable is introduced into the model beginning with the constant term and continuing with each variable as it appears in the dataset.

Analytic Solver Data Science produces 95% Confidence Intervals for the estimated values. For a given record, the Confidence Interval gives the mean value estimation with 95% probability. This means that with 95% probability, the regression line will pass through this interval.

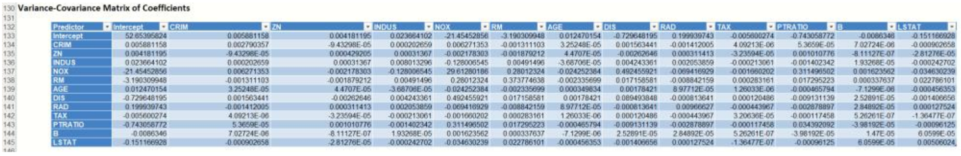
- **ANOVA:** The ANOVA table includes the degrees of freedom (DF), sum of squares (SS), mean squares (MS), F-Statistic and P-Value.



ANOVA

Source	DF	SS	MS	F-Statistic	P-Value
Regression	12	17338.44024	1444.87002	52.12882473	2.88268E-65
Error	291	8065.732885	27.71729514	N/A	N/A
Total	303	25404.17313	83.84215553	N/A	N/A

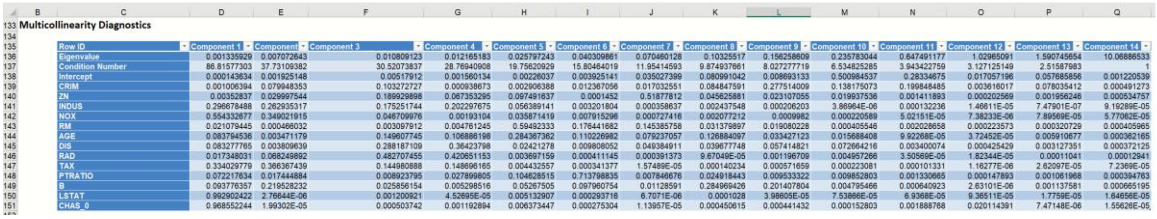
- **Variance-Covariance Matrix of Coefficients:** Entries in the matrix are the covariances between the indicated coefficients. The "on-diagonal" values are the estimated variances of the corresponding coefficients.



Variance-Covariance Matrix of Coefficients

Predictor	Intercept	CRIM	ZN	INDUS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
Intercept	52.65395824	0.005881158	0.004181195	0.023664102	-21.45452856	-3.190309948	0.012470154	-0.729648195	0.199939743	-0.005600274	-0.743058772	-0.0086346	-0.151166928
CRIM	0.005881158	0.002790357	-9.43298E-05	0.000202659	0.006271353	-0.001311103	3.25248E-05	0.001563441	-0.001412005	4.09213E-06	5.3659E-05	7.02724E-06	-0.000902658
ZN	0.004181195	-9.43298E-05	0.000429205	0.00031367	-0.002178303	-0.001879212	4.4707E-05	-0.00262646	0.000311413	-3.23594E-05	0.001010776	-8.11127E-07	-2.81276E-05
INDUS	0.023664102	0.000202659	0.00031367	0.008013296	-0.128006545	0.00491496	-3.68706E-05	0.004243361	0.002053859	-0.000213061	-0.001402342	1.93268E-05	-0.000242702
NOX	-21.45452856	0.006271353	-0.002178303	-0.128006545	29.61280186	0.28012324	-0.024252384	0.492455921	-0.069416929	-0.001660202	0.311496502	0.001623562	-0.034630239
RM	-3.190309948	-0.001311103	-0.001879212	0.00491496	0.28012324	0.373774638	-0.002335699	0.017158581	-0.008842159	0.000283161	0.017295223	0.000337637	0.022786101
AGE	0.012470154	3.25248E-05	4.4707E-05	-3.68706E-05	-0.024252384	-0.002335699	0.000349834	0.00178421	8.97712E-05	1.26033E-06	-0.000465794	-7.1299E-06	-0.000456353
DIS	-0.729648195	0.001563441	-0.00262646	0.004243361	0.492455921	0.017158581	0.00178421	0.089493848	-0.000813641	0.000120486	-0.009131139	2.52891E-05	-0.001406656
RAD	0.199939743	-0.001412005	0.000311413	0.002053859	-0.069416929	-0.008842159	8.97712E-05	-0.000813641	0.00966627	-0.000443967	-0.002878897	2.84892E-05	0.000127524
TAX	-0.005600274	4.09213E-06	-3.23594E-05	-0.000213061	-0.001660202	0.000283161	1.26033E-06	0.000120486	-0.000443967	3.20636E-05	-0.000117458	5.26261E-07	-1.36477E-07
PTRATIO	-0.743058772	5.3659E-05	0.001010776	-0.001402342	0.311496502	0.017295223	-0.000465794	-0.009131139	-0.002878897	-0.000117458	0.034392092	-3.98192E-05	-0.00096125
B	-0.0086346	7.02724E-06	-8.11127E-07	1.93268E-05	0.001623562	0.000337637	-7.1299E-06	2.52891E-05	2.84892E-05	5.26261E-07	-3.98192E-05	1.47E-05	6.0599E-05
LSTAT	-0.151166928	-0.000902658	-2.81276E-05	-0.000242702	-0.034630239	0.022786101	-0.000456353	-0.001406656	0.000127524	-1.36477E-07	-0.00096125	6.0599E-05	0.00506024

- **Multicollinearity Diagnostics:** This table helps assess whether two or more variables so closely track one another as to provide essentially the same information.



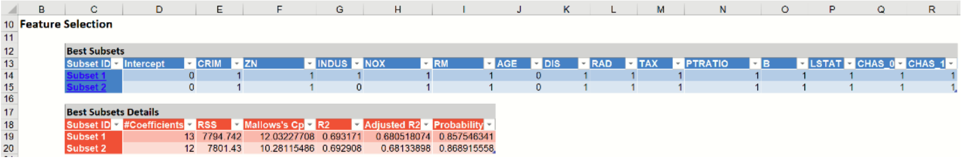
Multicollinearity Diagnostics

Row ID	Component 1	Component	Component 3	Component 4	Component 5	Component 6	Component 7	Component 8	Component 9	Component 10	Component 11	Component 12	Component 13	Component 14
Eigenvalue	0.001335929	0.007072643	0.010809123	0.012165183	0.025797243	0.040309861	0.070460128	0.10325517	0.156258609	0.235783044	0.647491177	1.02965091	1.590745654	10.06886533
Condition Number	86.81577303	37.73109382	30.52073837	28.76940908	19.75620929	15.80464019	11.95414593	9.874937661	8.027277719	6.534825285	3.943422759	3.127125149	2.51587983	1
Intercept	0.000143634	0.001925148	0.00517912	0.001560134	0.00226037	0.003925141	0.035027399	0.080991042	0.008693133	0.500984537	0.28334675	0.017057196	0.057685856	0.001220539
CRIM	0.001006394	0.079948353	0.103272727	0.000938673	0.002906388	0.012367056	0.017032551	0.084847591	0.277514009	0.138175073	0.199848485	0.003616017	0.078035412	0.000491273
ZN	0.00352837	0.029997544	0.189929898	0.067353295	0.097491637	0.0001452	0.51877812	0.045625881	0.023107055	0.019937536	0.001411893	0.000202569	0.001956246	0.000534757
INDUS	0.296678488	0.262935317	0.175251744	0.202297675	0.056389141	0.003201804	0.000358637	0.002437548	0.000206203	3.86964E-06	0.000132236	1.46611E-05	7.47901E-07	9.19289E-05
NOX	0.554332677	0.349021915	0.046709976	0.00193104	0.035871419	0.007915296	0.000727416	0.002077212	0.0009982	0.000220589	5.02151E-05	7.38233E-06	7.89569E-05	5.77062E-05
RM	0.021079445	0.000466032	0.003097912	0.004761245	0.59492333	0.176441682	0.145385758	0.031379897	0.019080228	0.000405546	0.002028658	0.000223573	0.000320729	0.000405965
AGE	0.083794536	0.003471179	0.149607745	0.106886198	0.284367362	0.110226982	0.079237057	0.126884097	0.033427123	0.015688408	9.92268E-05	3.72452E-05	0.005910677	0.000362165
DIS	0.083277765	0.003809639	0.288187109	0.36423798	0.02421278	0.009808052	0.049384911	0.039677748	0.057414821	0.072664216	0.003400074	0.000425429	0.003127351	0.000372125
RAD	0.017348031	0.068249892	0.482707455	0.420651153	0.003697159	0.000411145	0.000391373	9.67049E-05	0.001196709	0.004957266	3.50569E-05	1.82344E-05	0.00011041	0.00012941
TAX	0.334029779	0.366367439	0.144980888	0.148696165	0.004432557	0.000341377	1.57489E-05	0.000140234	0.000571659	0.000223081	0.000101331	1.16277E-06	2.62097E-05	7.2369E-05
PTRATIO	0.072217634	0.017444884	0.008923795	0.027899805	0.104628515	0.713798835	0.007846676	0.024918443	0.009533322	0.009852803	0.001330665	0.000147893	0.001061968	0.000394763
B	0.093776357	0.219528232	0.025856154	0.005298516	0.05267505	0.097960754	0.01128591	0.284969426	0.201407804	0.004795466	0.000640923	2.63101E-06	0.001137581	0.000665195
LSTAT	0.992902422	2.76644E-06	0.001200921	4.52695E-05	0.005132907	0.000293716	6.7071E-06	0.0001028	3.98605E-05	7.53866E-05	6.9368E-05	9.36511E-05	1.7759E-05	1.64656E-05
CHAS_0	0.968552244	1.99302E-05	0.000503742	0.001192894	0.006373447	0.000275304	1.13957E-05	0.000450615	0.000441432	0.000152803	0.001888768	0.020114391	7.47148E-06	1.55626E-05

The columns represent the variance components (related to principal components in multivariate analysis), while the rows represent the variance proportion decomposition explained by each variable in the model. The eigenvalues are those associated with the singular value decomposition of the variance-covariance matrix of the coefficients, while the condition numbers are the ratios of the square root of the largest eigenvalue to all the rest. In general, multicollinearity is likely to be a problem with a high condition number (more than 20 or 30), and high variance decomposition proportions (say more than 0.5) for two or more variables.

LinReg_FS

Select the **Feature Selection** link on the Output Navigator to display the Variable Selection table. This table displays a list of different models generated using the selections made on the *Feature Selection* dialog. When *Backward elimination* is used, Linear Regression may stop early when there is no variable eligible for elimination as evidenced in the table below (i.e. there are no subsets with less than 12 coefficients). Since *Fit Intercept* was selected on the *Parameters* tab, each subset includes an intercept.



Feature Selection

Best Subsets

Subset ID	Intercept	CRIM	ZN	INDUS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	CHAS_0	CHAS_1
Subset 1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1
Subset 2	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1

Best Subsets Details

Subset ID	#Coefficients	RSS	Mallows's Cp	R2	Adjusted R2	Probability
Subset 1	13	7794.742	12.03227708	0.693171	0.680518074	0.857546341
Subset 2	12	7801.43	10.28115486	0.692908	0.68133898	0.868915558

The error values for the Best Subsets are:

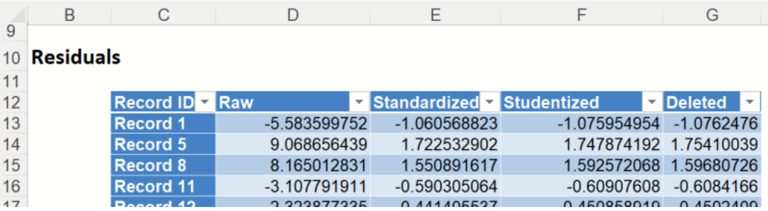
- RSS: The residual sum of squares, or the sum of squared deviations between the predicted probability of success and the actual value (1 or 0)
- Cp: Mallows Cp (Total squared error) is a measure of the error in the best subset model, relative to the error incorporating all variables. Adequate models are those for which Cp is roughly equal to the number of parameters in the model (including the constant), and/or Cp is at a minimum
- R-Squared: R-squared Goodness-of-fit
- Adj. R-Squared: Adjusted R-Squared values.
- "Probability" is a quasi hypothesis test of the proposition that a given subset is acceptable; if Probability < .05 we can rule out that subset.

Compare the RSS value as the number of coefficients in the subset increases from 13 to 12 (7794.742 to 7801.43). The RSS for 12 coefficients is just slightly higher than the RSS for 13 coefficients suggesting that a model with 12 coefficients may be sufficient to fit a regression.

LinReg_ResidInfluence

This output sheet includes two output tables: Residuals and Influence Diagnostics.

- **Residuals:** Click the **Residuals** link in the Output Navigator to open the Residuals table. This table displays the Raw Residuals, Standarized Residuals, Studentized Residuals and Deleted Residuals.



Residuals

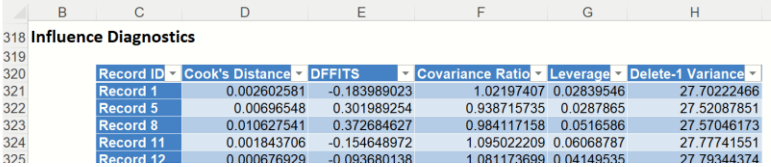
Record ID	Raw	Standardized	Studentized	Deleted
Record 1	-5.583599752	-1.060568823	-1.075954954	-1.0762476
Record 5	9.068656439	1.722532902	1.747874192	1.75410039
Record 8	8.165012831	1.550891617	1.592572068	1.59680726
Record 11	-3.107791911	-0.590305064	-0.60907608	-0.6084166
Record 12	2.323877335	0.441405537	0.450858919	0.4502409

Studentized residuals are computed by dividing the unstandardized residuals by quantities related to the diagonal elements of the hat matrix, using a common scale estimate computed without the i^{th} case in the model. These residuals have t - distributions with (n-k-1) degrees of freedom. As a result, any residual with absolute value exceeding 3 usually requires attention.

The Deleted residual is computed for the i^{th} observation by first fitting a model without the i^{th} observation, then using this model to predict the i^{th} observation. Afterwards the difference is taken between the predicted observation and the actual observation.

- **Influence Diagnostics:** Click the Influence Diagnositics link on the Output Navigator to display the Influence Diagnostics data table. This table contains various statistics computed by Analytic Solver Data Science.

  The Cooks Distance for each observation is displayed in this table. This is an overall measure of the impact of the i^{th} datapoint on the estimated regression coefficient. In linear models Cooks Distance has, approximately, an F distribution with k and (n-k) degrees of freedom.



Influence Diagnostics

Record ID	Cook's Distance	DFFITS	Covariance Ratio	Leverage	Delete-1 Variance
Record 1	0.002602581	-0.183989023	1.02197407	0.02839546	27.70222466
Record 5	0.00696548	0.301989254	0.938715735	0.0287865	27.52087851
Record 8	0.010627541	0.372684627	0.984117158	0.0516586	27.57046173
Record 11	0.001843706	-0.154648972	1.095022209	0.06068787	27.77741551
Record 12	0.000676929	-0.093680138	1.081173699	0.04149535	27.79344374

The DF fits for each observation is displayed in the output. DFFits gives information on how the fitted model would change if a point was not included in the model.

Analytic Solver Data Science computes DFFits using the following computation.

$$DFFits_i = \frac{\hat{y}_i - \hat{y}_i(-i)}{\sigma(-i)\sqrt{h_i}} = \cdots = \frac{\sqrt{h_i}e_i}{\sigma(-i)(1-h_i)} = \cdots = e_i^{stud}\sqrt{\frac{h_i}{1-h_i}}$$

where

y_hat_i = i-th fitted value from full model

y_hat_i(-i) = i-th fitted value from model not including i-th observation

sigma(-i) = estimated error variance of model not including i-th observation

h_i = leverage of i-th point (i.e. {i,i}-th element of Hat Matrix)

e_i = i-th residual from full model

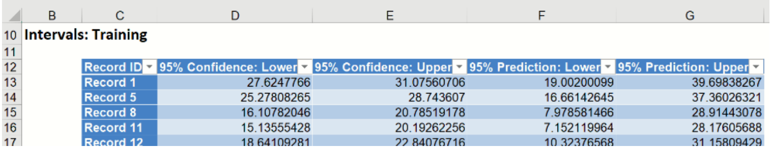
e_i^stud = i-th Studentized residual

The covariance ratios are displayed in the output. This measure reflects the change in the variance-covariance matrix of the estimated coefficients when the i^{th} observation is deleted.

The diagonal elements of the hat matrix are displayed under the Leverage column. This measure is also known as the Leverage of the i^{th} observation.

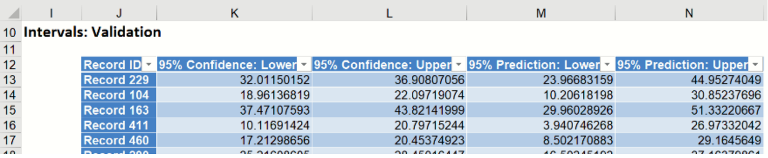
LinReg_Intervals

Click either the **Intervals: Training** or **Intervals: Validation** links in the Output Navigator to view the Intervals report for both the Training and Validation partitions.   Of primary interest in a data-science context will be the predicted and actual values for each record, along with the residual (difference) and Confidence and Prediction Intervals for each predicted value.



Intervals: Training

Record ID	95% Confidence: Lower	95% Confidence: Upper	95% Prediction: Lower	95% Prediction: Upper
Record 1	27.6247766	31.07560706	19.00200099	39.69838267
Record 5	25.27808265	28.743607	16.66142645	37.36026321
Record 8	16.10782046	20.78519178	7.978581466	28.91443078
Record 11	15.13555428	20.19262256	7.152119964	28.17605688
Record 12	18.64109281	22.84076716	10.32376568	31.15809429



Intervals: Validation

Record ID	95% Confidence: Lower	95% Confidence: Upper	95% Prediction: Lower	95% Prediction: Upper
Record 229	32.01150152	36.90807056	23.96683159	44.95274049
Record 104	18.96136819	22.09719074	10.20618198	30.85237696
Record 163	37.47107593	43.82141999	29.96028926	51.33220667
Record 411	10.11691424	20.79715244	3.940746268	26.97332042
Record 460	17.21298656	20.45374923	8.502170883	29.1645649

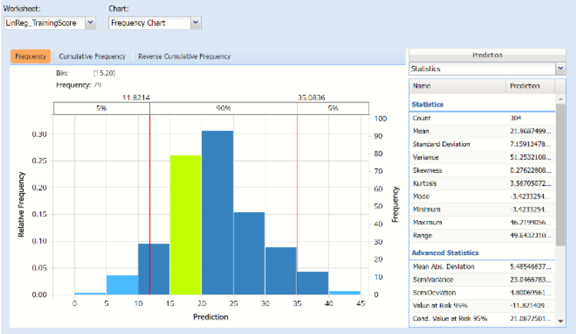
Analytic Solver Data Science produces 95% Confidence and Prediction Intervals for the predicted values. Typically, Prediction Intervals are more widely utilized as they are a more robust range for the predicted value. For a given record, the Confidence Interval gives the mean value estimation with 95% probability. This means that with 95% probability, the regression line will pass through this interval. The Prediction Interval takes into account possible future deviations of the predicted response from the mean. There is a 95% chance that the predicted value will lie within the Prediction interval.

LinReg_TrainingScore

Of primary interest in a data-science context will be the predicted and actual values for the MEDV variable along with the residual (difference) for each predicted value in the Training partition.

LinReg_TrainingScore displays the newly added Output Variable frequency chart, the Training: Prediction Summary and the Training: Prediction Details report. All calculations, charts and predictions on the LinReg_TrainingScore output sheet apply to the Training partition.

> Note: To view charts in the Cloud app, click the Charts icon on the Ribbon, select a worksheet under Worksheet and a chart under Chart.



Worksheet: LinReg_TrainingScore

Chart: Frequency Chart

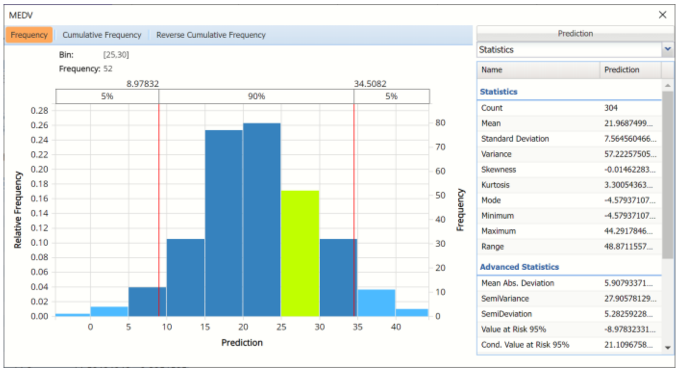
Bin: [15,20]

Frequency: 79

Name	Prediction
Statistics	
Count	304
Mean	21.9687499...
Standard Deviation	7.15913478...
Variance	51.2532108...
Skewness	0.27622808...
Kurtosis	3.56705072...
Mode	-3.4233254...
Minimum	-3.4233254...
Maximum	46.2199856...
Range	49.6432310...
Advanced Statistics	
Mean Abs. Deviation	5.48546637...
SemiVariance	23.0466783...
SemiDeviation	4.80069561...
Value at Risk 95%	-11.821409...
Cond. Value at Risk 95%	21.0872501...

- **Frequency Charts:** The output variable frequency chart for the training partition opens automatically once the *LinReg_TrainingScore* worksheet is selected. To close this chart, click the "x" in the upper right hand corner of the chart. To reopen, click onto another tab and then click back to the *LinReg_TrainingScore* tab. To move the dialog to a new location on the screen, simply grab the title bar and drag the dialog to the desired location. This chart displays a detailed, interactive frequency chart for the predicted values in the simulated, or synthetic data, and for the predicted values in the training partition.

Training (Actual) data, MEDV variable



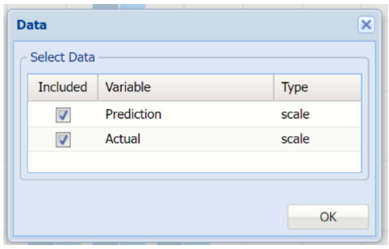
Prediction

Statistics

Name	Prediction
Statistics	
Count	304
Mean	21.9687499...
Standard Deviation	7.564560466...
Variance	57.22257505...
Skewness	-0.01462283...
Kurtosis	3.30054363...
Mode	-4.57937107...
Minimum	-4.57937107...
Maximum	44.2917846...
Range	48.8711557...
Advanced Statistics	
Mean Abs. Deviation	5.90793371...
SemiVariance	27.90578129...
SemiDeviation	5.28259228...
Value at Risk 95%	-8.97832331...
Cond. Value at Risk 95%	21.1096758...

To display the predicted values in the synthetic and training data, click Prediction in the upper right hand corner and select both checkboxes in the Data dialog.

Click Actual to add Prediction data to the interactive chart

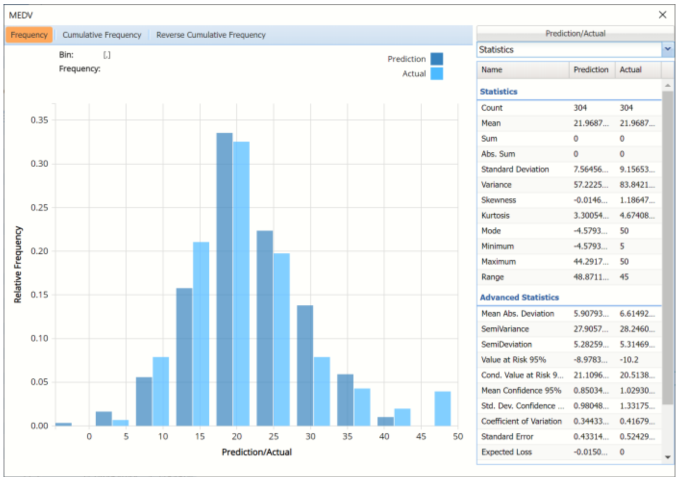


Data

Select Data

Included	Variable	Type
☑	Prediction	scale
☑	Actual	scale

Actual vs Predicted for Training Partition, MEDV variable

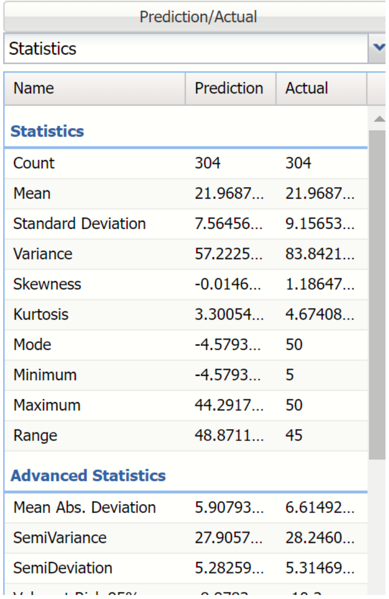


Prediction/Actual

Statistics

Name	Prediction	Actual
Statistics		
Count	304	304
Mean	21.9687...	21.9687...
Sum	0	0
Abs. Sum	0	0
Standard Deviation	7.56456...	9.15653...
Variance	57.2225...	83.8421...
Skewness	-0.0146...	1.18647...
Kurtosis	3.30054...	4.67408...
Mode	-4.5793...	50
Minimum	-4.5793...	5
Maximum	44.2917...	50
Range	48.8711...	45
Advanced Statistics		
Mean Abs. Deviation	5.90793...	6.61492...
SemiVariance	27.9057...	28.2460...
SemiDeviation	5.28259...	5.31469...
Value at Risk 95%	-8.9783...	-10.2
Cond. Value at Risk 9...	21.1096...	20.5138...
Mean Confidence 95%	0.85034...	1.02930...
Std. Dev. Confidence ...	0.98048...	1.33175...
Coefficient of Variation	0.34433...	0.41679...
Standard Error	0.43314...	0.52429...
Expected Loss	-0.0150...	0

Notice in the screenshot below that both the Actual and Prediction data appear in the chart together, and statistics for both data appear on the right.



Name	Prediction	Actual
Statistics		
Count	304	304
Mean	21.9687...	21.9687...
Standard Deviation	7.56456...	9.15653...
Variance	57.2225...	83.8421...
Skewness	-0.0146...	1.18647...
Kurtosis	3.30054...	4.67408...
Mode	-4.5793...	50
Minimum	-4.5793...	5
Maximum	44.2917...	50
Range	48.8711...	45
Advanced Statistics		
Mean Abs. Deviation	5.90793...	6.61492...
SemiVariance	27.9057...	28.2460...
SemiDeviation	5.28259...	5.31469...

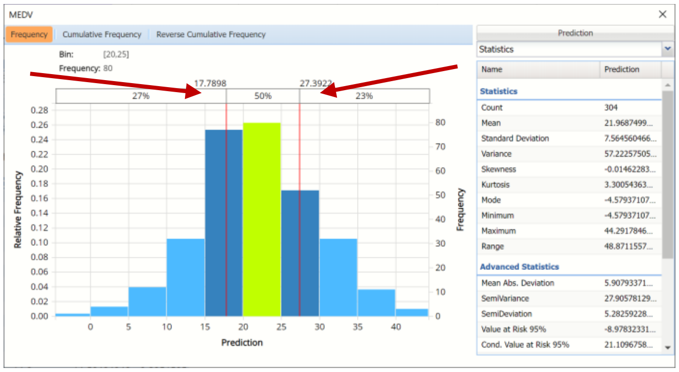
To remove either the Actual or Prediction data from the chart, click Prediction/Actual in the top right and then uncheck the data type to be removed.

This chart behaves the same as the interactive chart in the Analyze Data feature found on the Explore menu (and explained in depth in the Data Science Reference Guide).

- Use the mouse to hover over any of the bars in the graph to populate the Bin and Frequency headings at the top of the chart.

- When displaying either Original or Synthetic data (not both), red vertical lines will appear at the 5% and 95% percentile values in all three charts (Frequency, Cumulative Frequency and Reverse Cumulative Frequency) effectively displaying the 90^{th} confidence interval. The middle percentage is the percentage of all the variable values that lie within the 'included' area, i.e. the darker shaded area. The two percentages on each end are the percentage of all variable values that lie outside of the 'included' area or the "tails". i.e. the lighter shaded area. Percentile values can be altered by moving either red vertical line to the left or right.

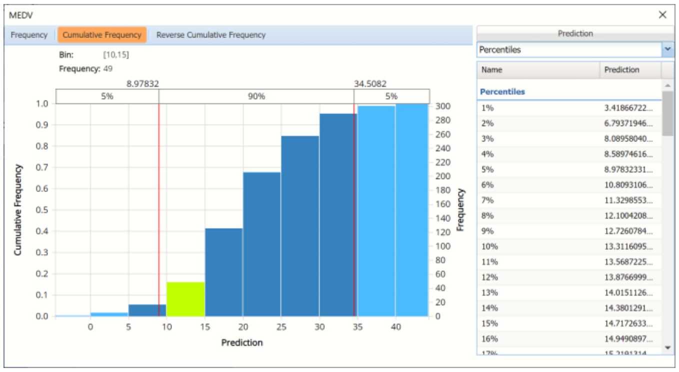
Frequency Chart for MEDV Prediction with red percentile lines moved



Name	Prediction
Statistics	
Count	304
Mean	21.9687499...
Standard Deviation	7.564560466...
Variance	57.22257505...
Skewness	-0.01462283...
Kurtosis	3.30054363...
Mode	-4.57937107...
Minimum	-4.57937107...
Maximum	44.2917846...
Range	48.8711557...
Advanced Statistics	
Mean Abs. Deviation	5.90793371...
SemiVariance	27.90578129...
SemiDeviation	5.28259228...
Value at Risk 95%	-8.97832331...
Cond. Value at Risk 95%	21.1096758...

- Click Cumulative Frequency and Reverse Cumulative Frequency tabs to see the Cumulative Frequency and Reverse Cumulative Frequency charts, respectively.

Cumulative Frequency chart with Percentiles displayed



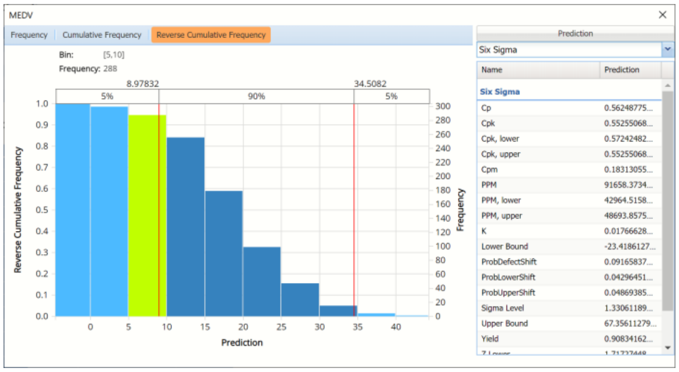
Bin: [10,15]
Frequency: 49

Percentiles

Name	Prediction
Percentiles	
1%	3.41866722...
2%	6.79371946...
3%	8.08958040...
4%	8.58974616...
5%	8.97832331...
6%	10.8093106...
7%	11.3298553...
8%	12.1004208...
9%	12.7260784...
10%	13.3116095...
11%	13.5687225...
12%	13.8766999...
13%	14.0151126...
14%	14.3801291...
15%	14.7172633...
16%	14.9490897...

- Click the down arrow next to Statistics to view Percentiles for each type of data along with Six Sigma indices. Use the Chart Options view to manually select the number of bins to use in the chart, as well as to set personalization options.

Reverse Cumulative Frequency chart with Six Sigma indices displayed



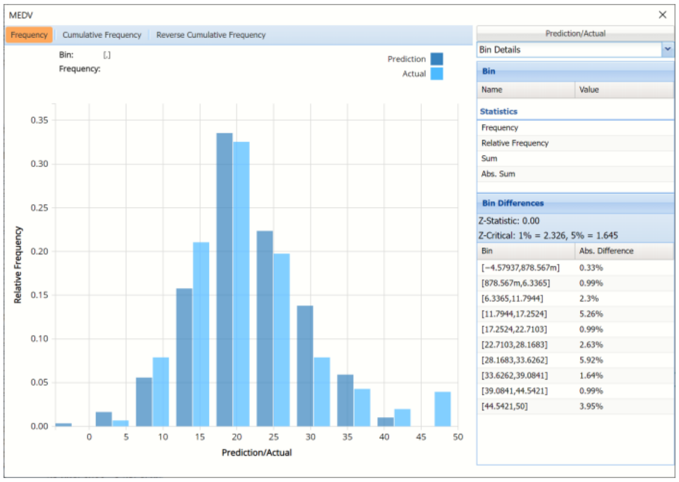
Bin: [5,10]
Frequency: 288

Six Sigma

Name	Prediction
Six Sigma	
Cp	0.56248775...
Cpk	0.55255068...
Cpk, lower	0.57242482...
Cpk, upper	0.55255068...
Cpm	0.18313055...
PPM	91658.3734...
PPM, lower	42964.5158...
PPM, upper	48693.8575...
K	0.01766628...
Lower Bound	-23.4186127...
ProbDefectShift	0.09165837...
ProbLowerShift	0.04296451...
ProbUpperShift	0.04869385...
Sigma Level	1.33061189...
Upper Bound	67.35611279...
Yield	0.90834162...

- Click the down arrow next to Statistics to view Bin Details for each bin in the chart. If both datasets are included in the chart, the Bin Differences are added to the pane.

Frequency chart with Bin Details displayed



Bin: [,]
Frequency:

Bin Details

Bin

Name	Value
Statistics	
Frequency	
Relative Frequency	
Sum	
Abs. Sum	

Bin Differences

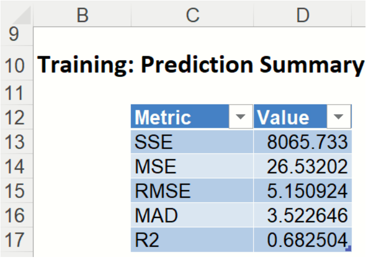
Z-Statistic: 0.00
Z-Critical: 1% = 2.326, 5% = 1.645

Bin	Abs. Difference
[−4.57937,878.567m]	0.33%
[878.567m,6.3365]	0.99%
[6.3365,11.7944]	2.3%
[11.7944,17.2524]	5.26%
[17.2524,22.7103]	0.99%
[22.7103,28.1683]	2.63%
[28.1683,33.6262]	5.92%
[33.6262,39.0841]	1.64%
[39.0841,44.5421]	0.99%
[44.5421,50]	3.95%

As discussed above, see the Analyze Data section of the Exploring Data chapter in the Data Science Reference Guide for an in-depth discussion of this chart as well as descriptions of all statistics, percentiles, bin details and six sigma indices.

- **Prediction Summary:** In the Prediction Summary report, Analytic Solver Data Science displays the total sum of squared errors summaries for the training partition. The total sum of squared errors is the sum of the squared errors (deviations between predicted and actual values) and the root mean square error (square root of the average squared error). The average error is typically very small, because positive prediction errors tend to be counterbalanced by negative ones.

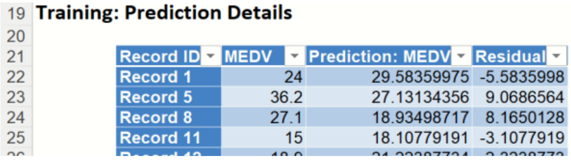
  Training Prediction Summary



Training: Prediction Summary

Metric	Value
SSE	8065.733
MSE	26.53202
RMSE	5.150924
MAD	3.522646
R2	0.682504

- **Prediction Details:** Scroll down to the Training: Prediction Details report to find the Prediction value for the MEDV variable for each record, as well as the Residual value.



Training: Prediction Details

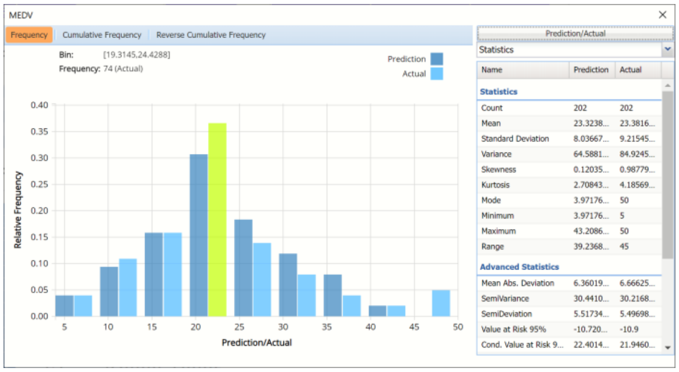
Record ID	MEDV	Prediction: MEDV	Residual
Record 1	24	29.58359975	-5.5835998
Record 5	36.2	27.13134356	9.0686564
Record 8	27.1	18.93498717	8.1650128
Record 11	15	18.10779191	-3.1077919

LinReg_ValidationLiftChart

Another interest in a data-science context will be the predicted and actual values for the MEDV variable along with the residual (difference) for each predicted value in the Validation partition.

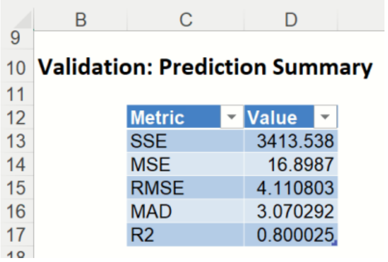
LinReg_ValidationScore displays the newly added Output Variable frequency chart, the Training: Prediction Summary and the Training: Prediction Details report. All calculations, charts and predictions on the LinReg_ValidationScore output sheet apply to the Validation partition.

- **Frequency Charts:** The output variable frequency chart for the validation partition opens automatically once the *LinReg_ValidationScore* worksheet is selected. This chart displays a detailed, interactive frequency chart for the Actual variable data and the Predicted data, for the validation partition. For more information on this chart, see the LinReg_TrainingLiftChart explanation above.



Name	Prediction	Actual
Statistics		
Count	202	202
Mean	23.3238...	23.3816...
Standard Deviation	8.03667...	9.21545...
Variance	64.5881...	84.9245...
Skewness	0.12035...	0.98779...
Kurtosis	2.70843...	4.18569...
Mode	3.97176...	50
Minimum	3.97176...	5
Maximum	43.2086...	50
Range	39.2368...	45
Advanced Statistics		
Mean Abs. Deviation	6.36019...	6.66625...
SemiVariance	30.4410...	30.2168...
SemiDeviation	5.51734...	5.49698...
Value at Risk 95%	-10.720...	-10.9
Cond. Value at Risk 9...	22.4014...	21.9460...

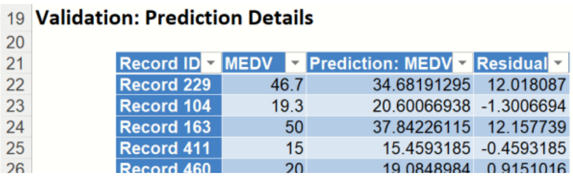
- **Prediction Summary:** In the Prediction Summary report, Analytic Solver Data Science displays the total sum of squared errors summaries for the Validation partition.



Validation: Prediction Summary

Metric	Value
SSE	3413.538
MSE	16.8987
RMSE	4.110803
MAD	3.070292
R2	0.800025

- **Prediction Details:** Scroll down to the Training: Prediction Details report to find the Prediction value for the MEDV variable for each record, as well as the Residual value.



Validation: Prediction Details

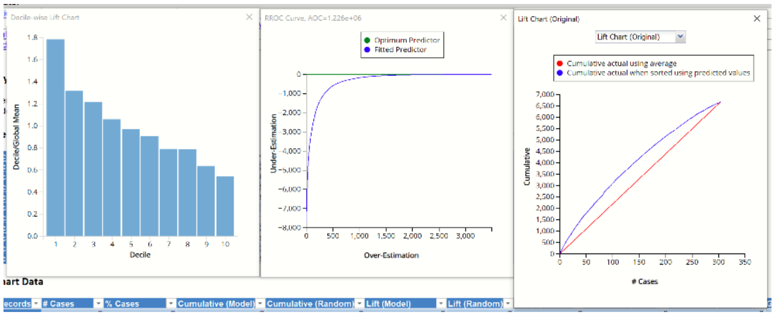
Record ID	MEDV	Prediction: MEDV	Residual
Record 229	46.7	34.68191295	12.018087
Record 104	19.3	20.60066938	-1.3006694
Record 163	50	37.84226115	12.157739
Record 411	15	15.4593185	-0.4593185
Record 460	20	19.0848984	0.9151016

LinReg_TrainingLiftChart & LinReg_ValidationLiftChart

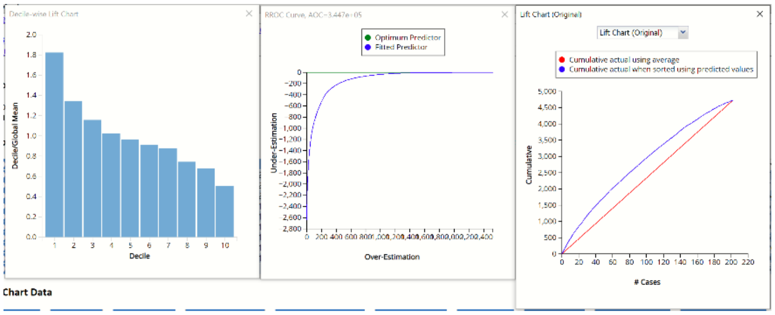
Lift charts and RROC Curves (on the *LinReg_TrainingLiftChart* and *LinReg_ValidationLiftChart* tabs, respectively) are visual aids for measuring model performance. Lift Charts consist of a lift curve and a baseline. The greater the area between the lift curve and the baseline, the better the model. RROC (regression receiver operating characteristic) curves plot the performance of regressors by graphing over-estimations (or predicted values that are too high) versus underestimations (or predicted values that are too low.) The closer the curve is to the top left corner of the graph (in other words, the smaller the area above the curve), the better the performance of the model.

After the model is built using the training data set, the model is used to score on the training data set and the validation data set (if one exists). Then the data set(s) are sorted using the predicted output variable value. After sorting, the actual outcome values of the output variable are cumulated and the lift curve is drawn as the number of cases versus the cumulated value. The baseline (red line connecting the origin to the end point of the blue line) is drawn as the number of cases versus the average of actual output variable values multiplied by the number of cases. The decilewise lift curve is drawn as the decile number versus the cumulative actual output variable value divided by the decile's mean output variable value. The bars in this chart indicate the factor by which the MLR model outperforms a random assignment, one decile at a time. Refer to the validation graph below. In the first decile in the validation dataset, taking the most expensive predicted housing prices in the dataset, the predictive performance of the model is about 1.8 times better as simply assigning a random predicted value.

Decile-Wise Lift Chart, ROC Curve and Lift Chart from Training Partition

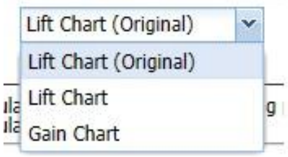


Decile-Wise Lift Chart, ROC Curve and Lift Chart from Validation Partition



In an RROC curve, we can compare the performance of a regressor with that of a random guess (red line) for which under estimations are equal to over-estimations shifted to the minimum under estimate. Anything to the left of this line signifies a better prediction and anything to the right signifies a worse prediction. The best possible prediction performance would be denoted by a point at the top left of the graph at the intersection of the x and y axis. This point is sometimes referred to as the "perfect classification". Area Over the Curve (AOC) is the space in the graph that appears above the ROC curve and is calculated using the formula: sigma^2 * n^2/2 where n is the number of records The smaller the AOC, the better the performance of the model. In this example we see that the area above the curve in both datasets, or the AOC, is fairly large which indicates that this model might not be the best fit to the data.

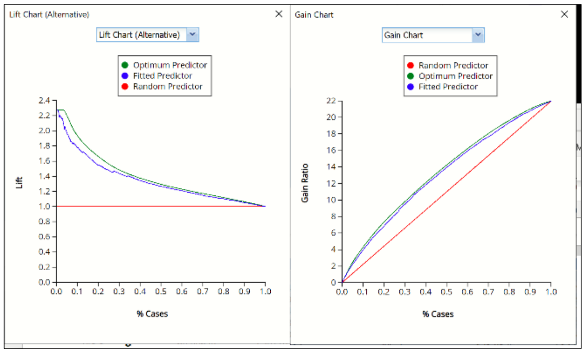
Two new charts were introduced in V2017: a new Lift Chart and the Gain Chart. To display these new charts, click the down arrow next to Lift Chart (Original), in the Original Lift Chart, then select the desired chart.



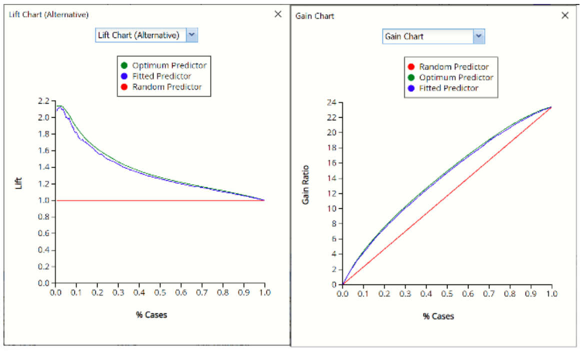
Select Lift Chart to display Analytic Solver Data Science's new Lift Chart. Each of these charts consists of an Optimum Predictor curve, a Fitted Predictor curve, and a Random Predictor curve. The Optimum Predictor curve plots a hypothetical model that would provide perfect classification for our data. The Fitted Predictor curve plots the fitted model and the Random Predictor curve plots the results from using no model or by using a random guess (i.e. for x% of selected observations, x% of the total number of positive observations are expected to be correctly classified).

The Alternative Lift Chart plots Lift against % Cases. The Gain chart plots Gain Ratio against % Cases.

Lift Chart (Alternative) and Gain Chart for Training Partition



Lift Chart (Alternative) and Gain Chart for Validation Partition

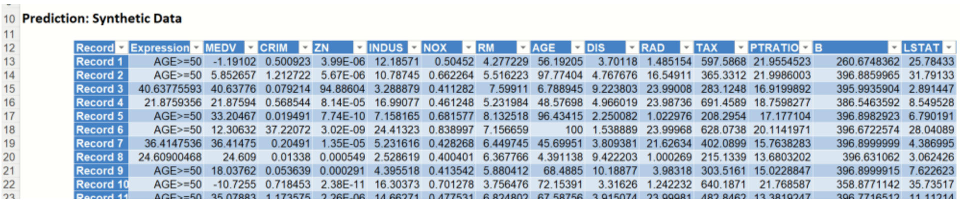


LinReg_Simulation

As discussed above, Analytic Solver Data Science generates a new output worksheet, LinReg_Simulation, when *Simulate Response Prediction* is selected on the Simulation tab of the Linear Regression dialog.

This report contains the synthetic data, the predicted values for the simulation, or synthetic, data, the predicted values for the training partition (using the fitted model) and the Excel – calculated Expression column, if populated in the dialog. A dialog may be opened to switch between the synthetic data, training data and the expression results, or a combination of two, as long as they are of the same type.

Synthetic Data



Prediction: Synthetic Data

Record	Expression	MEDV	CRIM	ZN	INDUS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
Record 1	AGE>=50	-1.19102	0.500923	3.99E-06	12.18571	0.50452	4.277229	56.19205	3.70118	1.485154	597.5868	21.9554523	260.6748362	25.78433
Record 2	AGE>=50	5.852657	1.212722	5.67E-06	10.78745	0.662264	5.516223	97.77404	4.767676	16.54911	365.3312	21.9986003	396.8859965	31.79133
Record 3	40.63775593	40.63776	0.079214	94.88604	3.288879	0.411282	7.59911	6.788945	9.223803	23.99008	283.1248	16.9199892	395.9935904	2.891447
Record 4	21.8759356	21.87594	0.568544	8.14E-05	16.99077	0.461248	5.231984	48.57698	4.966019	23.98736	691.4589	18.7598277	386.5463592	8.549528
Record 5	AGE>=50	33.20467	0.019491	7.74E-10	7.158165	0.681577	8.132518	96.43415	2.250082	1.022976	208.2954	17.177104	396.8982923	6.790191
Record 6	AGE>=50	12.30632	37.22072	3.02E-09	24.41323	0.838997	7.156659	100	1.538889	23.99968	628.0738	20.1141971	396.6722574	28.04089
Record 7	36.4147536	36.41475	0.20491	1.35E-05	5.231616	0.428268	6.449745	45.69951	3.809381	21.62634	402.0899	15.7638283	396.8999999	4.386995
Record 8	24.60900468	24.609	0.01338	0.000549	2.528619	0.400401	6.367766	4.391138	9.422203	1.000269	215.1339	13.6803202	396.631062	3.062426
Record 9	AGE>=50	18.03762	0.053639	0.000291	4.395518	0.413542	5.880412	68.4885	10.18877	3.98318	303.5161	15.0228847	396.8999915	7.622623
Record 10	AGE>=50	-10.7255	0.718453	2.38E-11	16.30373	0.701278	3.756476	72.15391	3.31626	1.242232	640.1871	21.768587	358.8771142	35.73517
Record 11	AGE>=50	35.07883	1.173575	2.26E-06	14.66271	0.477531	6.824802	67.58756	3.915074	23.99981	482.8462	13.3819247	396.7716512	11.11214

Note the first column in the output, Expression. This column was inserted into the Synthetic Data results because Calculate Expression was selected and an Excel function was entered into the Expression field, on the Simulation tab of the Linear Regression dialog

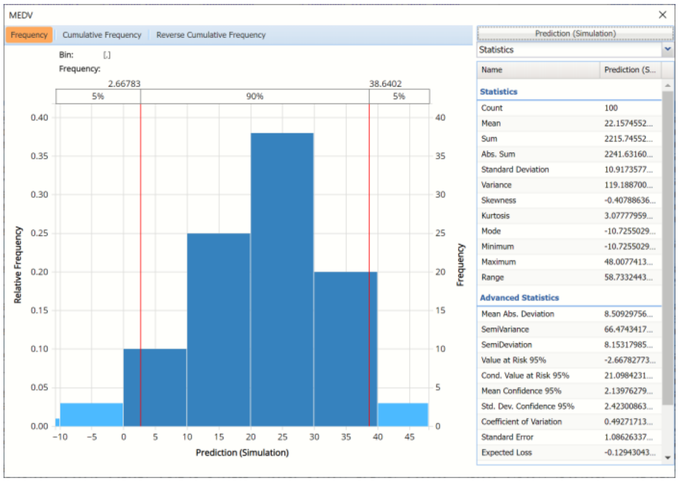
[@MEDV]/[@RM]

As a result, the values in this column will be equal to the MEDV variable value divided by the number of rooms.

The remainder of the data in this report is synthetic data, generated using the Generate Data feature described in the chapter with the same name, that appears in the Data Science Reference Guide.

The chart that is displayed once this tab is selected, contains frequency information pertaining to the predicted value for the MEDV variable in the synthetic data.

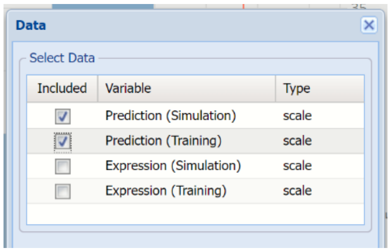
Frequency Chart for Prediction (Simulation) data



Name	Prediction (S...
Statistics	
Count	100
Mean	22.1574552...
Sum	2215.74552...
Abs. Sum	2241.63160...
Standard Deviation	10.9173577...
Variance	119.188700...
Skewness	-0.40788636...
Kurtosis	3.07777959...
Mode	-10.7255029...
Minimum	-10.7255029...
Maximum	48.0077413...
Range	58.7332443...
Advanced Statistics	
Mean Abs. Deviation	8.50929756...
SemiVariance	66.4743417...
SemiDeviation	8.15317985...
Value at Risk 95%	-2.66782773...
Cond. Value at Risk 95%	21.0984231...
Mean Confidence 95%	2.13976279...
Std. Dev. Confidence 95%	2.42300863...
Coefficient of Variation	0.49271713...
Standard Error	1.08626337...
Expected Loss	-0.12943043...

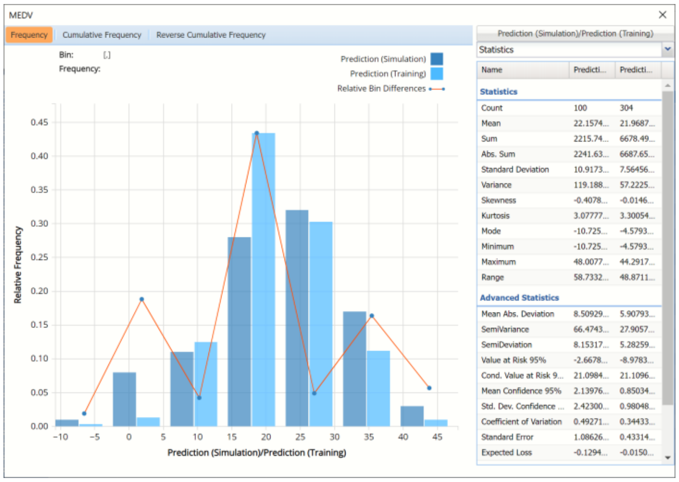
Click *Prediction (Simulation)* to add the Actual data to the chart.

Data dialog



Included	Variable	Type
☑	Prediction (Simulation)	scale
☑	Prediction (Training)	scale
☐	Expression (Simulation)	scale
☐	Expression (Training)	scale

Prediction (Simulation) and Prediction (Training) for MEDV variable



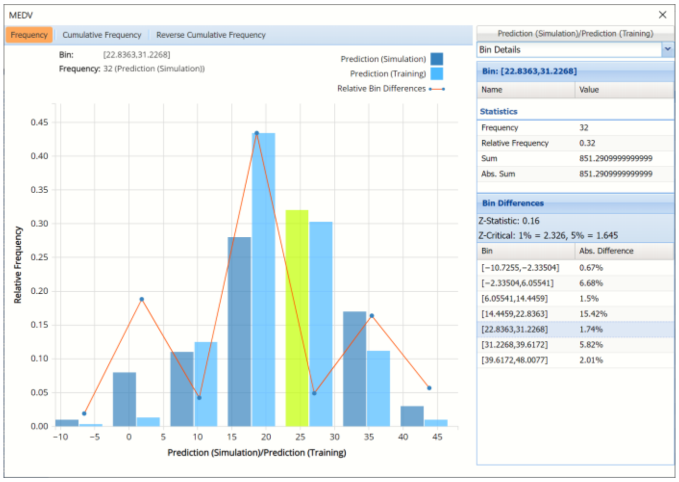
Bin: [.]
Frequency:

Prediction (Simulation)/Prediction (Training)

Statistics

Name	Predicti...	Predicti...
Statistics		
Count	100	304
Mean	22.1574...	21.9687...
Sum	2215.74...	6678.49...
Abs. Sum	2241.63...	6687.65...
Standard Deviation	10.9173...	7.56456...
Variance	119.188...	57.2225...
Skewness	-0.4078...	-0.0146...
Kurtosis	3.07777...	3.30054...
Mode	-10.725...	-4.5793...
Minimum	-10.725...	-4.5793...
Maximum	48.0077...	44.2917...
Range	58.7332...	48.8711...
Advanced Statistics		
Mean Abs. Deviation	8.50929...	5.90793...
SemiVariance	66.4743...	27.9057...
SemiDeviation	8.15317...	5.28259...
Value at Risk 95%	-2.6678...	-8.9783...
Cond. Value at Risk 9...	21.0984...	21.1096...
Mean Confidence 95%	2.13976...	0.85034...
Std. Dev. Confidence ...	2.42300...	0.98048...
Coefficient of Variation	0.49271...	0.34433...
Standard Error	1.08626...	0.43314...
Expected Loss	-0.1294...	-0.0150...

The Relative Bin Differences curve gives the absolute difference between the bins for each dataset. (Click the down arrow next to Frequency and select Bin Details to view.)



Bin: [22.8363,31.2268]
Frequency: 32 (Prediction (Simulation))

Prediction (Simulation)/Prediction (Training)

Bin Details

Bin: [22.8363,31.2268]

Name	Value
Statistics	
Frequency	32
Relative Frequency	0.32
Sum	851.2909999999999
Abs. Sum	851.2909999999999

Bin Differences

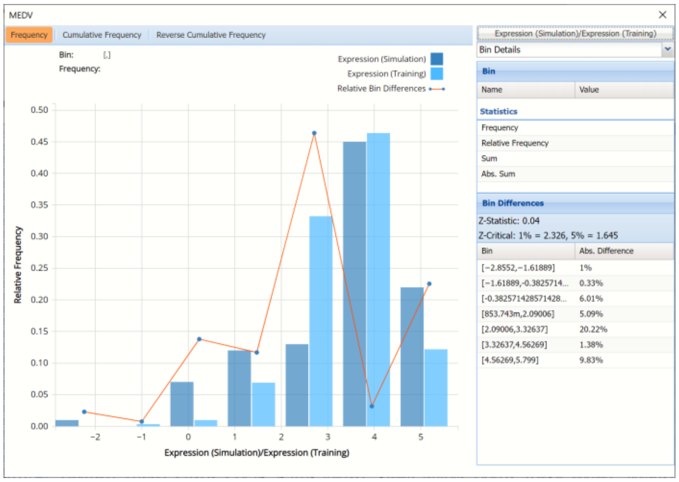
Z-Statistic: 0.16
Z-Critical: 1% = 2.326, 5% = 1.645

Bin	Abs. Difference
[−10.7255,−2.33504]	0.67%
[−2.33504,6.05541]	6.68%
[6.05541,14.4459]	1.5%
[14.4459,22.8363]	15.42%
[22.8363,31.2268]	1.74%
[31.2268,39.6172]	5.82%
[39.6172,48.0077]	2.01%

Click the down arrow next to Frequency to change the chart view to Relative Frequency or to change the look by clicking Chart Options. Statistics on the right of the chart dialog are discussed earlier in this section. For more information on the generated synthetic data, see the Generate Data chapter that appears earlier in this guide.

Click the Prediction (Simulation)/Prediction(Training) button to reopen the Data dialog. This time, select Expression (Simulation) and Expression (Training).

Expression Simulation vs Expression Training results



Bin

Name	Value

Statistics

Frequency	
Relative Frequency	
Sum	
Abs. Sum	

Bin Differences

Z-Statistic: 0.04

Z-Critical: 1% = 2.326, 5% = 1.645

Bin	Abs. Difference
[−2.8552,−1.61889]	1%
[−1.61889,-0.3825714...	0.33%
[-0.382571428571428...	6.01%
[853.743m,2.09006]	5.09%
[2.09006,3.32637]	20.22%
[3.32637,4.56269]	1.38%
[4.56269,5.799]	9.83%

This chart displays the predicted cost per room (MEDV/RM) for the Training partition as well as the synthetic data.

LinReg_Stored

For information on Stored Model Sheets, in this example *LinReg_Stored*, please refer to the “Scoring New Data” chapter that apperas later in this guide.

Scoring New Data

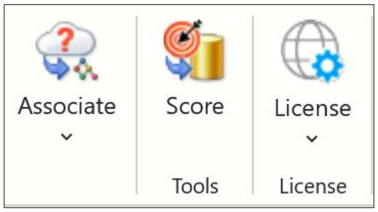
Introduction

Analytic Solver Data Science provides a method for scoring new data in a database or worksheet with any of the prediction, classification, forecasting or transformation features.

Scoring New Data

When Analytic Solver Data Science calculates prediction, classification, forecasting and transformation results, internal values and coefficients are generated and used in the computations. These values are saved to output sheets named, X_Stored -- where X is the abbreviated name of the data science method. For example, the name given to the stored model sheet for Linear Regression is "LinReg_Stored".

Note: In previous versions of XLMiner, this utility was a separate add-on application named XLMCalc. Starting with XLMiner V12, this utility is included free of charge and can be accessed under Score in the Tools section of the XLMiner ribbon.

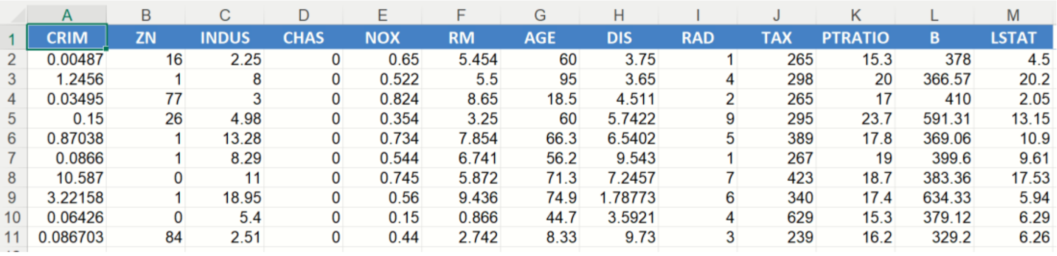


For example, assume the Linear Regression prediction method has just finished. The Stored Model Sheet (*LinReg_Stored*) will contain the regression equation in PMML format. When the Score Test Data utility is invoked or when the *PsiPredict()* function is present (see below), Analytic Solver will apply this equation from the Stored Model Sheet to the test data.

Along with values required to generate the output, the Stored Model Sheet also contains information associated with the input variables that were present in the training data. The dataset on which the scoring will be performed should contain *at least* these original Input variables. Analytic Solver Data Science offers a “matching” utility that will match the Input variables in the training set to the variables in the new dataset so the variable names are not required to be identical in both data sets (training and test). See the sections below for more information on scoring to a database.

Scoring New Data Example

This example illustrates how to score new data using a stored model sheet using output from a Multiple Linear Regression fitted model. In other words, the fitted Linear Regression model from the Predicting Housing Prices using MLR will be used to predict values for the MEDV variable for 10 *new* housing tracts. The new dataset may be found below. This procedure may be repeated using any stored model sheets generated using Analytic Solver Data Science.



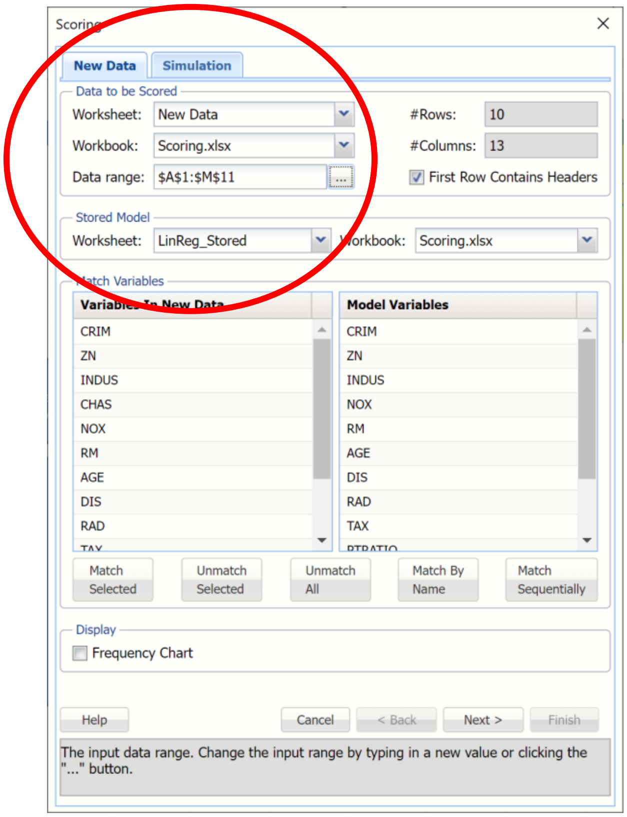
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
2	0.00487	16	2.25	0	0.65	5.454	60	3.75	1	265	15.3	378	4.5
3	1.2456	1	8	0	0.522	5.5	95	3.65	4	298	20	366.57	20.2
4	0.03495	77	3	0	0.824	8.65	18.5	4.511	2	265	17	410	2.05
5	0.15	26	4.98	0	0.354	3.25	60	5.7422	9	295	23.7	591.31	13.15
6	0.87038	1	13.28	0	0.734	7.854	66.3	6.5402	5	389	17.8	369.06	10.9
7	0.0866	1	8.29	0	0.544	6.741	56.2	9.543	1	267	19	399.6	9.61
8	10.587	0	11	0	0.745	5.872	71.3	7.2457	7	423	18.7	383.36	17.53
9	3.22158	1	18.95	0	0.56	9.436	74.9	1.78773	6	340	17.4	634.33	5.94
10	0.06426	0	5.4	0	0.15	0.866	44.7	3.5921	4	629	15.3	379.12	6.29
11	0.086703	84	2.51	0	0.44	2.742	8.33	9.73	3	239	16.2	329.2	6.26

Click **Help – Example Models** on the Data Science ribbon, then **Forecasting/Data Science Examples** and open the example file **Scoring.xlsx**.

LinReg_Stored was generated while performing the steps in the “Multiple Linear Regression Prediction Method” chapter. See this chapter for details on performing a Multiple Linear Regression.

Scoring.xlsx also contains a *New Data* worksheet with 10 new records. Our goal is to score this new dataset to come up with a predicted housing price for each of the 10 new records.

Click **Score** on the Data Science ribbon. Under *Data to be scored*, confirm that *New Data* appears as the Worksheet, Scoring.xlsx as the Workbook, the Data range is **A1:M11** and *LinReg_Stored* is selected in the *Worksheet* drop down menu under *Stored Model*.



Scoring

Data to be Scored

Worksheet: New Data

Workbook: Scoring.xlsx

Data range: $A$1:$M$11

#Rows: 10

#Columns: 13

☑ First Row Contains Headers

Stored Model

Worksheet: LinReg_Stored

Workbook: Scoring.xlsx

Match Variables

Variables In New Data	Model Variables
CRIM	CRIM
ZN	ZN
INDUS	INDUS
CHAS	NOX
NOX	RM
RM	AGE
AGE	DIS
DIS	RAD
RAD	TAX
TAX	PTRATIO

Display

□ Frequency Chart

The input data range. Change the input range by typing in a new value or clicking the "..." button.

Variables in the New Data may be matched with Variables in Stored Model using three easy techniques: by name, by sequence or manually.

If **Match By Name** is clicked, all similar named variables in the stored model sheet will be matched with similar named variables in the new dataset.

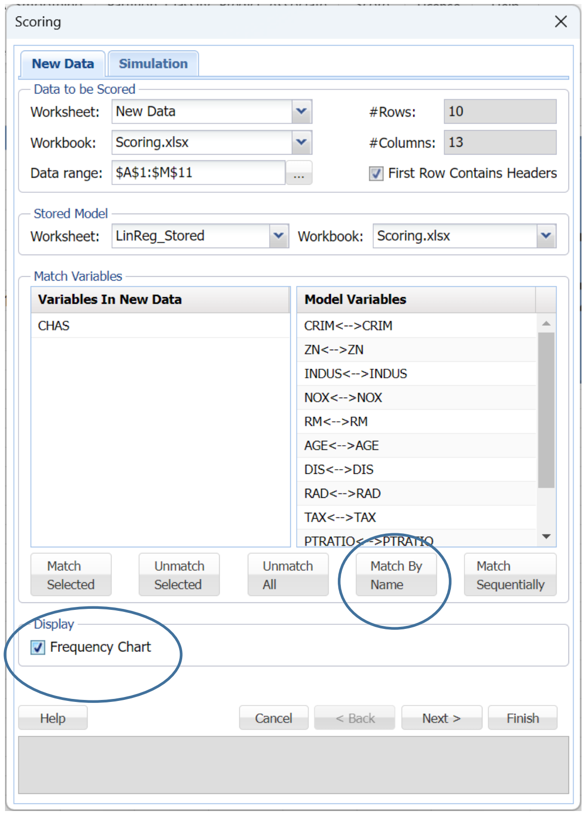
If **Match Sequentially** is clicked, the Variables in the stored model will be matched with the Variables in the new data in order that they appear in the two list boxes. For example, the variable *CRIM* from the new dataset will be matched with the variable *CRIM* from the stored model sheet, the variable *ZN* from the new data will be matched with the variable *ZN* from the stored model sheet and so on.

To manually map variables from the stored model sheet to the new data set, select a variable from the new data set in the *Variables in New Data* list box, then select the variable to be matched in the stored model sheet in the *Variables in Stored Model* list box, then click **Match**. For example to match the *CRIM* variable in the new dataset to the *CRIM* variable in the stored model sheet, select **CRIM** from the *Variables in New Data* list box, select **CRIM** from the stored model sheet in the *Variables in Stored Model* list box, then click **Match Selected** to match the two variables.

To unmatch all variables click **Unmatch all**. To unmatch two specific variables, select the matched variables, then click **Unmatch Selected**.

Click **Match By Name** to quickly match all variables in the fitted model with the variables in the new data.

Click the Frequency Chart checkbox, in the bottom left of the chart, to display frequency charts for all variables in the output.



Data to be Scored

Worksheet: New Data

Workbook: Scoring.xlsx

Data range: $A$1:$M$11

#Rows: 10

#Columns: 13

☑ First Row Contains Headers

Stored Model

Worksheet: LinReg_Stored

Workbook: Scoring.xlsx

Match Variables

Variables In New Data	Model Variables
CHAS	CRIM<-->CRIM
	ZN<-->ZN
	INDUS<-->INDUS
	NOX<-->NOX
	RM<-->RM
	AGE<-->AGE
	DIS<-->DIS
	RAD<-->RAD
	TAX<-->TAX
	PTRATIO<-->PTRATIO

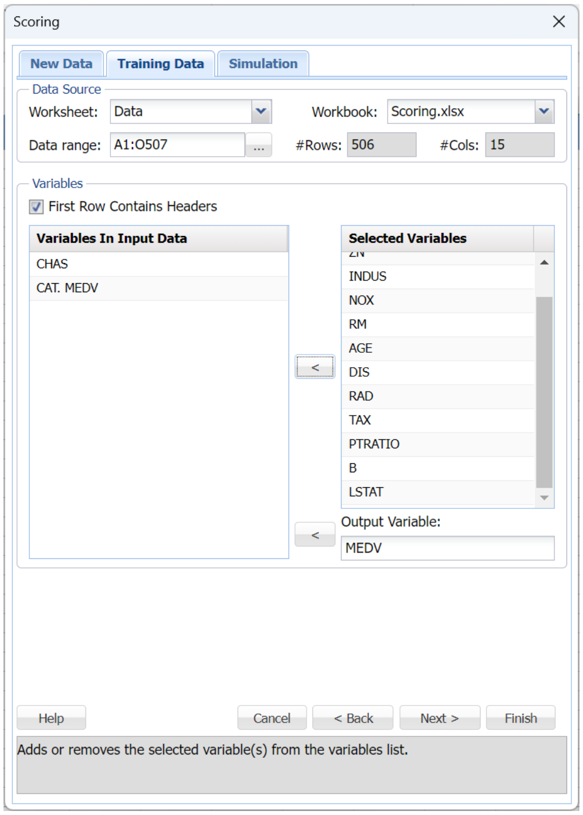
Display

☑ Frequency Chart

From here you can either click **Finish** to score the data on the New Data worksheet. Or, if you are scoring a classification or prediction model, you can click **Next** to advance to the Simulation tab where you can perform a complete risk analysis on the data.

Click **Next.**

Once the Simulate Response Prediction checkbox is selected, a new Training Data tab appears to the left of the Simulation tab. Use this tab to select the continuous variables to be included in the risk analysis. Make sure to select Worksheet: Data and Data range: A1:O507 within the Data Source section of the Training Data tab.



Data Source

Worksheet: Data
Workbook: Scoring.xlsx
Data range: A1:O507
#Rows: 506
#Cols: 15

Variables

☑ First Row Contains Headers

Variables In Input Data
CHAS
CAT. MEDV

Selected Variables
ZN
INDUS
NOX
RM
AGE
DIS
RAD
TAX
PTRATIO
B
LSTAT

Output Variable: MEDV

Adds or removes the selected variable(s) from the variables list.

The same variables must be selected as what were selected previously when the stored model sheet was created.

Click the LinReg_Output worksheet to see the variables selected for the original fitted model: CRIM, ZN, INDUS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B and LSTAT with MEDV as the output variable.

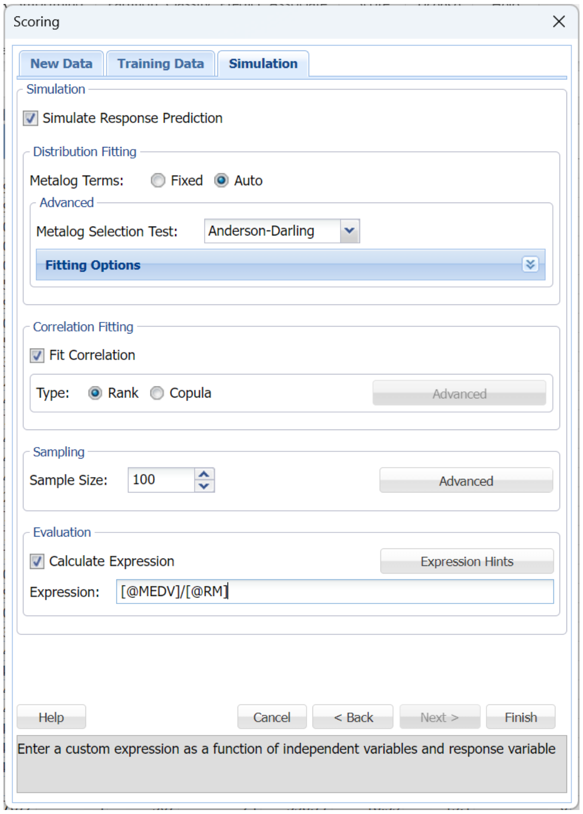
Click Next to return to the Simulation tab. Options for the risk analysis of the new data using the trained and validated machine learning model fit in the previous Predicting Housing Prices using MLR chapter, are set on the Simulation tab. This example uses the default of 100 simulated cases and enters an expression to calculate the median price per room in each census tract.

Enter the following formula for Expression: [@MEDV]/[@RM]

A chart with the results of this expression as applied to both the new data and the training data will be inserted into the output.

All options on the Simulation tab are left at their default settings. Recall that Fitting Options (click the down arrow to open) controls the automated fitting of Metalog probability distributions to each feature in the dataset. By default, bounds in the dataset are used as bounds for the family of Metalog distributions. However, users can easily and quickly adjust or remove the lower and/or upper bounds. Correlation Fitting will, by default, construct a rank-order correlation matrix that includes all features, but choosing “Copula” activates the “Advanced” button, where types of copulas (Clayton, Frank, Gumbel, etc) may be selected. For more information on each option, see the Generate Data chapter within the Data Science Reference Guide.

Click Finish to score the new data (from the New Data worksheet) and the synthetic data and to also perform a risk analysis on the original data (from the Data worksheet).



Simulation

☑ Simulate Response Prediction

Distribution Fitting

Metalog Terms: ○ Fixed ● Auto

Advanced

Metalog Selection Test: Anderson-Darling

Correlation Fitting

☑ Fit Correlation

Type: ● Rank ○ Copula

Sampling

Sample Size: 100

Evaluation

☑ Calculate Expression

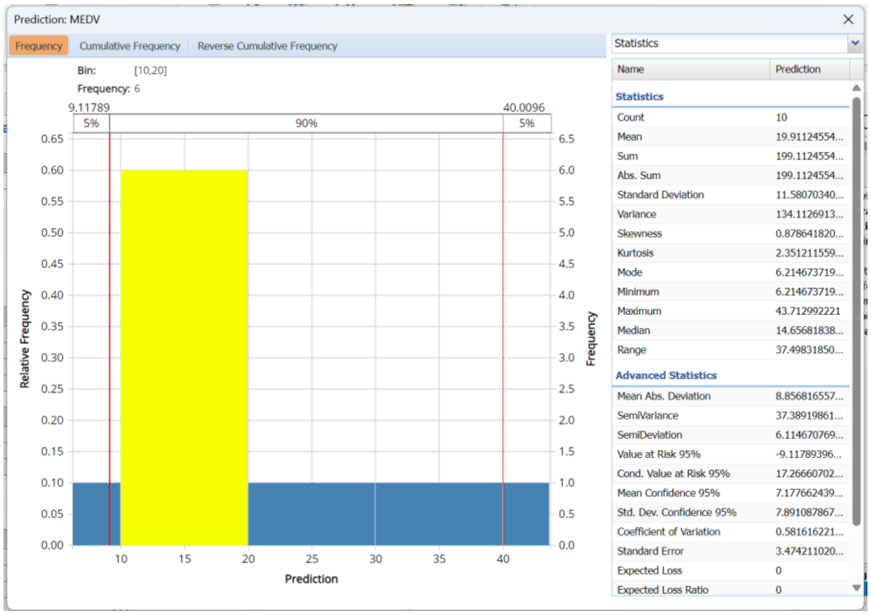
Expression: [@MEDV]/[@RM]

Enter a custom expression as a function of independent variables and response variable

Two new worksheets, Scoring_LinearRegression and Scoring_Simulation, are inserted to the right of the LinReg_Stored tab.

Scoring_LinearRegression

The first thing you'll notice is the frequency chart that appears when the tab opens. This chart displays the frequency of the output variable, MEDV, for the new data dataset. Use the mouse to hover over any of the bars in the graph to populate the Bin and Frequency headings at the top of the chart.



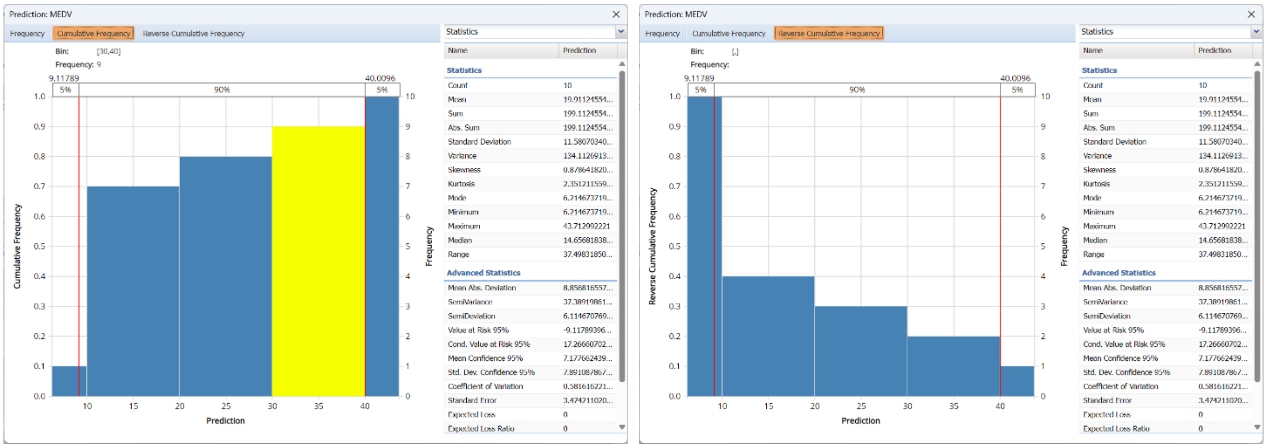
Name	Prediction
Statistics	
Count	10
Mean	19.91124554...
Sum	199.1124554...
Abs. Sum	199.1124554...
Standard Deviation	11.58070340...
Variance	134.1126913...
Skewness	0.878641820...
Kurtosis	2.351211559...
Mode	6.214673719...
Minimum	6.214673719...
Maximum	43.712992221
Median	14.65681838...
Range	37.49831850...
Advanced Statistics	
Mean Abs. Deviation	8.856816557...
SemiVariance	37.38919861...
SemiDeviation	6.114670769...
Value at Risk 95%	-9.11789396...
Cond. Value at Risk 95%	17.26660702...
Mean Confidence 95%	7.177662439...
Std. Dev. Confidence 95%	7.891087867...
Coefficient of Variation	0.581616221...
Standard Error	3.474211020...
Expected Loss	0
Expected Loss Ratio	0

From this chart, you can see the number of records contained in the new data plus Advanced and Summary statistics.

Red vertical lines appear at the 5% and 95% percentile values in all three charts (Frequency, Cumulative Frequency and Reverse Cumulative Frequency) effectively displaying the 90th confidence interval. The middle percentage is the percentage of all the variable values that lie within the 'included' area, i.e. the darker shaded area. The two percentages on each end are the percentage of all variable values that lie outside of the 'included' area or the "tails". Percentile values can be altered by moving either red vertical line to the left or right.

Click Cumulative Frequency and Reverse Cumulative Frequency tabs to see the Cumulative Frequency and Reverse Cumulative Frequency charts, respectively.

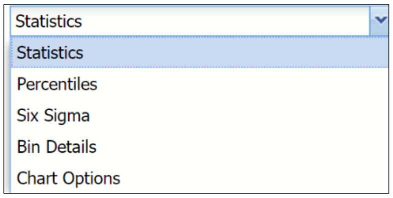
Prediction dialog shown with Cumulative Frequency and Reverse Cumulative Frequency charts



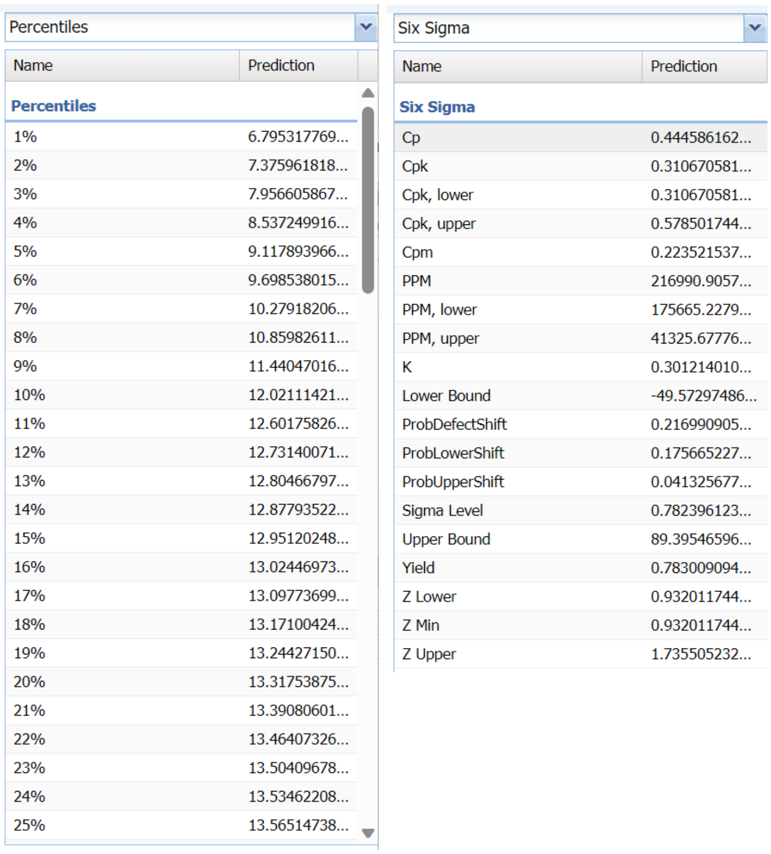
Name	Prediction
Statistics	
Count	10
Mean	19.91124554...
Sum	199.1124554...
Abs. Sum	199.1124554...
Standard Deviation	11.58070340...
Variance	134.1126913...
Skewness	0.878641820...
Kurtosis	2.351211559...
Mode	6.214673719...
Minimum	6.214673719...
Maximum	43.712992221
Median	14.65681838...
Range	37.49831850...
Advanced Statistics	
Mean Abs. Deviation	8.856816557...
SemiVariance	37.38919861...
SemiDeviation	6.114670769...
Value at Risk 95%	-9.11789396...
Cond. Value at Risk 95%	17.26660702...
Mean Confidence 95%	7.177662439...
Std. Dev. Confidence 95%	7.891087867...
Coefficient of Variation	0.581616221...
Standard Error	3.474211020...
Expected Loss	0
Expected Loss Ratio	0

Name	Prediction
Statistics	
Count	10
Mean	19.91124554...
Sum	199.1124554...
Abs. Sum	199.1124554...
Standard Deviation	11.58070340...
Variance	134.1126913...
Skewness	0.878641820...
Kurtosis	2.351211559...
Mode	6.214673719...
Minimum	6.214673719...
Maximum	43.712992221
Median	14.65681838...
Range	37.49831850...
Advanced Statistics	
Mean Abs. Deviation	8.856816557...
SemiVariance	37.38919861...
SemiDeviation	6.114670769...
Value at Risk 95%	-9.11789396...
Cond. Value at Risk 95%	17.26660702...
Mean Confidence 95%	7.177662439...
Std. Dev. Confidence 95%	7.891087867...
Coefficient of Variation	0.581616221...
Standard Error	3.474211020...
Expected Loss	0
Expected Loss Ratio	0

Click the down arrow next to Statistics to view Percentiles, Six Sigma metrics, details related to the histogram bins in the chart and also to change the look of the chart using Chart Options.



Percentiles and Six Sigma views



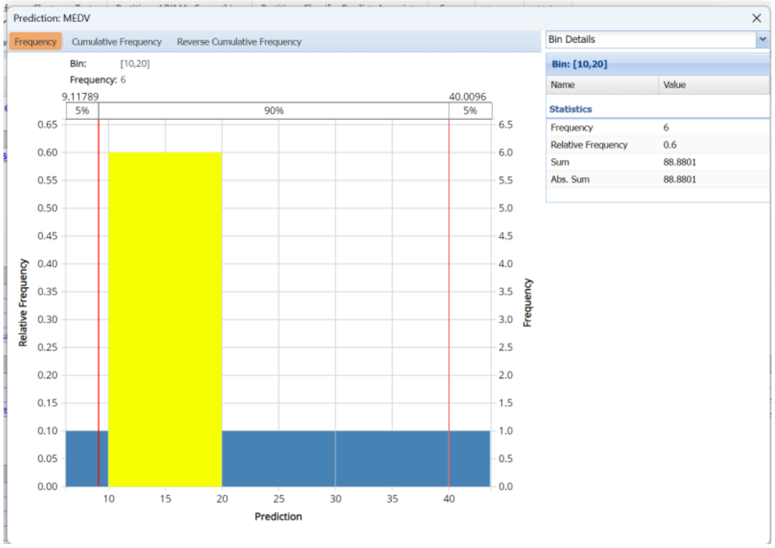
Name	Prediction
Percentiles	
1%	6.795317769...
2%	7.375961818...
3%	7.956605867...
4%	8.537249916...
5%	9.117893966...
6%	9.698538015...
7%	10.27918206...
8%	10.85982611...
9%	11.44047016...
10%	12.02111421...
11%	12.60175826...
12%	12.73140071...
13%	12.80466797...
14%	12.87793522...
15%	12.95120248...
16%	13.02446973...
17%	13.09773699...
18%	13.17100424...
19%	13.24427150...
20%	13.31753875...
21%	13.39080601...
22%	13.46407326...
23%	13.50409678...
24%	13.53462208...
25%	13.56514738...

Name	Prediction
Six Sigma	
Cp	0.444586162...
Cpk	0.310670581...
Cpk, lower	0.310670581...
Cpk, upper	0.578501744...
Cpm	0.223521537...
PPM	216990.9057...
PPM, lower	175665.2279...
PPM, upper	41325.67776...
K	0.301214010...
Lower Bound	-49.57297486...
ProbDefectShift	0.216990905...
ProbLowerShift	0.175665227...
ProbUpperShift	0.041325677...
Sigma Level	0.782396123...
Upper Bound	89.39546596...
Yield	0.783009094...
Z Lower	0.932011744...
Z Min	0.932011744...
Z Upper	1.735505232...

Select Bin Details to view details pertaining to each bin in the chart.

Hover over the bars in the chart to display important bin statistics such as frequency, relative frequency, sum and absolute sum.

Bin Details view



Bin: [10,20]

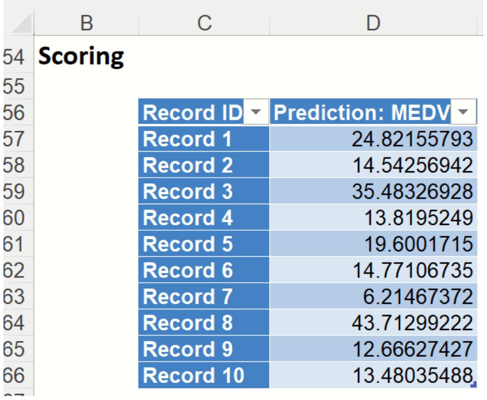
Name	Value
Statistics	
Frequency	6
Relative Frequency	0.6
Sum	88.8801
Abs. Sum	88.8801

- Frequency is the number of observations assigned to the bin.
- Relative Frequency is the number of observations assigned to the bin divided by the total number of observations.
- Sum is the sum of all observations assigned to the bin.
- Absolute Sum is the sum of the absolute value of all observations assigned to the bin, i.e. |observation 1| + |observation 2| + |observation 3| + …

As discussed above, see the Generate Data section of the Exploring Data chapter (within the Analytic Solver Reference Guide) for an in-depth discussion of this chart as well as descriptions of all statistics, percentiles, bin details and six sigma indices.

Click the X in the upper right hand corner to close the frequency chart. You can reopen the chart simply by clicking on a different worksheet tab and then clicking back to the Scoring_LinearRegression tab.

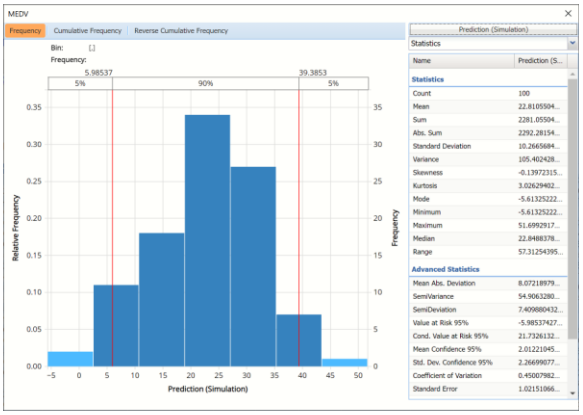
The results from scoring can be found under: Scoring. These are the mean predicted prices for the MEDV variable for each of the ten new records.



Scoring

Record ID	Prediction: MEDV
Record 1	24.82155793
Record 2	14.54256942
Record 3	35.48326928
Record 4	13.8195249
Record 5	19.6001715
Record 6	14.77106735
Record 7	6.21467372
Record 8	43.71299222
Record 9	12.66627427
Record 10	13.48035488

Click the 2nd output tab, Scoring_Simulation. You'll notice a similar chart appear. This chart displays a frequency histogram of the predicted values for the synthetic data.



Prediction (Simulation)

Name	Prediction (S...
Statistics	
Count	100
Mean	22.8105504...
Sum	2281.05504...
Abs. Sum	2292.28154...
Standard Deviation	10.2665684...
Variance	105.402428...
Skewness	-0.13972315...
Kurtosis	3.02629402...
Mode	-5.61325222...
Minimum	-5.61325222...
Maximum	51.6992917...
Median	22.8488378...
Range	57.31254395...
Advanced Statistics	
Mean Abs. Deviation	8.07218979...
SemiVariance	54.9063280...
SemiDeviation	7.409880432...
Value at Risk 95%	-5.98537427...
Cond. Value at Risk 95%	21.7326132...
Mean Confidence 95%	2.01221045...
Std. Dev. Confidence 95%	2.26699077...
Coefficient of Variation	0.45007982...
Standard Error	1.02151066...

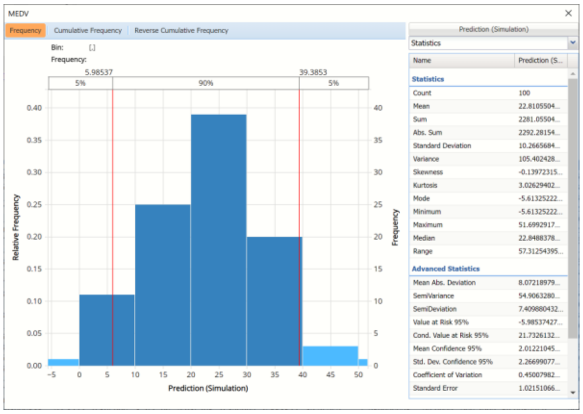
Scoring_Simulation

As discussed in the earlier chapters Automated Risk Analysis of ML Models and Fitting the Best Model and Predicting Housing Prices using MLR, Analytic Solver Data Science generates a new output worksheet, Scoring_Simulation, when *Simulate Response Prediction* is selected on the Simulation tab of the Scoring dialog.

From this chart, users can view charts containing histograms of the synthetic data, the training data and the results of the Expression applied to both datasets, if populated in the dialog.

This chart is similar to the chart on the Scoring_LinearRegression output sheet in that it also contains the same three charts described above (Frequency, Cumulative Frequency and Reverse Cumulative Frequency) and the same views: Statistics, Percentiles, Six Sigma and Bin Details. Click each chart tab to view the selected chart and click the down arrow next to Statistics to change the view.

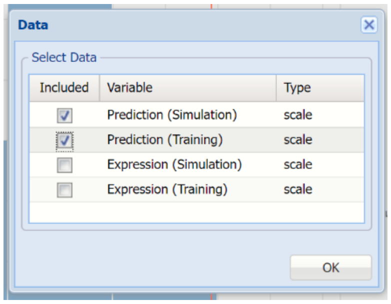
By default, this chart first opens to the Prediction (Simulation) view, which displays a histogram of the predicted values in the simulation (or synthetic) data.



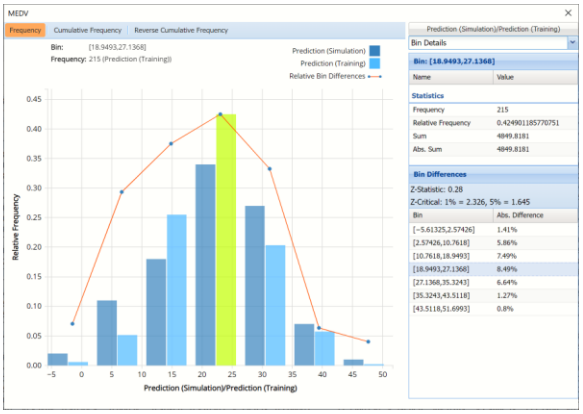
Prediction (Simulation)

Name	Prediction (S...
Statistics	
Count	100
Mean	22.8105504...
Sum	2281.05504...
Abs. Sum	2292.28154...
Standard Deviation	10.2665684...
Variance	105.402428...
Skewness	-0.13972315...
Kurtosis	3.02629402...
Mode	-5.61325222...
Minimum	-5.61325222...
Maximum	51.6992917...
Median	22.8488378...
Range	57.31254395...
Advanced Statistics	
Mean Abs. Deviation	8.07218979...
SemiVariance	54.9063280...
SemiDeviation	7.409880432...
Value at Risk 95%	-5.98537427...
Cond. Value at Risk 95%	21.7326132...
Mean Confidence 95%	2.01221045...
Std. Dev. Confidence 95%	2.26699077...
Coefficient of Variation	0.45007982...
Standard Error	1.02151066...

Click Prediction (Simulation) to open the Data dialog. Select Prediction (Simulation) and Prediction (Training) to view the predicted values in both the training and synthetic datasets.



Included	Variable	Type
☑	Prediction (Simulation)	scale
☑	Prediction (Training)	scale
☐	Expression (Simulation)	scale
☐	Expression (Training)	scale



Bin: [18.9493,27.1368]

Name	Value
Statistics	
Frequency	215
Relative Frequency	0.424901185770751
Sum	4849.8181
Abs. Sum	4849.8181

Bin Differences

Z-Statistic: 0.28

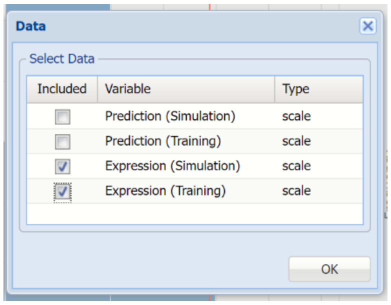
Z-Critical: 1% = 2.326, 5% = 1.645

Bin	Abs. Difference
[-5.61325,2.57426]	1.41%
[2.57426,10.7618]	5.86%
[10.7618,18.9493]	7.49%
[18.9493,27.1368]	8.49%
[27.1368,35.3243]	6.64%
[35.3243,43.5118]	1.27%
[43.5118,51.6993]	0.8%

The dark blue bars display the frequencies of the MEDV variable in the synthetic data, or Prediction (Simulation), and the lighter blue bars display the frequencies of the MEDV variable in the original data, or Prediction (Training). The synthetic data is predicting a slightly higher number of homes in the $27,000 to $35,000 (remember these are 1940's housing prices!) range and fewer number of homes in the $19,000 to $27,000 range.

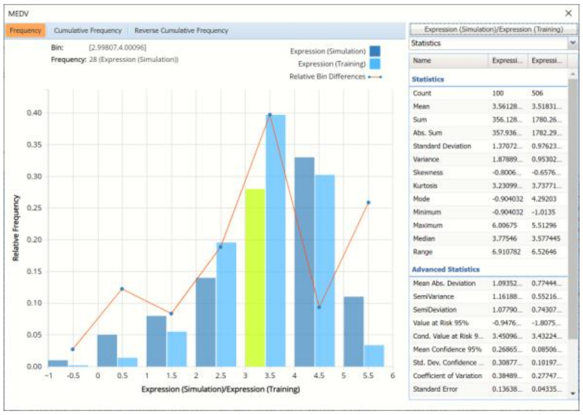
Notice that red curve which connects the relative Bin Differences for each bin. As discussed in the previous chapters, Bin Differences are computed based on the frequencies of records which predictions fall into each bin. For example, consider the highlighted bin in the screenshot above [x0, x1] = [18.949, 27.137]. This bin contains 34 records in the synthetic data and 215 records in the training data. The relative frequency of the Simulation data is 34/100 = 34% and the relative frequency of the Training data is 215/506 = 42.5%. Hence the Absolute Difference (in frequencies) is = 42.5 – 34 = 8.5%.

Click back to the Data dialog and select Expression (Simulation) and Expression (Training) to view the results of the expression for both datasets.



Included	Variable	Type
☐	Prediction (Simulation)	scale
☐	Prediction (Training)	scale
☑	Expression (Simulation)	scale
☑	Expression (Training)	scale

Recall the expression: [@MEDV]/[@RM] which is simply the median value of houses in each housing tract divided by the average number of rooms in each dwelling, or a rudimentary calculation of the price per room.

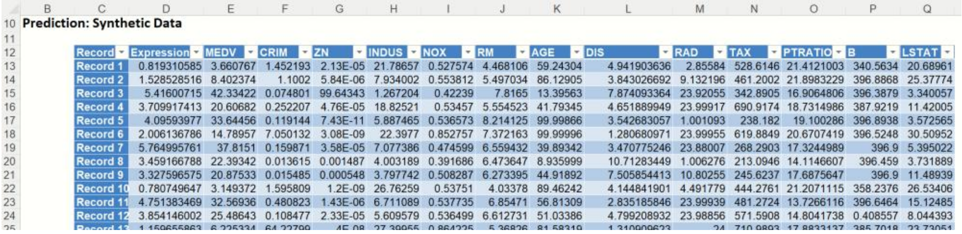


Bin: [2.99807,4.00096]
Frequency: 28 (Expression (Simulation))

Name	Expressi...	Expressi...
Statistics		
Count	100	506
Mean	3.56128...	3.51831...
Sum	356.128...	1780.26...
Abs. Sum	357.936...	1782.29...
Standard Deviation	1.37072...	0.97623...
Variance	1.87889...	0.95302...
Skewness	-0.8006...	-0.6576...
Kurtosis	3.23099...	3.73771...
Mode	-0.904032	4.29203
Minimum	-0.904032	-1.0135
Maximum	6.00675	5.51296
Median	3.77546	3.577445
Range	6.910782	6.52646
Advanced Statistics		
Mean Abs. Deviation	1.09352...	0.77444...
SemiVariance	1.16188...	0.55216...
SemiDeviation	1.07790...	0.74307...
Value at Risk 95%	-0.9476...	-1.8075...
Cond. Value at Risk 9...	3.45096...	3.43224...
Mean Confidence 95%	0.26865...	0.08506...
Std. Dev. Confidence ...	0.30877...	0.10197...
Coefficient of Variation	0.38489...	0.27747...
Standard Error	0.13638...	0.04335...

The highlighted bar above shows 28 records in the synthetic data are contained in the Bin: [2.998, 4.00] meaning that there are 28 records in the synthetic data where the value of the expression falls within this interval.

Click the X in the upper right hand corner to close the dialog to view the simulated data for each variable. (To reopen, simply click another tab and then click back to the Scoring_Simulation tab.)



Prediction: Synthetic Data

Record	Expression	MEDV	CRIM	ZN	INDUS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
Record 1	0.819310585	3.660767	1.452193	2.13E-05	21.78657	0.527574	4.468106	59.24304	4.941903636	2.85584	528.6146	21.4121003	340.5634	20.68961
Record 2	1.528528516	8.402374	1.1002	5.84E-06	7.934002	0.553812	5.497034	86.12905	3.843026692	9.132196	461.2002	21.8983229	396.8868	25.37774
Record 3	5.41600715	42.33422	0.074801	99.64343	1.267204	0.42239	7.8165	13.39563	7.874093364	23.92055	342.8905	16.9064806	396.3879	3.340057
Record 4	3.709917413	20.60682	0.252207	4.76E-05	18.82521	0.53457	5.554523	41.79345	4.651889949	23.99917	690.9174	18.7314986	387.9219	11.42005
Record 5	4.09593977	33.64456	0.119144	7.43E-11	5.887465	0.536573	8.214125	99.99866	3.542683057	1.001093	238.182	19.100286	396.8938	3.572565
Record 6	2.006136786	14.78957	7.050132	3.08E-09	22.3977	0.852757	7.372163	99.99996	1.280680971	23.99955	619.8849	20.6707419	396.5248	30.50952
Record 7	5.764995761	37.8151	0.159871	3.58E-05	7.077386	0.474599	6.559432	39.89342	3.470775246	23.88007	268.2903	17.3244989	396.9	5.395022
Record 8	3.459166788	22.39342	0.013615	0.001487	4.003189	0.391686	6.473647	8.935999	10.71283449	1.006276	213.0946	14.1146607	396.459	3.731889
Record 9	3.327596575	20.87533	0.015485	0.000548	3.797742	0.508287	6.273395	44.91892	7.505854413	10.80255	245.6237	17.6875647	396.9	11.48939
Record 10	0.780749647	3.149372	1.595809	1.2E-09	26.76259	0.53751	4.03378	89.46242	4.144841901	4.491779	444.2761	21.2071115	358.2376	26.53406
Record 11	4.751383469	32.56936	0.480823	1.43E-06	6.711089	0.537735	6.85471	56.81309	2.835185846	23.99939	481.2724	13.7266116	396.6464	15.12485
Record 12	3.854146002	25.48643	0.108477	2.33E-05	5.609579	0.536499	6.612731	51.03386	4.799208932	23.98856	571.5908	14.8041738	0.408557	8.044393
Record 13	1.159655863	6.225334	64.22799	4E-08	27.39955	0.864225	5.36826	81.58319	1.310909623	24	710.9893	17.8833137	385.7018	23.73051

Using Data Science Psi Functions in Excel

Analytic Solver Data Science utilizes XML PMML format to store the supported models and use them for "scoring" (classifying, predicting, forecasting, transforming) new data using four new generic scoring functions: **PsiPredict()**, **PsiForecast()**, **PsiTransform()** and **PsiPosteriors().** PsiPredict() and PsiForecast() provide previously available functionality plus new additional functionality such as storing and scoring ensemble models with any available weak learner and computing the fitted values for new time series data. PsiTransform() and PsiPosteriors() provide new functionality and the availability of new models for storing or scoring.

PSI Scoring Function	Description
PsiPredict()	Predicts the target for input data using a Classification or Regression model and computes the fitted values for a Time Series model stored in PMML format.
PsiForecast()	Computes the forecasts for the input data using a Time Series model stored in PMML format.
PsiPosteriors()	Computes the posterior probabilities for the input data

	using a Classification model stored in PMML format.
PsiTransform()	Transforms the input data using a Transformation model stored in PMML format.

It's possible to score data with a prediction or classification method or perform a time series forecast manually (without the need to click the Score icon on the ribbon) by entering a Psi Solver function into an Excel cell as an array.

Scoring Data Using Psi Functions

Using the example above, click over to the *PsiPredict* worksheet and select a blank cell on the worksheet. Enter "=PsiPredict(LinReg_Stored!B12:B69,'New Data'!A1:L11)". The formula result will "spill" into the cells below.

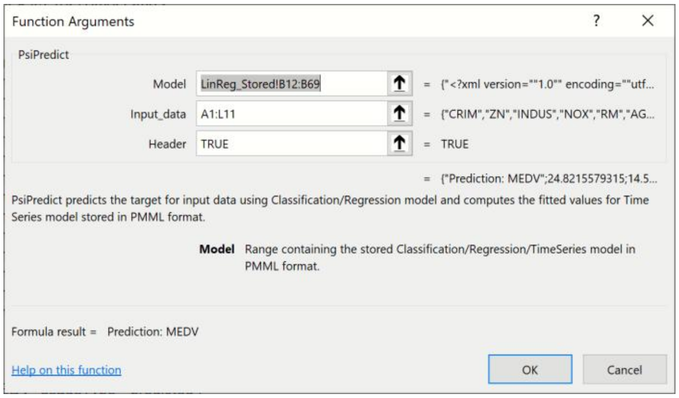
Note: If using a version of Excel that does not support Dynamic Arrays, select cells N1:N11, enter the formula and press SHIFT + CTRL + ENTER to enter the formula as an array into all 10 cells (N2:N11).

The first argument, LinReg_Stored!B12:B69, is the range of cells used by Analytic Solver Data Science to store the linear regression model on the *LinReg_Stored* worksheet. Clearly this data range will change as the classification or prediction method changes and as the number of features included in the dataset changes.

The second argument, New Data!A1:L11, is the range containing the new data on the *New Data* worksheet. The new data must contain at least one row of data containing the same number of features (or columns) as the data used to create the model. In this example, we included 12 features in the linear regression model: CRIM, ZN, INDUS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B & LSTAT. As a result, our new data also containes these same 12 features. We could have performed this prediction on only one row of new data, NewData!A2:L2, but choose to use all 10 rows.

Note: If there are new or missing features in the new dataset, the Psi Data Science function will return #VALUE.

It's also possible to enter this formula using the Insert Function dialog by clicking Formulas – Insert Function, select PSI Data Science for Category, then PsiPredictMLR.



PsiPredict

Model LinReg_Stored!B12:B69 = {"<?xml version=""1.0"" encoding=""utf...

Input_data A1:L11 = {"CRIM","ZN","INDUS","NOX","RM","AG...

Header TRUE = TRUE

= {"Prediction: MEDV";24.8215579315;14.5...

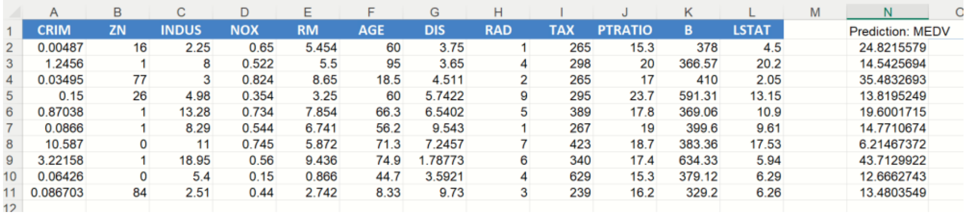
PsiPredict predicts the target for input data using Classification/Regression model and computes the fitted values for Time Series model stored in PMML format.

Model Range containing the stored Classification/Regression/TimeSeries model in PMML format.

Formula result = Prediction: MEDV

Note: The Insert Function dialog is not supported when using Data Science Cloud. To use this function or any other Psi function, simply type the function directly into the cell.

The scoring results are shown in the screenshot below in the N column.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	CRIM	ZN	INDUS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT		Prediction: MEDV
2	0.00487	16	2.25	0.65	5.454	60	3.75	1	265	15.3	378	4.5		24.8215579
3	1.2456	1	8	0.522	5.5	95	3.65	4	298	20	366.57	20.2		14.5425694
4	0.03495	77	3	0.824	8.65	18.5	4.511	2	265	17	410	2.05		35.4832693
5	0.15	26	4.98	0.354	3.25	60	5.7422	9	295	23.7	591.31	13.15		13.8195249
6	0.87038	1	13.28	0.734	7.854	66.3	6.5402	5	389	17.8	369.06	10.9		19.6001715
7	0.0866	1	8.29	0.544	6.741	56.2	9.543	1	267	19	399.6	9.61		14.7710674
8	10.587	0	11	0.745	5.872	71.3	7.2457	7	423	18.7	383.36	17.53		6.21467372
9	3.22158	1	18.95	0.56	9.436	74.9	1.78773	6	340	17.4	634.33	5.94		43.7129922
10	0.06426	0	5.4	0.15	0.866	44.7	3.5921	4	629	15.3	379.12	6.29		12.6662743
11	0.086703	84	2.51	0.44	2.742	8.33	9.73	3	239	16.2	329.2	6.26		13.4803549
12														

The PsiPredict() function is interactive meaning that if a variable value is changed, for example the first LSTAT value in cell L2 changes from 4.5 to 9.5, the Predicted Value in cell N2 will immediately update to reflect a new predicted value.

The remaining PSI Data Science functions can be used in the same way using models from their respective stored model sheets. See below for specifications for PsiPredict(), PsiPosteriors() and PsiTransform(). See the section below for information on PsiForecast().

PsiPredict()

`PsiPredict(Model, Input_Data, [Header])`

Predicts the response, target, output or dependent variable for `Input_Data` whether it is continuous (Regression) or categorical (Classification) when the model is stored in PMML format. In addition, this function also computes the fitted values for a Time Series model when the model is stored in PMML format.

`Model`: Range containing the stored Classification, Regression or TimeSeries model in PMML format.

`Input_Data`: Range containing the new data for computing predictions. Range must contain a header row with column names and at least one row of data containing the exact same features (or columns) as the data used to create the model.

`Header`: If True, a heading will be inserted above the forecasted values. If omitted or false, a heading will not appear.

The contents of the Dynamic Array will "spill" down the column. If a nonblank cell is "blocking" the contents of the Dynamic Array, PsiForecast() will return #SPILL until such time as the blockage is removed. Use the optional *numForecasts* argument to specify the number of forecasts in the Dynamic Array. If not present, one forecast will be returned.

Output: A single column containing the header (if the header argument is set to TRUE) and predicted/fitted values for each record in `Input_Data`.

To know if the result of the prediction is continuous or categorical, you must know what kind of model you are passing as an argument to the scoring function – if you previously fitted the classification model and are now predicting the new feature vectors, you should expect to get the compatible categorical response. On the other hand, you should expect the continuous response from the new data prediction when using a fitted regression model. In previous versions, the user was expected to know the exact type model, such as Mulitple Linear Regression or Discriminant Analysis, to know what kind of output will be produced, whereas in V2017 and later, it is sufficient to know whether you're pointing to a classification or regression model in order to determine the type of the response. Note: If the user intends to use an "unknown" model for scoring, the stored worksheets contain the complete information about the model including several clear indications of the model type and data dictionaries with the types of features and response.

In addition, PsiPredict() can compute the *fitted values* for the new time series based on the provided Time Series model. Unlike future-looking forecasting, provided by PsiForecast(), PsiPredict() computes a model *prediction* for each observation in the provided new time series.

Supported Models:

- Classification:
  - Linear Discriminant Analysis
  - Logistic Regression
  - K-Nearest Neighbors
  - Classification Tree
  - Naïve Bayes
  - Neural Network
  - Random Trees
  - Bagging (with any supported weak learner)
  - Boosting (with any supported weak learner)
- Regression:
  - Logistic Regression
  - K-Nearest Neighbors
  - Neural Network
  - Bagging (with any supported weak learner)
  - Boosting (with any supported weak learner)
- Time Series (fitted values)
  - ARIMA
  - Exponential Smoothing
  - Double Exponential Smoothing
  - Holt-Winters Smoothing

Previous related Psi Scoring functions:

- Classification: PsiClassifyLR, PsiClassifyDA, PsiClassifyCT, PsiClassifyNB, PsiClassifyNN, PsiClassifyCTEnsemble, PsiClassifyNNEnsemble
- Regression: PsiPredictMLR, PsiPredictRT, PsiPredictNN, PsiPredictNNEnsemble, PsiPredictRTEnsemble

Prediction/Classification/Time Series Algorithm	**Stored Model Sheet**
Linear Discriminant Analysis Classification	DA_Stored
Logistic Regression Classification	LogReg_Stored
k-Nearest Neighbors Classification	KNNC_Stored
Classification Trees	CT_Stored
Naïve Bayes Classification	NB_Stored
Neural Networks Classification	NNC_Stored
Ensemble Methods for Classification	CBoosting_Stored CBagging_Stored CRandTrees_Stored

Linear Regression	LinReq_Stored
k-Nearest Neighbors Regression	KNNP_Stored
Regression Tree	RT_Stored
Neural Network Regression	NNP_Stored
Ensemble Methods for Regression	RBoosting_Stored RBagging_Stored RRandTrees_Stored
ARIMA	ARIMA_Stored
Exponential Smoothing	Expo_Stored
Double Exponential Smoothing	DoubleExpo_Stored
Moving Average Smoothing	MovingAvg_Stored
Holt Winters Smoothing	MultHoltWinters_Stored AddHoltWinters_Stored NoTrendHoltWinters_Stored

PsiPosteriors()

`PsiPosteriors(Model, Input_Data, [Header])`

Computes the posterior probabilities for `Input_Data` using a Classification model stored in PMML format.

`Model`: Range containing the stored Classification model in PMML format.

`Input_Data`: Range containing the new data for computing posterior probabilities. Range must contain a header with column names and at least one row of data containing the exact same features (or columns) as the data used to create the model.

`Header`: If True, a heading is inserted in the output above the forecasted values. If False or omitted, a heading is not inserted into the output.

In Data Science Cloud and in new versions of desktop Excel, PsiPosterior() returns a Dynamic Array (see Note in section heading, above) To use this function in the Cloud, you need only enter the Psi function in one cell as a normal function, i.e., not as a control array. The contents of the Dynamic Array will "spill" down the column. If a nonblank cell is "blocking" the contents of the Dynamic Array, PsiForecast() will return #SPILL until such time as the blockage is removed. Use the optional *numForecasts* argument to specify the number of forecasts in the Dynamic Array. If not present, one forecast will be returned.

Output: Multiple columns containing a header with class labels and estimated posterior probabilities for each class label for all records in `Input_Data`.

Supported Models:

- Classification:
  - Linear Discriminant Analysis
  - Logistic Regression

- K-Nearest Neighbors
- Classification Tree
- Naïve Bayes
- Neural Network
- Random Trees
- Bagging (with any supported weak learner)
- Boosting (with any supported weak learner)

Previous related Psi Scoring functions: N/A

Classification Algorithm	Stored Model Sheet
Linear Discriminant Analysis Classification	DA_Stored
Logistic Regression Classification	LogReg_Stored
k-Nearest Neighbors Classification	KNNC_Stored
Classification Trees	CT_Stored
Naïve Bayes Classification	NB_Stored
Neural Networks Classification	NNC_Stored
Ensemble Methods for Classification	CBoosting_Stored CBagging_Stored CRandTrees_Stored

PsiTransform()

`PsiTransform(Model, Input_Data, [Header])`

Transforms the `Input_Data` using a Transformation model stored in PMML format.

`Model`: Range containing the stored Transformation model in PMML format.

`Input_Data`: Range containing the new data for transformation. Range must contain a header with column names and at least one row of data containing the exact same features (or columns) as the data used to create the model.

`Header`: If True, a heading is inserted above the forecasted values. If False or omitted, a heading is not inserted.

In Data Science Cloud and in newer versions of desktop Excel, PsiTransform() returns a Dynamic Array (see Note in section heading, above). To use this function in the Cloud, you need only enter the Psi function in one cell as a normal function, i.e., not as a control array. The contents of the Dynamic Array will "spill" down the column. If a nonblank cell is "blocking" the contents of the Dynamic Array, PsiForecast() will return #SPILL until such time as the blockage is removed. Use the optional *numForecasts* argument to specify the number of forecasts in the Dynamic Array. If not present, one forecast will be returned.

Output*:* One or multiple columns containing a header and transformed data.

Supported Models:

- Transformation:
  - Rescaling
- Text Science

- TF-IDF Vectorization (input data – text variable with the corpus of documents)
- LSA Concept Extraction (input data – term-document matrix, where columns represent terms and rows represent documents)

Previous related Psi Scoring functions: N/A

Algorithm	Stored Model Sheet
Rescaling	Rescaling_Stored
Text Science	TFIDF_Stored LSA_Stored

Time Series Forecasting

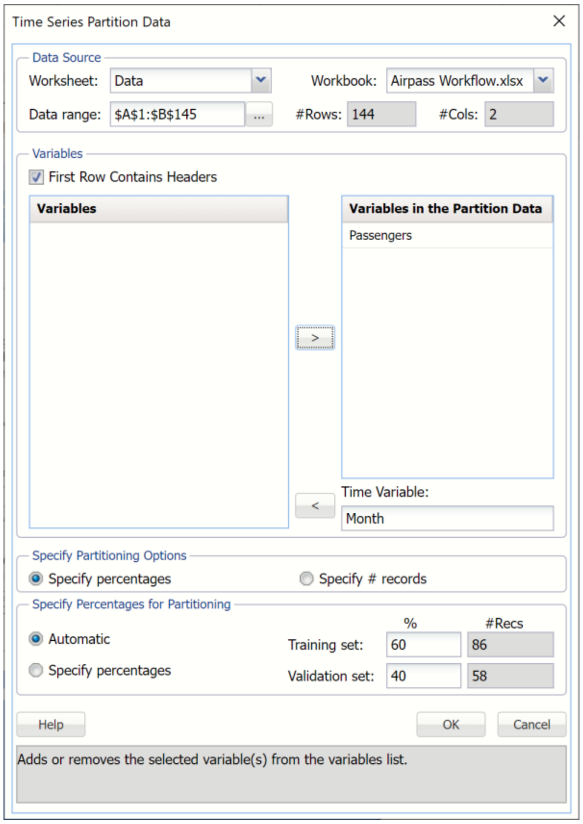
Starting with version 2014-R2, Analytic Solver Data Science includes the ability to forecast a future point in a time series in one of your spreadsheet formulas (without using the Score button on the Ribbon) using a PsiForecast() function in conjunction with a model created using ARIMA or one of our smoothing methods (Exponential, Double Exponential, Moving Average, or Holt Winters).

PsiForecast() is similar to the previous PSIForecastXXX functions supported in V2014, 2015, and 2016: it will compute future-looking forecasts based on the fitted model, using the provided new time series observations as initial points. The result of PsiForecast() is deterministic when the Simulate argument is set to FALSE and non-deterministic when the Simulate argument is set to TRUE. When Simulate is TRUE, the forecasts are adjusted with random errors that follow a normal distribution, based on the forecasts' statistics.

Open the Airpass.xlsx example dataset by clicking Help – Examples on the Data Science ribbon, then clicking Forecasting/Data Science Examples. This example dataset includes International Airline Passenger Information by month for years 1949 – 1960. Since the number of airline passengers increases during certain times of the year, for example Spring, Summer, and in the month of December, we can say that this dataset includes "seasonality".

First, we will partition this dataset into two datasets: a training dataset and a validation dataset. We'll use the training dataset to create the ARIMA model and then we'll apply the model to the validation dataset to forecast six future data points, or one half year of data.

Click Partition in the Time Series section of the Data Science ribbon to open the Time Series Partition Data dialog. Select Passengers for the Variables in the Partition Data and Month for the Time Variable.


Time Series Partition Data

Data Source

Worksheet: Data

Workbook: Airpass Workflow.xlsx

Data range: $A$1:$B$145

#Rows: 144

#Cols: 2

Variables

☑ First Row Contains Headers

Variables	Variables in the Partition Data
	Passengers

Time Variable: Month

Specify Partitioning Options

● Specify percentages ○ Specify # records

Specify Percentages for Partitioning

● Automatic

○ Specify percentages

	%	#Recs
Training set:	60	86
Validation set:	40	58

Adds or removes the selected variable(s) from the variables list.

Click **OK** to accept the defaults for Specify Partitioning Options and Specify Percentages for Partitioning. Recall that when a time series dataset is partitioned, the dataset is partitioned sequentially. Therefore, 60% or the first 86 records, will be assigned to the training dataset and the remaining 40%, or 58 records, will be assigned to the validation dataset. (For more information on partitioning a time series dataset, see the previous chapter Exploring a Time Series Dataset.)

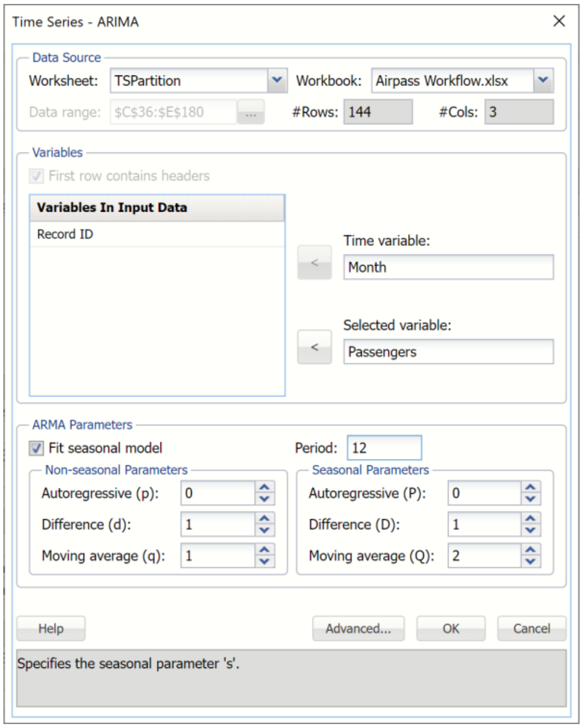
The TSPartition worksheet will be inserted into the Model tab of the Analytic Solver task pane under Transformations – Time Series Partition. Recall the steps needed to produce the forecast. Click ARIMA -- ARIMA to open the ARIMA dialog. Month has been pre selected as the Time variable. Select Passengers as the Selected variable.

This example will use a SARIMA model, or Seasonal Autoregressive Integrated Moving Average model, to predict the next six datapoints in the dataset. (For more information on this type of time series model, please see the earlier chapter, “Exploring a Time Series Dataset.”) A seasonal ARIMA model requires 7 parameters, 3 nonseasonal (autoregressive (p), integrated (d), and moving average (q)), 3 seasonal (autoregressive (P), integrated (D), and moving average (Q)), and period. Each parameter must be a non-negative integer.

Selecting appropriate values for p, d, q, P, D, Q and period is beyond the scope of this User Guide. Consequently, this example will use a well documented SARIMA model with parameters p = 0, d = 1, q = 1, P = 0, D = 1, Q = 2 and period (P) = 12. Please refer to the classic time series analysis text **Time Series Analysis: Forecasting and Control** written by George Box and Gwilym Jenkins for more information on parameter selection.

Select Fit seasonal model and enter 12 for Period since it takes a full 12 months for the seasonal pattern to repeat. Set the Non-seasonal Parameters as

Autoregressive (p) = 0, Difference (d) = 1, Moving Average (q) = 1 and the Seasonal Parameters as Autoregressive (P) = 0, Difference (D) = 1, and Moving Average (Q) = 2.



Time Series - ARIMA

Data Source

Worksheet: TSPartition

Workbook: Airpass Workflow.xlsx

Data range: $C$36:$E$180

#Rows: 144

#Cols: 3

Variables

☑ First row contains headers

Variables In Input Data

Record ID

Time variable: Month

Selected variable: Passengers

ARMA Parameters

☑ Fit seasonal model

Period: 12

Non-seasonal Parameters

Autoregressive (p): 0

Difference (d): 1

Moving average (q): 1

Seasonal Parameters

Autoregressive (P): 0

Difference (D): 1

Moving average (Q): 2

Specifies the seasonal parameter 's'.

Click **OK** to create the SARIMA model.

ARIMA_Output will be inserted into the Model tab of the task pane under Reports – ARIMA. This output contains the Training Error Measures and Fitted Model Statistics. (For more information on this report, please see the chapter Exploring a Time Series Dataset within the Analytic Solver Data Science Reference Guide.) ARIMA_Stored contains the stored model parameters.

Now we'll use this ARIMA model to predict new data points in the validation dataset using the PsiForecast() function. When array-entered into seven different Excel cells, this function will forecast six different future points in the dataset. (*Note: The first forecasted point will be more accurate than the second, the second forecasted point more accurate than the third and so on.)* The PsiForecast() function will be interactive in the sense that if any of the input values (values passed in the 2nd argument) change, the forecast will be recomputed.

The PsiForecastARIMA function takes five arguments but two are optional: Model, Input Data, Simulate, Num_forecasts, and Header. Select a blank cell on the Data worksheet and enter **=PsiForecast(**.

Recall that a seasonal ARIMA model requires 7 parameters, 3 nonseasonal (autoregressive (p), integrated (d), and moving average (q)), 3 seasonal (autoregressive (P), integrated (D), and moving average (Q)), and period. Each parameter must be a non-negative integer.

The first argument, *Model*, is the range of cells used by Analytic Solver Data Science to store the ARIMA model on the *ARIMA_Stored* worksheet. This data range will change as the forecast method changes. Select or enter **ARIMA_Stored!B12:B38,** for this argument.

The second argument, *Input_Data*, is the range containing the initial starting points from the validation data set. The minimum number of initial points that should be specified for a seasonal ARIMA model is the larger of p + d + s * (P + D) and q + s * Q. In this example, p + d + s * (P + D) is equal to 13 (0 + 1 + 12 * (0 + 1) and q + s * Q is equal to 13 (1 + 12 * 1), therefore the minimum number of initial starting points required is 13 (MAX (13, 13)). If you provide fewer than the minimum required number of starting points, PsiForecast() will return a column of zeros. (See the table below for the minimum number of initial starting points required by each Forecasting method included in Analytic Solver Data Science.) The maximum number of starting points is the number of points in the validation dataset. All points supplied in the second argument will be used in the forecast. Select or enter **Data!B1:B145,** for this argument.

Pass True or False for the third argument. Passing False will result in a static forecast that will only update if a cell passed in the 2nd argument is changed. If True is passed for this argument, a random error will be included in the forecasted points. See the Time Series Simulation example below for more information on passing True for this argument. In this case, pass **False.**

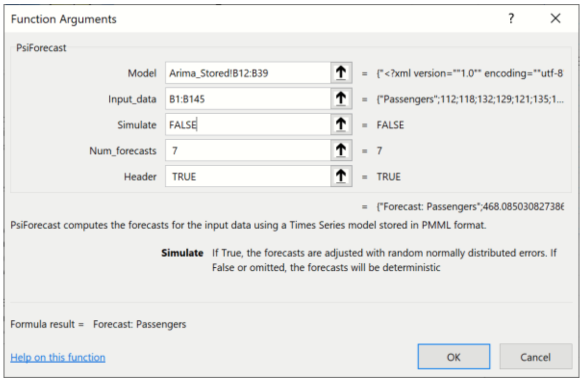
Pass **7** for the next argument, number of forecasts (Num_forecasts).

Pass **True** for Header to display a header at the top of the results.

Your formula should now be the following:
=PsiForecast(ARIMA_Stored!B12:B39,Data!B1:B145, False, 7, True).

Note: If using a version of Excel that does not support Dynamic Arrays, press **CTRL** + **SHIFT** + **ENTER** to enter this formula as an array in all seven cells (B146:B152).

It's also possible to enter this formula using the Insert Function dialog by clicking Formulas – Insert Function, select PSI Data Science for Category, then PsiForecastARIMA.



Model Arima_Stored!B12:B39 = {"<?xml version=""1.0"" encoding=""utf-8

Input_data B1:B145 = {"Passengers";112;118;132;129;121;135;1...

Simulate FALSE = FALSE

Num_forecasts 7 = 7

Header TRUE = TRUE

= {"Forecast: Passengers";468.08503082738€

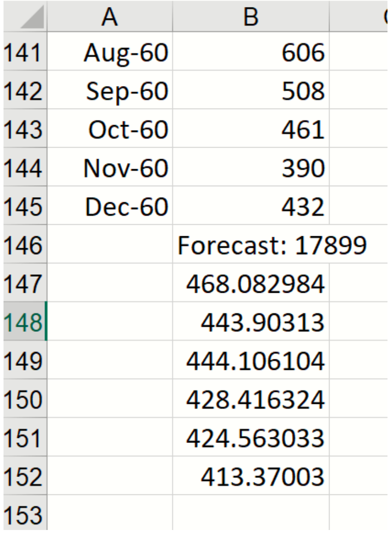
PsiForecast computes the forecasts for the input data using a Times Series model stored in PMML format.

Simulate If True, the forecasts are adjusted with random normally distributed errors. If False or omitted, the forecasts will be deterministic

Formula result = Forecast: Passengers

The results from this function are displayed below.

Notice that the formula is entered into cell B146 and the contents of the PsiForecast() Dynamic Array "spill" down into cells B147:B152.



	A	B
141	Aug-60	606
142	Sep-60	508
143	Oct-60	461
144	Nov-60	390
145	Dec-60	432
146		Forecast: 17899
147		468.082984
148		443.90313
149		444.106104
150		428.416324
151		424.563033
152		413.37003
153		

If any values change in the ranges ARIMA_Stored!B12:B38 or Data!B2:B145, the forecast will be recomputed; but if the input argument values stay the same, the PsiForecast() function will always return the same forecast values. As mentioned above, the first forecasted value is the most accurate predicted point. Accuracy declines as the number of forecasted points increases.

See the section below for specifications on PsiForecast().

PsiForecast()

`PsiForecast(Model, Input_Data, [Simulate], [Num_Forecasts], [Header])`

Computes the forecasts for `Input_Data` using a Time Series model stored in PMML format.

`Model`: Range containing the stored Times Series model in PMML format.

`Input_Data`: Range containing the new Time Series data for computing the forecasts. Range must contain a header with the time series name and a sufficient number of records for the forecasting with a given model.

`Simulate`: If True, the forecasts are adjusted with random normally distributed errors. If False or omitted, the forecasts will be deterministic.

`Num_Forecasts`: Enter the number of desired forecasts.

`Header`: If True, a heading will be inserted above the forecasted results. If False or omitted, a heading will not be included in the result.

Output: A single column containing the header, if used, and forecasts for input time series. The number of produced forecasts is determined by the number of selected cells in the array-formula entry.

Supported Models:

- Arima
- Exponential Smoothing
- Double Exponential Smoothing
- Holt Winters Smoothing

Previous related Psi Scoring functions: PsiForecastARIMA, PsiForecastExp, PsiForecastDoubleExp, PsiForecastMovingAvg, PsiForecastHoltWinters

Time Series Simulation

Analytic Solver Data Science includes the ability to perform a time series simulation, where future points in a time series are forecast on each Monte Carlo trial, using a model created via ARIMA or one of our smoothing methods (Exponential, Double Exponential, Moving Average, or Holt Winters).

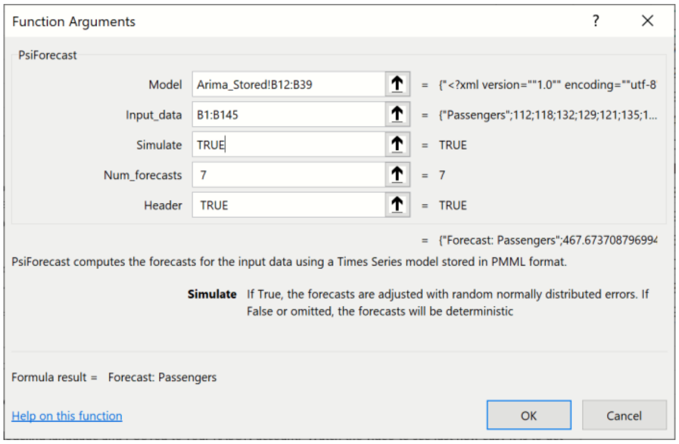
To run a time series simulation, we must pass "True" as the third argument to PsiForecast(). When the third argument is set to True, Analytic Solver will add a random (positive or negative) "epsilon" value to each forecasted point. Each time a simulation is run, 1000 trial "epsilon" values are generated using the PsiNormal distribution with parameters mean and standard deviation computed by the PsiForecast() function. You can view the output of this simulation in the same way as you would view "normal" simulation results in Analytic Solver Comprohensive, Analytic Solver Simulation or Analytic Solver Upgrade, simply by creating a PsiOutput() function and then double clicking the Output cell to view the Simulation Results dialog.

Select a blank cell, or Data!C146:C152 if using a version of Excel that does not support Dynamic Arrays, then click Formulas – Insert Function to display the Function Argument dialog.

As discussed previously, the first argument, ARIMA_Stored!B12:B38, is the range of cells used by Analytic Solver to store the ARIMA model on the *ARIMA_Stored* worksheet. This data range will change as the forecast method changes.

For the second argument the range containing the initial points in the series must be greater than the minimum number of initial points for a static forecast. For a seasonal ARIMA model when Simulate = True, the minimum number of initial points must be greater than Max((p + d + s * (P + D), (q + s * Q). In this example, p + d + s * (P + D) is equal to 13 (0 + 1 + 12 * (0 + 1) and q + s * Q is equal to 13 (1 + 12 * 1), therefore the minimum number of initial starting points required is 13 (Minimum #Initial Points > MAX (13, 13)). However, when PsiForecastARIMA() is called with Simulate = True, it is recommended to add an additional number of datapoints, equal to the #Periods, to the minimum number required. In this instance the number of initial points will be 25: 13 (minimum # of points) + 12 (# of points for Period in the Time Series - ARIMA dialog). If you provide fewer than the minimum required number of starting points (13 in this example) PsiForecastARIMA() will return #VALUE. (See the table below for the minimum number of initial starting points required by each forecast method in Analytic Solver.) All points supplied in the second argument will be used in the forecast. Select or enter **Data!B1:B145,** for this argument.

Passing TRUE for the third argument indicates to Analytic Solver Data Science that you plan to use this function call in a Monte Carlo simulation, so it should add a random epsilon value (different on each Monte Carlo trial) to each forecasted point.



Function Arguments

PsiForecast

Model	Arima_Stored!B12:B39	= {"<?xml version=""1.0"" encoding=""utf-8
Input_data	B1:B145	= {"Passengers";112;118;132;129;121;135;1...
Simulate	TRUE	= TRUE
Num_forecasts	7	= 7
Header	TRUE	= TRUE

= {"Forecast: Passengers";467.67370879699

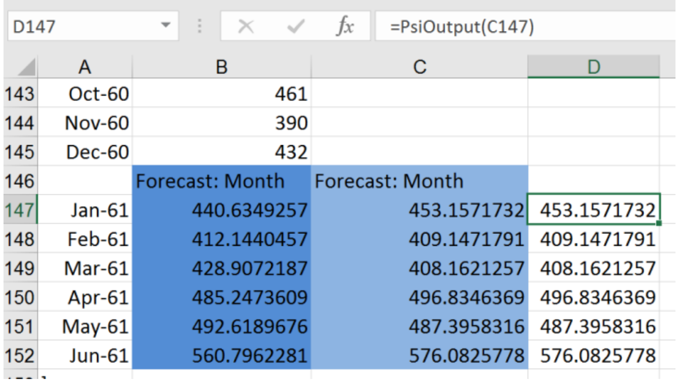
PsiForecast computes the forecasts for the input data using a Times Series model stored in PMML format.

Simulate If True, the forecasts are adjusted with random normally distributed errors. If False or omitted, the forecasts will be deterministic

Formula result = Forecast: Passengers

In versions of Excel supporting Dynamic Arrays, this formula is entered into cell C146 and the contents of the PsiForecast() Dynamic Array "spills" down into cells C147:C152. Note: If versions of Excel that do not support Dynamic Arrays, the formula must be entered as an Excel array.

To view the results of the simulation including frequency and sensitivity charts, statistics, and percentiles for the full range of trial values in any version of Excel, we must first create an output cell. Select cell B147, then click Analytic Solver – Results – Referred Cell. Select cell D147 (or any blank cell on the spreadsheet) to enter the PsiOutput formula. Copy this formula from cell D147 down to cell D152. Therefore D147 = PsiOutput(C147), D148 = PsiOutput(C148), and so on.



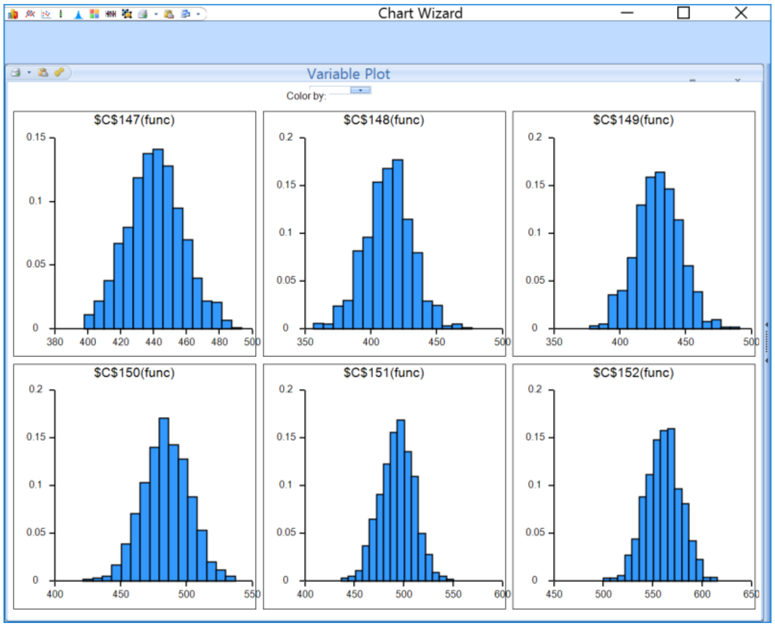
D147 =PsiOutput(C147)

	A	B	C	D
143	Oct-60	461		
144	Nov-60	390		
145	Dec-60	432		
146		Forecast: Month	Forecast: Month	
147	Jan-61	440.6349257	453.1571732	453.1571732
148	Feb-61	412.1440457	409.1471791	409.1471791
149	Mar-61	428.9072187	408.1621257	408.1621257
150	Apr-61	485.2473609	496.8346369	496.8346369
151	May-61	492.6189676	487.3958316	487.3958316
152	Jun-61	560.7962281	576.0825778	576.0825778

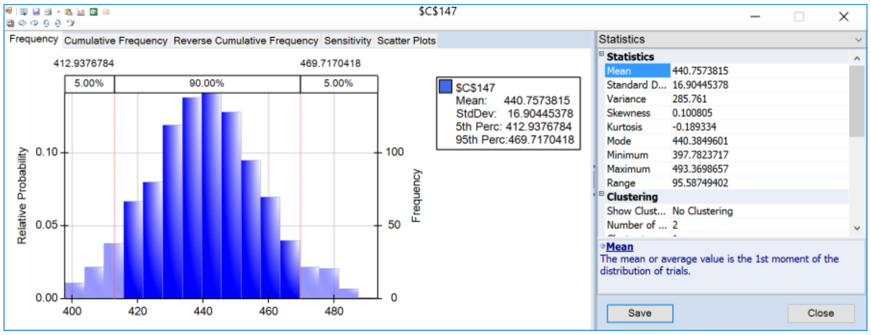
Click the down arrow on the Simulate icon and select Run Once. Instantly, Analytic Solver will perform a simulation with 1,000 Monte Carlo simulation trials (the default number). Since this is the first time a simulation has been performed, the following dialog opens. Subsequent simulations will not produce this report. However, it is possible to reopen the individual frequency charts by double clicking each of the output cells (B147:B152).

Important Note: For Users who are familiar with simulation models in Analytic Solver Simulation, you'll notice that the time series simulation model that we just created now includes 6 uncertain functions, B147:B152, which are the cells containing our PsiForecast() functions. For more information on simulation with Analytic Solver, please see the Analytic Solver User Guide chapter, "Examples: Simulation and Risk Analysis".

PsiForecast() is not recognized as an uncertain function in the Cloud apps. If the simulation argument is set to "True", Analytic Solver App will generate a single random point around the forecast.



This dialog displays frequency charts for each of the six cells containing the forecasted data points. Double click the chart for cell C147 (top left) to open the Simulation Results dialog for the PsiForecast() function in cell C147. From here you can view frequency and sensitivity charts, statistics and percentiles for each forecasted point.



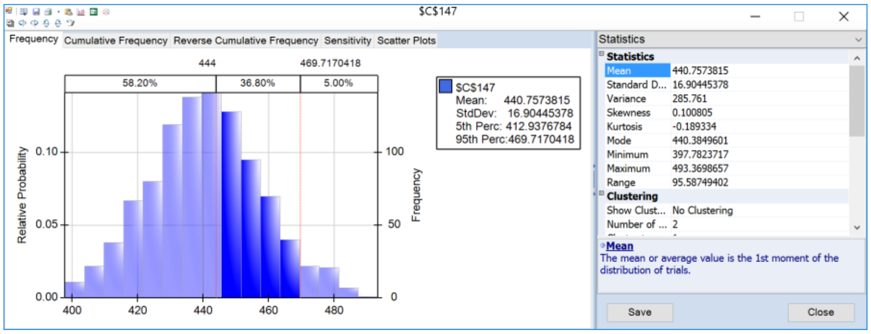
$C$147
Mean: 440.7573815
StdDev: 16.90445378
5th Perc: 412.9376784
95th Perc:469.7170418

Statistics	
Mean	440.7573815
Standard D...	16.90445378
Variance	285.761
Skewness	0.100805
Kurtosis	-0.189334
Mode	440.3849601
Minimum	397.7823717
Maximum	493.3698657
Range	95.58749402
Clustering	
Show Clust...	No Clustering
Number of ...	2

Mean
The mean or average value is the 1st moment of the distribution of trials.

The frequency chart displays the distribution of all 1000 trial values for cell C147 with and observed mean 440.76 and standard deviation of 16.90 shown in the Chart Statistics. Select Simulate – Run Once a few more times (or click the green “play” button on the Solver Pane Model tab). Each time you do, another 1,000 Monte Carlo trials are run, and a slightly different mean will be displayed.

Enter 444 for the Lower Cutoff in the Chart Statistics section of the right panel, A vertical bar appears over the Frequency chart to display the frequency with which the forecasted value was greater than this value during the simulation. You can use this as an *estimate* of the probability that the actual value will be less than the forecasted value. In this case there was a 58.20% chance that the number of international airline passengers would be less than 444,000 in January 1961 and a 41.80% chance that the number of passengers would be greater than 444,000.



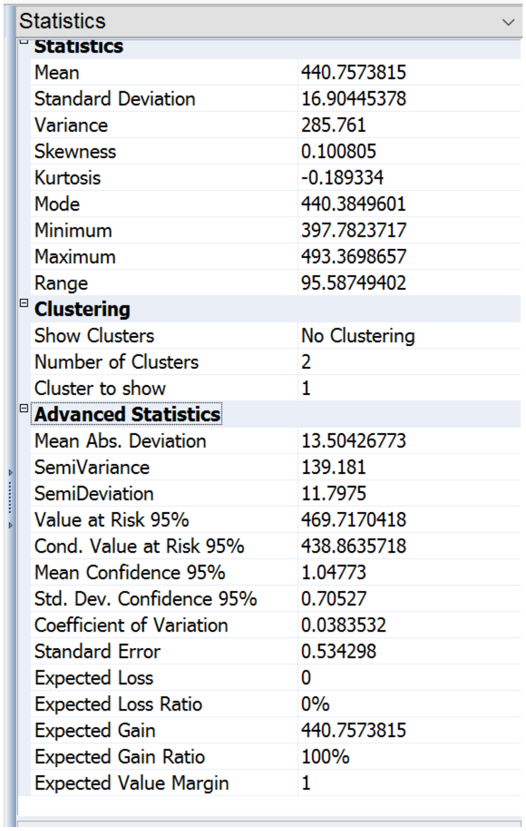
$C$147
Mean: 440.7573815
StdDev: 16.90445378
5th Perc: 412.9376784
95th Perc:469.7170418

Statistics

Statistics	
Mean	440.7573815
Standard D...	16.90445378
Variance	285.761
Skewness	0.100805
Kurtosis	-0.189334
Mode	440.3849601
Minimum	397.7823717
Maximum	493.3698657
Range	95.58749402
Clustering	
Show Clust...	No Clustering
Number of ...	2

Mean
The mean or average value is the 1st moment of the distribution of trials.

Looking to the right, you’ll find the Statistics pane, which includes summary statistics for the full range of forecasted outcomes. We can see that the minimum forecasted value during this simulation was 397.78, and the maximum forecasted value was 493.37. Value at Risk 95% shows that 95% of the time, the number of international airline passengers was 469.72 or less in January 1961, in this simulation. The Conditional Value at Risk 95% value indicates that the average number of passengers we would have seen (up to the 95% percentile) was 438.86. For more information on Analytic Solver Platform’s full range of features, see the Analytic Solver User Guide chapter, “Examples: Simulation and Risk Analysis”.



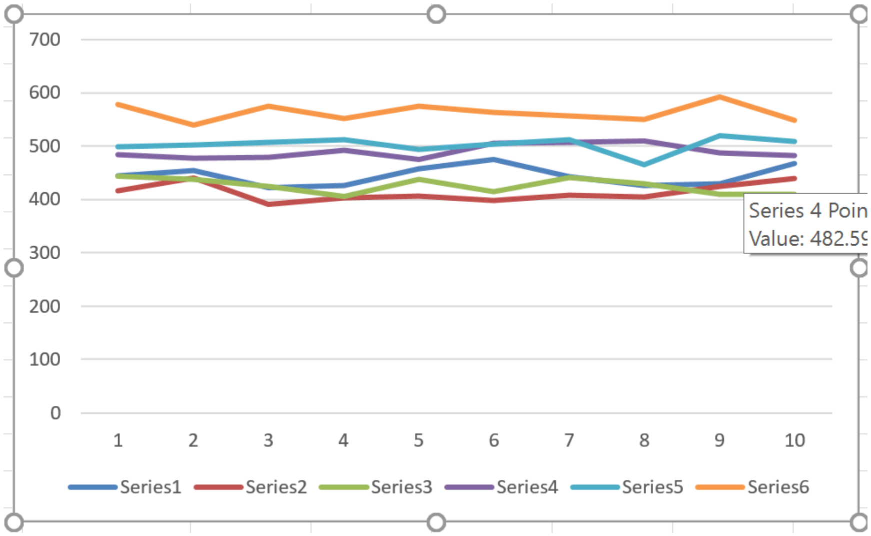
Statistics

Statistics	
Mean	440.7573815
Standard Deviation	16.90445378
Variance	285.761
Skewness	0.100805
Kurtosis	-0.189334
Mode	440.3849601
Minimum	397.7823717
Maximum	493.3698657
Range	95.58749402
Clustering	
Show Clusters	No Clustering
Number of Clusters	2
Cluster to show	1
Advanced Statistics	
Mean Abs. Deviation	13.50426773
SemiVariance	139.181
SemiDeviation	11.7975
Value at Risk 95%	469.7170418
Cond. Value at Risk 95%	438.8635718
Mean Confidence 95%	1.04773
Std. Dev. Confidence 95%	0.70527
Coefficient of Variation	0.0383532
Standard Error	0.534298
Expected Loss	0
Expected Loss Ratio	0%
Expected Gain	440.7573815
Expected Gain Ratio	100%
Expected Value Margin	1

Select cells E147:E156 and then enter the formula, =PsiData(C147), then press CTRL + SHIFT + ENTER to array enter the formula into all 10 cells. Repeat the same steps to array enter “=PsiData(C148)” in cells F147:F156, “=PsiData(C149)” in cells G147:G156, “=PsiData(C150)” in cells H147:H156, “PsiData(C151)” in cells I147:I156, and “=PsiData(C152)” in cells J147:J156. Then click Simulate – Run Once to run a simulation.

The ten Excel cells in these columns will update with trial values for each of the PsiForecast() functions in column C. For example, cells E147:N147 will contain the first 10 trial values for the PsiForecast() function in cell C147, Cells E147:N147 will contain the first 10 trial values for cell C148 and so on. (For more information on the PsiData() function, please see the Excel Solvers Reference Guide chapter, “Psi Function Reference.”)

If we create an Excel chart of these values, you’ll see a chart similar to the one below where each of Series1 through Series6 represents a different Monte Carlo trial. The random “epsilon” value added to each forecast value accounts for (all of) the variation among the lines. If the third argument were FALSE or omitted, all of the lines would overlap, assuming that the table or parameters and the starting values were not changing.



The remaining Forecasting methods can be used in the same way using PsiForecast() with information from their respective Stored Model sheets.

Forecasting Algorithm	Stored Model Sheet	Minimum # of Initial Points when Simulate = False	Minimum # of Initial Points when Simulate = True
Non- Seasonal ARIMA	ARIMA_Stored	Max(p + d, q)	Max(p + d, q)
Seasonal ARIMA	ARIMA_Stored	Max((p + d + s *(P + D), (q + s * Q)	1 + Max((p + d + s *(P + D), (q + s * Q)**
Exponential Smoothing	Expo_Stored	1	1
Double Exponential Smoothing	DoubleExpo_Stored	1	1
Moving Average Smoothing	MovingAvg_Stored	# of Intervals	# of Intervals
Holt Winters Smoothing	MulHoltWinters_Stored	2 * #Periods	2 * #Periods

	AddHoltWinters_Stored NoTrendHoltWinters_Stored		

**Adding a number of data points equal to the Number of Periods (as shown on the Time Series – ARIMA dialog) to the Minimum # of Initial Points when Simulate = True is recommended when calling PsiForecast() with Simulate = True.

Scoring to a Database

This example describes the steps required to create a classification model using the Discriminant Analysis classification algorithm and then uses that model to score new data. *Note that this is only supported in Analytic Solver Desktop. This is not supported in the Data Science Cloud app.*

The example dataset Boston_Housing.xlsx will be used to illustrate the steps required. Recall that this example dataset includes 14 variables related to housing prices collected from census tracts in the Boston area. For more information on this example dataset and Discriminant Analysis in general, please see the Discriminant Analysis chapter within the Analytic Solver Data Science Reference Guide.

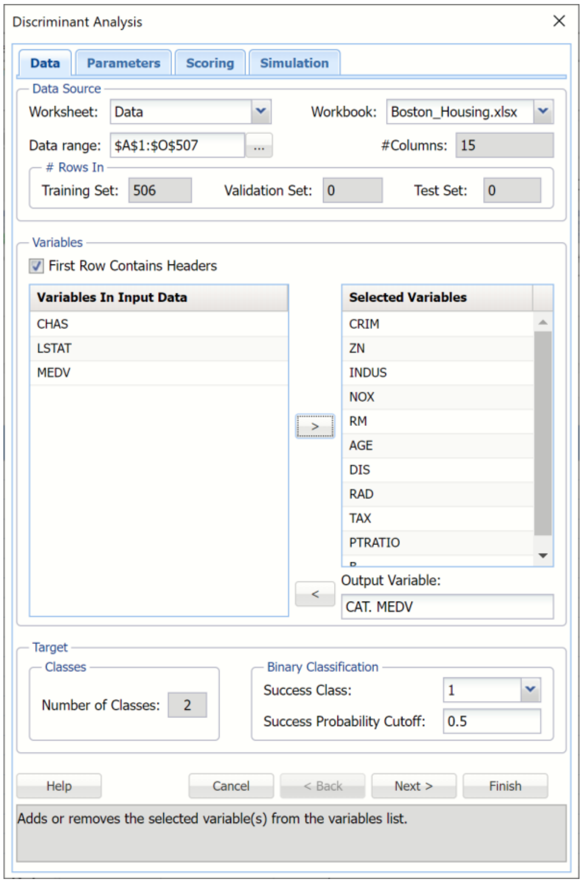
Open Boston_Housing.xlsx, then click **Classify – Discriminant Analysis** to open the *Discriminant Analysis – Data* tab.

Select the **CAT. MEDV** variable in the *Variables In Input Data* list box then click > to select as the *Output Variable*. Immediately, the options for *Classes in the Output Variable* are enabled. *#Classes* is prefilled as "2" since the *CAT. MEDV* variable contains two classes, 0 and 1.

"Success" Class (for Lift Chart) is selected by default and Class 1 is to be considered a "success" or the significant class in the Lift Chart. (Note: This option is enabled when the number of classes in the output variable is equal to 2.)

Enter a value between 0 and 1 here to denote the *Specify initial cutoff probability for success.* If the calculated probability for success for an observation is greater than or equal to this value, than a "success" (or a 1) will be predicted for that observation. If the calculated probability for success for an observation is less than this value, then a "non-success" (or a 0) will be predicted for that observation. The default value is 0.5. (Note: This option is only enabled when the # of classes is equal to 2.)

Select **CRIM**, **ZN**, **INDUS**, **NOX**, **RM**, **AGE**, **DIS**, **RAD**, **TAX**, **PTRATIO**, & **B** in the *Variables In Input Data* list box then click > to move to the *Selected Variables* list box. (*CHAS*, *LSTAT,* & *MEDV* should remain in the *Variables In Input Data* list box as shown below.)



Discriminant Analysis

Data | Parameters | Scoring | Simulation

Data Source

Worksheet: Data

Workbook: Boston_Housing.xlsx

Data range: $A$1:$O$507

#Columns: 15

Rows In

Training Set: 506 Validation Set: 0 Test Set: 0

Variables

☑ First Row Contains Headers

Variables In Input Data	Selected Variables
CHAS	CRIM
LSTAT	ZN
MEDV	INDUS
	NOX
	RM
	AGE
	DIS
	RAD
	TAX
	PTRATIO

Output Variable: CAT. MEDV

Target

Classes

Number of Classes: 2

Binary Classification

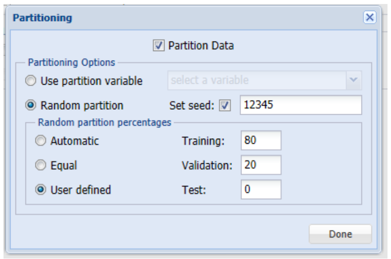
Success Class: 1

Success Probability Cutoff: 0.5

Adds or removes the selected variable(s) from the variables list.

Click **Next** to advance to the Parameters tab.

Since we did not partition the dataset before we started the classification method, we can partition the dataset now. Click **Partition Data** and then select the Partition Data option to enable the Partitioning Options. Select **User Defined**, then enter **80** for *Training* and ensure **20** is automatically entered for *Validation*. Click Done to close the dialog.



Partitioning

☑ Partition Data

Partitioning Options

○ Use partition variable select a variable

● Random partition Set seed: ☑ 12345

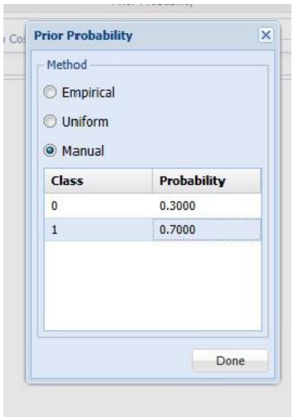
Random partition percentages

○ Automatic Training: 80

○ Equal Validation: 20

● User defined Test: 0

Click Prior Probability to open the Prior Probability dialog. Three options appear: Empirical, Uniform, and Manual.



Prior Probability

Method

○ Empirical

○ Uniform

● Manual

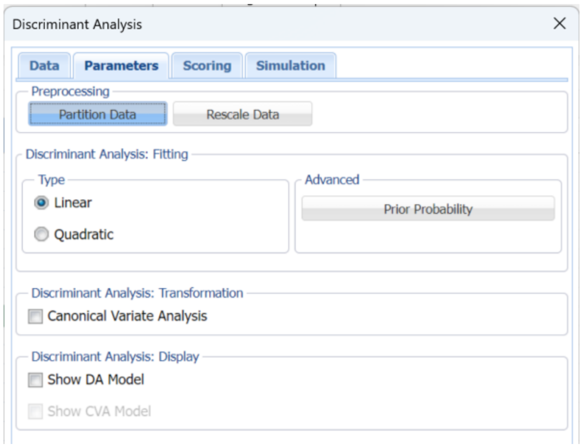
Class	Probability
0	0.3000
1	0.7000

If the first option is selected, *Empirical*, Analytic Solver Data Science will assume that the probability of encountering a particular class in the dataset is the same as the frequency with which it occurs in the training data.

If the second option is selected, *Uniform*, Analytic Solver Data Science will assume that all classes occur with equal probability.

Select the third option, *Manual*, to manually enter the desired probability values for each class.

Under *Probability,* enter **0.7** for Class 1 and **0.3** for Class 0. Click Done to close the dialog.



Discriminant Analysis

Preprocessing

Discriminant Analysis: Fitting

Type

● Linear

○ Quadratic

Advanced

Discriminant Analysis: Transformation

□ Canonical Variate Analysis

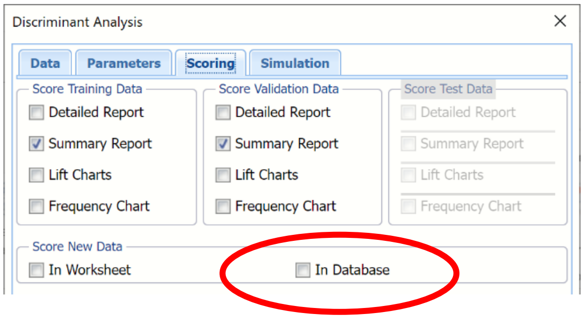
Discriminant Analysis: Display

□ Show DA Model

□ Show CVA Model

Click **Next** to advance to the *Discriminant Analysis – Scoring* tab.

Since we did not create a test partition, the options for Score test data are disabled. See the chapter "Data Science Partitioning" within the Analytic Solver Data Science Reference Guide for information on how to create a test partition.



Discriminant Analysis

Score Training Data

□ Detailed Report

☑ Summary Report

□ Lift Charts

□ Frequency Chart

Score Validation Data

□ Detailed Report

☑ Summary Report

□ Lift Charts

□ Frequency Chart

Score Test Data

□ Detailed Report

□ Summary Report

□ Lift Charts

□ Frequency Chart

Score New Data

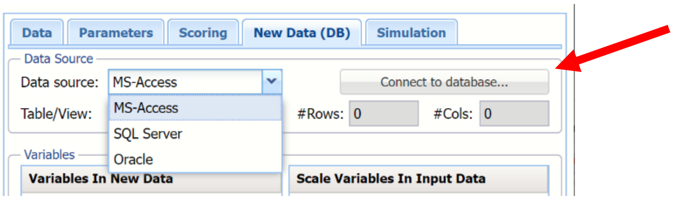
□ In Worksheet

□ In Database

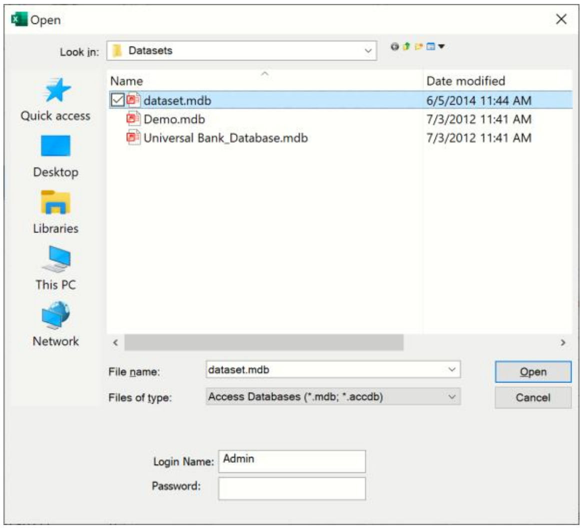
In the *Score New Data* group, select **In Database**. The *Scoring to Database* tab opens.

The first step on this tab is to select the **Data source**. Once the *Data source* is selected, **Connect to database…** will be enabled.

This example illustrates how to score to an MS-Access database. Select **MS-Access** for the *Data source*, then click **Connect to Database…**

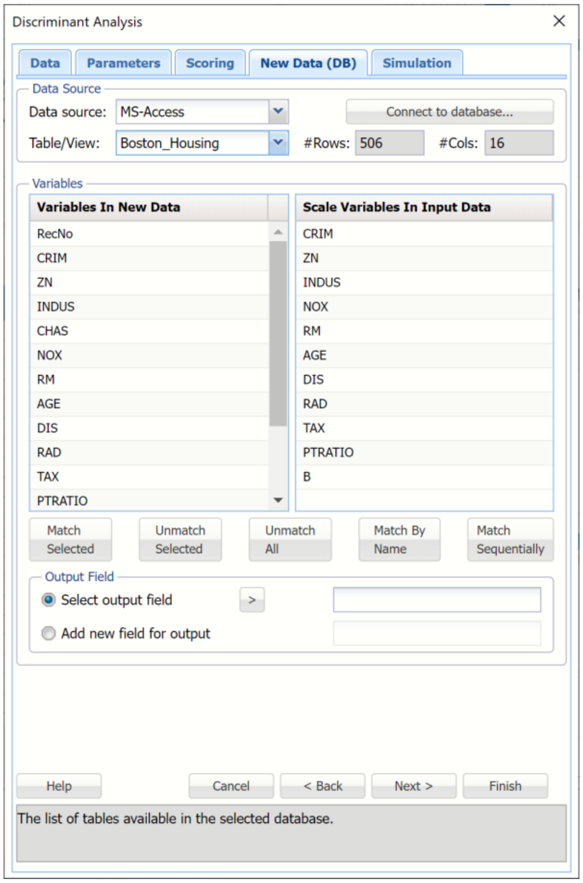


An *Open* dialog appears, browse to the location where dataset.mdb is saved, …


Open
Look in: Datasets
Quick access
Desktop
Libraries
This PC
Network
Name
Date modified
dataset.mdb 6/5/2014 11:44 AM
Demo.mdb 7/3/2012 11:41 AM
Universal Bank_Database.mdb 7/3/2012 11:41 AM
File name: dataset.mdb
Files of type: Access Databases (*.mdb; *.accdb)
Open
Cancel
Login Name: Admin
Password:

…then click **Open**. *Note the Login Name and Password fields at the bottom of the tab. If your database is password protected, enter the appropriate information here.*

The *Scoring to Database* tab re-appears. Select **Boston_Housing** for *Table/View*. The tab will be populated with variables from the database, dataset.mdb, under *Variables in New Data* and with variables from the Boston_Housing.xlsx dataset under *Scale Variables In Input Data*.



Data Source

Data source: MS-Access

Table/View: Boston_Housing

#Rows: 506 #Cols: 16

Variables

Variables In New Data	Scale Variables In Input Data
RecNo	CRIM
CRIM	ZN
ZN	INDUS
INDUS	NOX
CHAS	RM
NOX	AGE
RM	DIS
AGE	RAD
DIS	TAX
RAD	PTRATIO
TAX	B
PTRATIO	

Output Field

● Select output field

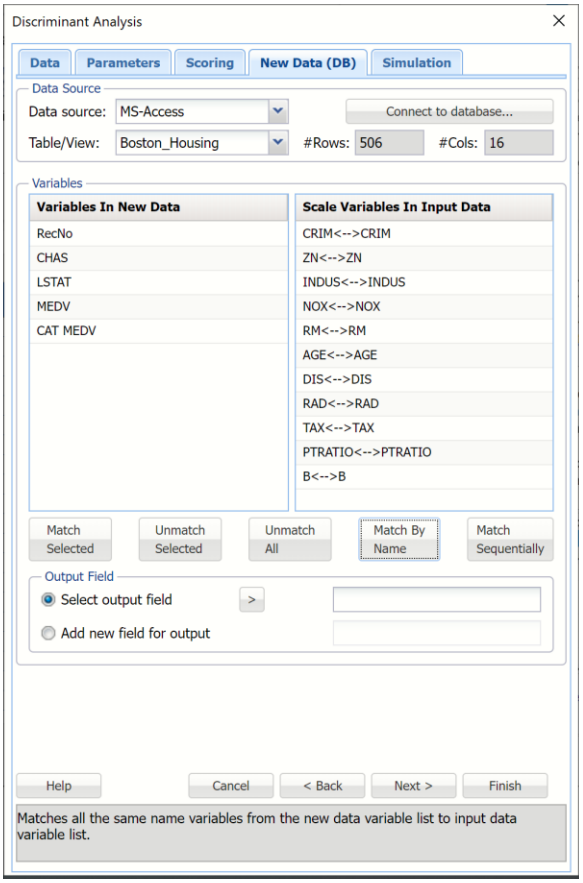
○ Add new field for output

The list of tables available in the selected database.

Analytic Solver Data Science offers three easy techniques to match variables in the dataset to variables in the database:

1. Matching by Name.
2. Matching Sequentially
3. Manually Matching.

If **Match By Name** is clicked, all similar named variables in Boston_Housing.xlsx will be matched with similar named variables in dataset.mdb, as shown in the screenprint below. Note that the additional database fields remain in the *Variables in New Data* list box while all variables in the *Scale Variables In Input Data* list box have been matched.



Data Source

Data source: MS-Access

Table/View: Boston_Housing #Rows: 506 #Cols: 16

Variables

Variables In New Data	Scale Variables In Input Data
RecNo	CRIM<-->CRIM
CHAS	ZN<-->ZN
LSTAT	INDUS<-->INDUS
MEDV	NOX<-->NOX
CAT MEDV	RM<-->RM
	AGE<-->AGE
	DIS<-->DIS
	RAD<-->RAD
	TAX<-->TAX
	PTRATIO<-->PTRATIO
	B<-->B

Output Field

● Select output field

○ Add new field for output

Matches all the same name variables from the new data variable list to input data variable list.

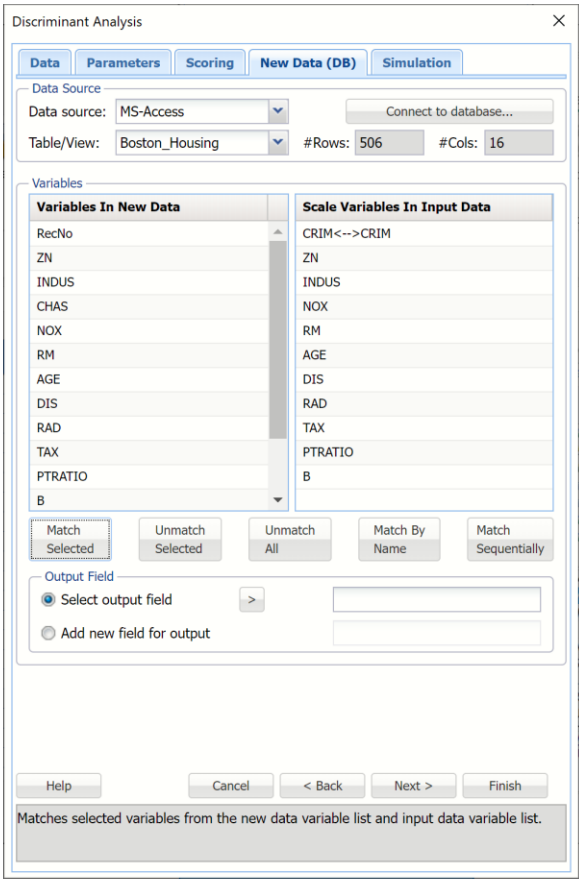
If **Match Sequentially** is clicked, the first 11 variables in Boston_Housing.xlsx will be matched with the first 11 variables in the dataset.mdb database.

The first 11 variables in both the database and the dataset are now matched under *Scale Variables In Input Data*. The additional database fields remain under *Variables in New Data*.

Note: If Match Sequentially is clicked, variables from each list are matched in sequence, i.e. 1st variable under Variables in New Data is matched with the 1st variable under Scale Variables in Input Data, 2nd variable under Variables in New Data is matched with the 2nd variable under Scale Variables in Input Data, etc.

To manually map variables from the dataset to the database, select a field from the database in the *Variables in New Data* list box, then select the variable to be matched in the dataset in the *Scale Variables In Input Data* list box, then click **Match Selected**.

For example to match the CRIM variable in the database to the CRIM variable in the dataset, select **CRIM** from the dataset.mdb database in the *Variables in New Data* list box, select **CRIM** from *Scale Variables In Input Data*, then click **Match Selected** to match the two variables.



Data source: MS-Access
Table/View: Boston_Housing
#Rows: 506
#Cols: 16

Variables In New Data	Scale Variables In Input Data
RecNo	CRIM<-->CRIM
ZN	ZN
INDUS	INDUS
CHAS	NOX
NOX	RM
RM	AGE
AGE	DIS
DIS	RAD
RAD	TAX
TAX	PTRATIO
PTRATIO	B
B	

● Select output field
○ Add new field for output

Matches selected variables from the new data variable list and input data variable list.

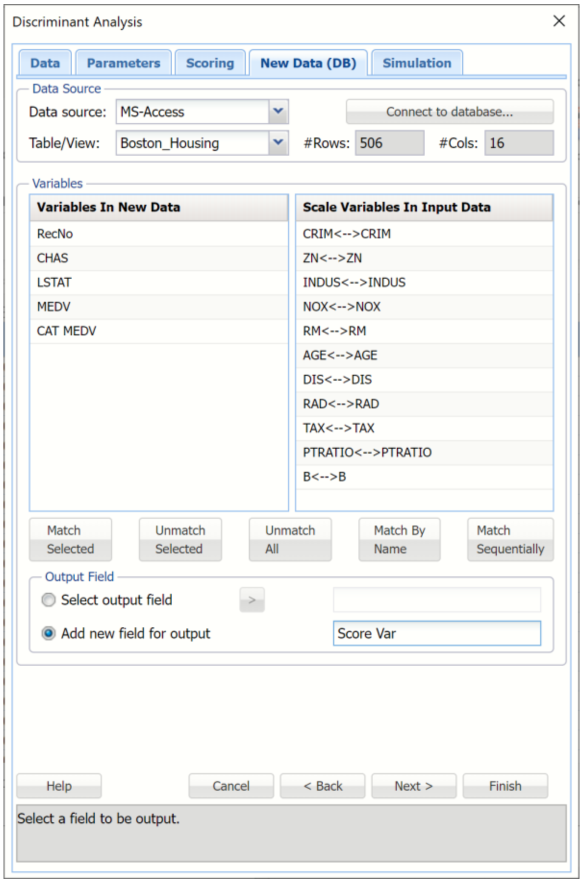
Notice that *CRIM* has been removed from the *Variables in New Data* list box and is now listed next to *CRIM* in the *Scale Variables In input data* list box. Continue with these steps to match the remaining 10 variables in the Boston_Housing.xlsx dataset.

To unmatch all variables click **Unmatch all**. To unmatch a single match, highlight the match, then click **Unmatch selected.**

An Output Field may be selected from the remaining database fields listed under *Variables in New Data* or a new Output Field can be added. *Note: An output field must be a string.*

For this example,

1. Click **Unmatch All**,
2. Click **Match By Name**,
3. Select **Add new field for output**,
4. Enter a name in the field to the right such as "Score Var".



Data Source

Data source: MS-Access

Table/View: Boston_Housing #Rows: 506 #Cols: 16

Variables

Variables In New Data	Scale Variables In Input Data
RecNo	CRIM<-->CRIM
CHAS	ZN<-->ZN
LSTAT	INDUS<-->INDUS
MEDV	NOX<-->NOX
CAT MEDV	RM<-->RM
	AGE<-->AGE
	DIS<-->DIS
	RAD<-->RAD
	TAX<-->TAX
	PTRATIO<-->PTRATIO
	B<-->B

Output Field

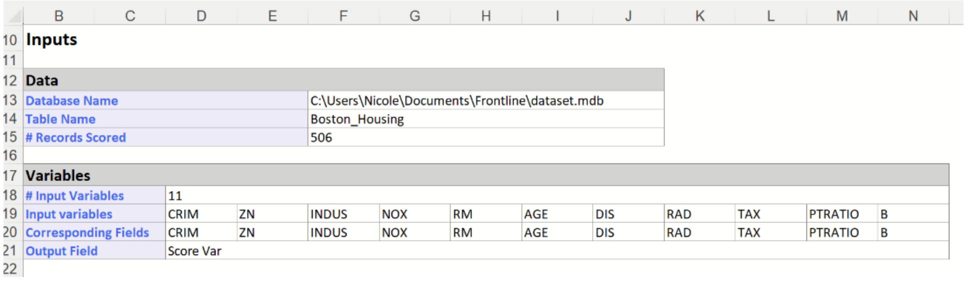
○ Select output field

● Add new field for output: Score Var

Select a field to be output.

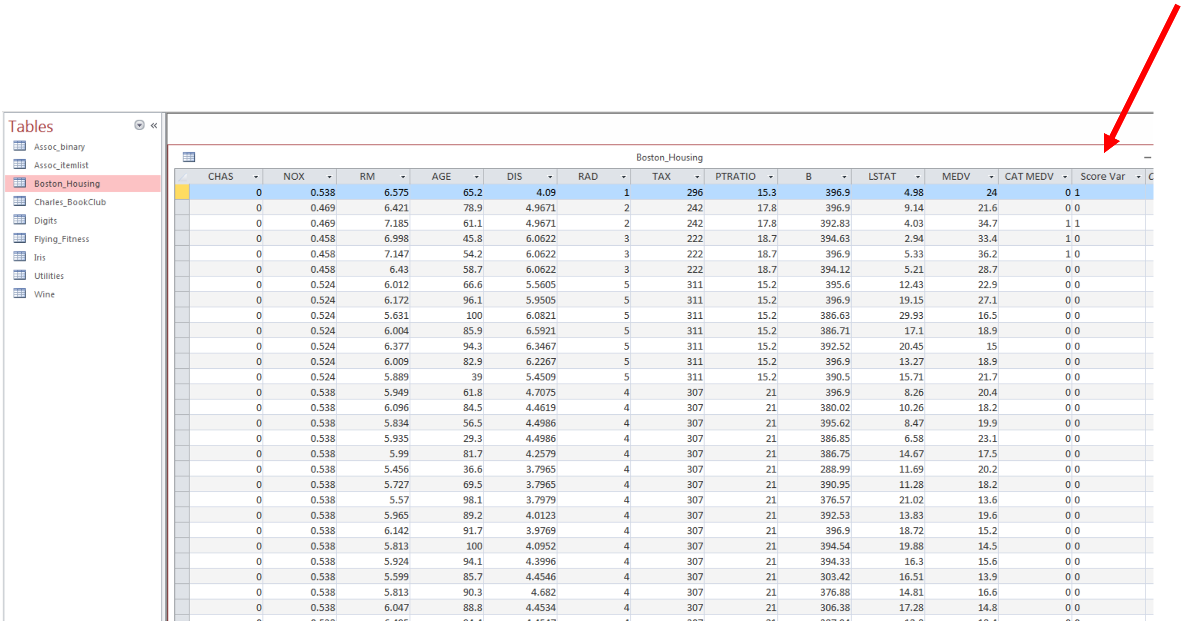
From here you can either click Finish to score the new data or you can click Next to advance to the Simulation tab where you can perform a complete risk analysis on the data. See the previous chapter, Automatic Risk Analysis of Machine Learning Models, help topic for more information on the options displayed on the Simulation tab.

Click **Finish** on the *New Data (DB)* tab. The *DA_NewScoreDB* worksheet is inserted to the right of the Data worksheet. In addition, the worksheet name is also inserted into the Model tab of the Analytic Solver task pane under Reports – Discriminant Analysis. This worksheet simply includes the name of the database, Table Name, # of records scored and the variables.



	B	C	D	E	F	G	H	I	J	K	L	M	N
10	Inputs												
11													
12	Data												
13	Database Name				C:\Users\Nicole\Documents\Frontline\dataset.mdb								
14	Table Name				Boston_Housing								
15	# Records Scored				506								
16													
17	Variables												
18	# Input Variables		11										
19	Input variables		CRIM	ZN	INDUS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
20	Corresponding Fields		CRIM	ZN	INDUS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
21	Output Field		Score Var										
22													

To view the scored records, open the *dataset.mdb* database in Microsoft Access and inspect the *Score Var* column as shown in the screenshot below. (Click the Enable Content button to view the dataset.)



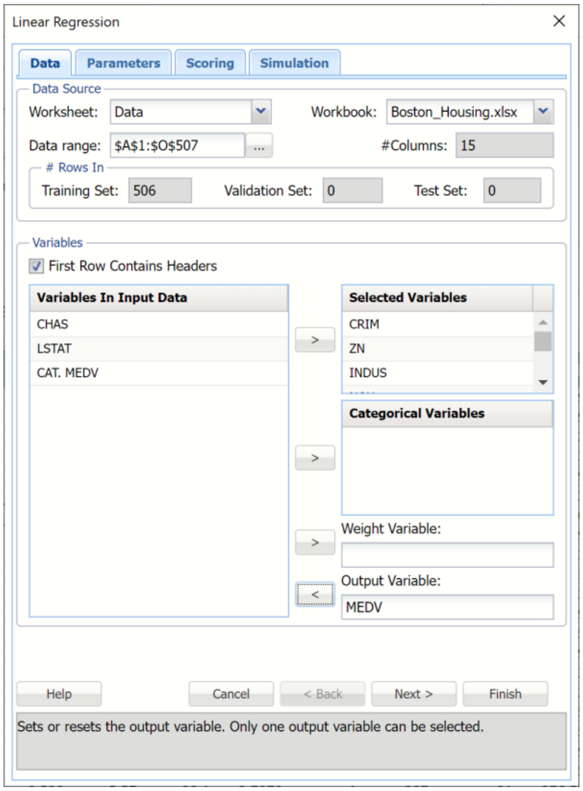
Boston_Housing

CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV	CAT MEDV	Score Var
0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24	0	1
0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6	0	0
0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7	1	1
0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4	1	0
0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2	1	0
0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7	0	0
0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9	0	0
0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1	0	0
0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5	0	0
0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9	0	0
0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15	0	0
0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9	0	0
0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7	0	0
0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4	0	0
0	0.538	6.096	84.5	4.4619	4	307	21	380.02	10.26	18.2	0	0
0	0.538	5.834	56.5	4.4986	4	307	21	395.62	8.47	19.9	0	0
0	0.538	5.935	29.3	4.4986	4	307	21	386.85	6.58	23.1	0	0
0	0.538	5.99	81.7	4.2579	4	307	21	386.75	14.67	17.5	0	0
0	0.538	5.456	36.6	3.7965	4	307	21	288.99	11.69	20.2	0	0
0	0.538	5.727	69.5	3.7965	4	307	21	390.95	11.28	18.2	0	0
0	0.538	5.57	98.1	3.7979	4	307	21	376.57	21.02	13.6	0	0
0	0.538	5.965	89.2	4.0123	4	307	21	392.53	13.83	19.6	0	0
0	0.538	6.142	91.7	3.9769	4	307	21	396.9	18.72	15.2	0	0
0	0.538	5.813	100	4.0952	4	307	21	394.54	19.88	14.5	0	0
0	0.538	5.924	94.1	4.3996	4	307	21	394.33	16.3	15.6	0	0
0	0.538	5.599	85.7	4.4546	4	307	21	303.42	16.51	13.9	0	0
0	0.538	5.813	90.3	4.682	4	307	21	376.88	14.81	16.6	0	0
0	0.538	6.047	88.8	4.4534	4	307	21	306.38	17.28	14.8	0	0

Note: If the database has been saved to a location that does not allow write-access, you will get the error, **[Microsoft][ODBC Microsoft Access Driver] Cannot modify the design of table 'Boston_Housing'. It is in a read-only database.**

Scoring to a Worksheet

Analytic Solver Data Science can also perform scoring on new data in a worksheet. To illustrate, we'll re-use the Boston Housing example dataset. (To open, click Help – Example Models – Forecasting / Data Science Examples.) Click **Predict – Linear Regression** to open the *Linear Regression - Data* tab and select the Output and Selected Variables as shown in the screenshot below.


Linear Regression

Data Source

Worksheet: Data

Workbook: Boston_Housing.xlsx

Data range: $A$1:$O$507

#Columns: 15

Rows In

Training Set: 506

Validation Set: 0

Test Set: 0

Variables

☑ First Row Contains Headers

Variables In Input Data
CHAS
LSTAT
CAT. MEDV

Selected Variables
CRIM
ZN
INDUS

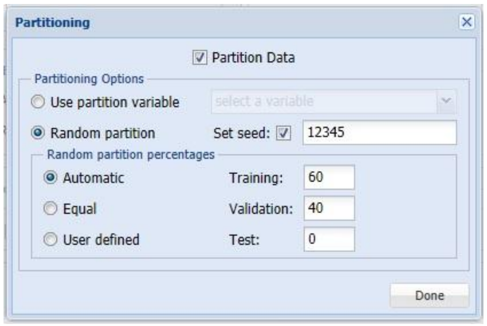
Categorical Variables

Weight Variable:

Output Variable: MEDV

Sets or resets the output variable. Only one output variable can be selected.

Click **Next** to advance to the Parameters tab. Since we haven't yet partitioned the dataset, we will do so now. Click Partition Data to open the Partition Data Dialog, then select the Partition Data option to enable the Standard Partitioning Options. For more information on these partitioning options, please see the Data Science Partitioning chapter that occurs earlier in this guide. Click Done to accept the partitioning defaults.


Partitioning

☑ Partition Data

Partitioning Options

○ Use partition variable select a variable

● Random partition Set seed: ☑ 12345

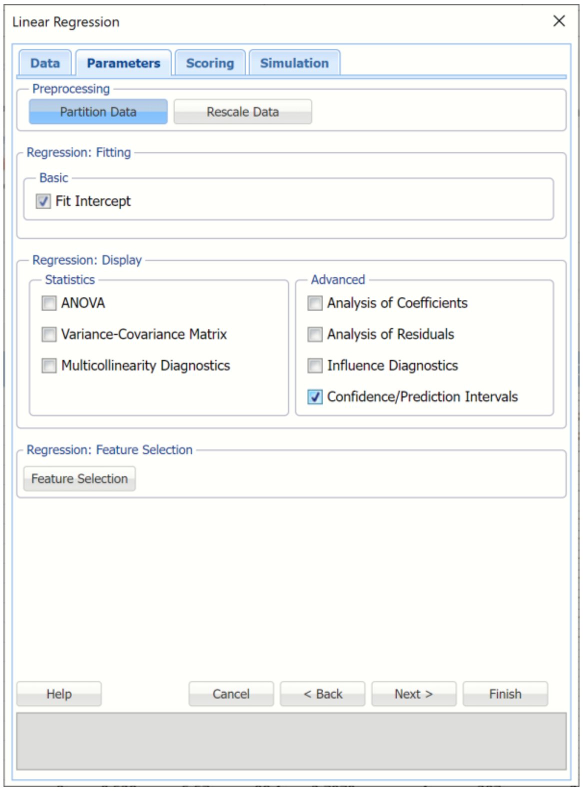
Random partition percentages

● Automatic Training: 60

○ Equal Validation: 40

○ User defined Test: 0

Under Advanced, Select Confidence/Prediction Intervals to include these intervals for the new predictions.


Linear Regression

Data | Parameters | Scoring | Simulation

Preprocessing: Partition Data | Rescale Data

Regression: Fitting

Basic

☑ Fit Intercept

Regression: Display

Statistics

□ ANOVA

□ Variance-Covariance Matrix

□ Multicollinearity Diagnostics

Advanced

□ Analysis of Coefficients

□ Analysis of Residuals

□ Influence Diagnostics

☑ Confidence/Prediction Intervals

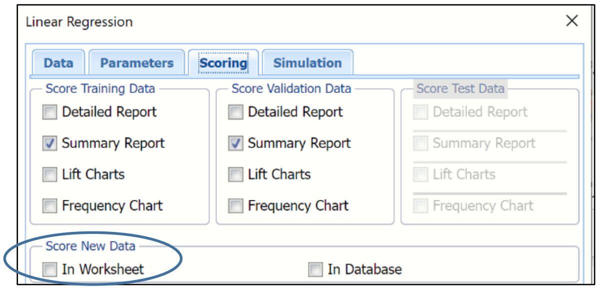
Regression: Feature Selection

Feature Selection

Help | Cancel | < Back | Next > | Finish

Click Next to advance to the Scoring tab.

Select **In Worksheet** in the *Score new data* group. The tab for *Match variables in the New Range* appears.


Linear Regression

Data | Parameters | Scoring | Simulation

Score Training Data

□ Detailed Report

☑ Summary Report

□ Lift Charts

□ Frequency Chart

Score Validation Data

□ Detailed Report

☑ Summary Report

□ Lift Charts

□ Frequency Chart

Score Test Data

□ Detailed Report

□ Summary Report

□ Lift Charts

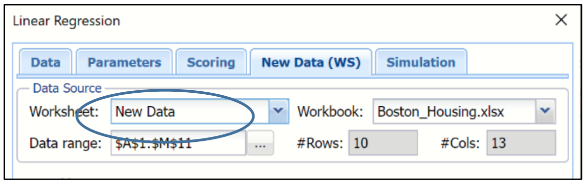
□ Frequency Chart

Score New Data

□ In Worksheet

□ In Database

Select **New Data** for *Worksheet* at the top of the tab.


Linear Regression

Data | Parameters | Scoring | New Data (WS) | Simulation

Data Source

Worksheet: New Data

Workbook: Boston_Housing.xlsx

Data range: $A$1:$M$11

#Rows: 10

#Cols: 13

The variables listed under *Variables in New Data* are from the New Data worksheet and the variables listed under Scale *Variables In Input Data* are from the Data worksheet. Variables can be matched in three different ways.

1. Matching by Name.
2. Matching Sequentially
3. Manually Matching.

If *Match By Name* is clicked, variables with the same names in each set will be matched.

If *Match Sequentially* is clicked, variables from each list are matched in sequence, i.e. 1st variable under Variables in New Data is matched with the 1st variable under Scale Variables in Input Data, 2nd variable under Variables in New Data is matched with the 2nd variable under Scale Variables in Input Data, etc.

Variables may also be matched manually by selecting a variable under *Variables in New Data*, selecting a variable in *Scale Variables In Input Data*, and clicking *Match Selected*. For example, select **CRIM** under *Variables in new data* and **CRIM** under *Scale Variables In input data*, then click **Match Selected**.

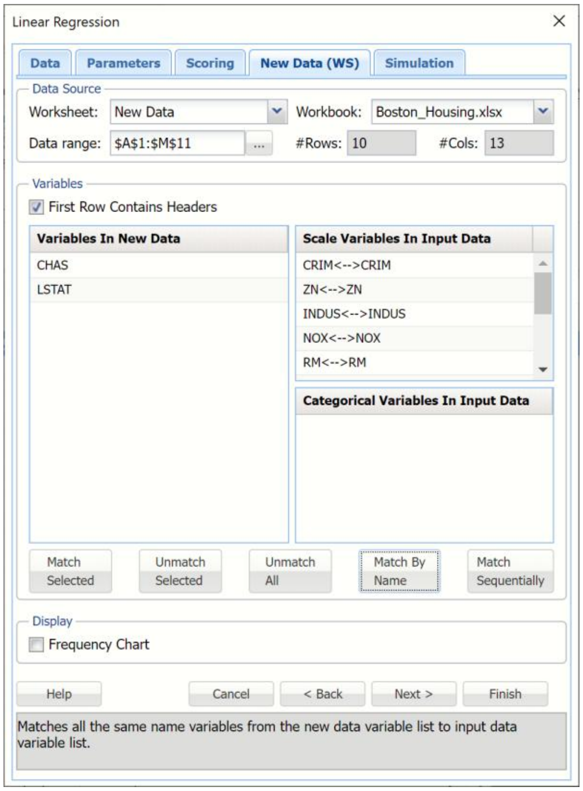
Click Match By Name to match the variables.

To unmatch all matched variables, click **Unmatch all**. To unmatch only one set of matched variables, select the matched variables in the *Scale Variables In input data* list box, then select **Unmatch Selected**.

Risk Analysis Options To unmatch only one set of matched variables, select the matched variables in the *Scale Variables In input data* list box, then select **Unmatch Selected**.

To add frequency charts to the output, select Frequency Chart. See the first section in the chapter, above, for the output from this checkbox. Leave this checkbox unchecked for this example.

From here you can either click Finish to score the new data or you can click Next to advance to the Simulation tab where you can perform a complete risk analysis on the data. See the previous chapter, Automatic Risk Analysis of Machine Learning Models, help topic for more information on the options displayed on the Simulation tab.



Worksheet: New Data

Workbook: Boston_Housing.xlsx

Data range: $A$1:$M$11

#Rows: 10

#Cols: 13

☑ First Row Contains Headers

Variables In New Data
CHAS
LSTAT

Scale Variables In Input Data
CRIM<-->CRIM
ZN<-->ZN
INDUS<-->INDUS
NOX<-->NOX
RM<-->RM

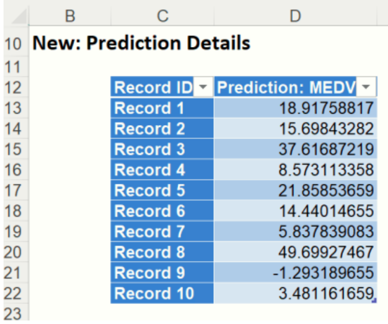
Categorical Variables In Input Data

☐ Frequency Chart

Matches all the same name variables from the new data variable list to input data variable list.

Click **Finish**.

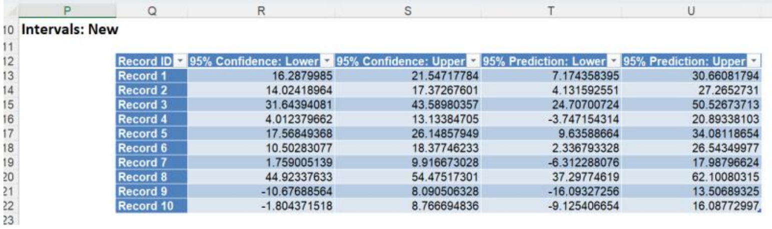
Click the *LinReg_NewScore* worksheet, inserted to the right of the Data tab, to view the output as shown below.



New: Prediction Details

Record ID	Prediction: MEDV
Record 1	18.91758817
Record 2	15.69843282
Record 3	37.61687219
Record 4	8.573113358
Record 5	21.85853659
Record 6	14.44014655
Record 7	5.837839083
Record 8	49.69927467
Record 9	-1.293189655
Record 10	3.481161659

Click the **Intervals: New** link on the Output Navigator to view the Confidence and Prediction Intervals on the LinReg_Intervals worksheet, for the new predictions.



Intervals: New

Record ID	95% Confidence: Lower	95% Confidence: Upper	95% Prediction: Lower	95% Prediction: Upper
Record 1	16.2879985	21.54717784	7.174358395	30.66081794
Record 2	14.02418964	17.37267601	4.131592551	27.2652731
Record 3	31.64394081	43.58980357	24.70700724	50.52673713
Record 4	4.012379662	13.13384705	-3.747154314	20.89338103
Record 5	17.56849368	26.14857949	9.63588664	34.08118654
Record 6	10.50283077	18.37746233	2.336793328	26.54349977
Record 7	1.759005139	9.916673028	-6.312288076	17.98796624
Record 8	44.92337633	54.47517301	37.29774619	62.10080315
Record 9	-10.67688564	8.090506328	-16.09327256	13.50689325
Record 10	-1.804371518	8.766694836	-9.125406654	16.08772997

Here we see the 95% Confidence and Prediction Intervals. Typically, Prediction Intervals are more widely utilized as they are a more robust range for the predicted value. For a given record, the Confidence Interval gives the mean value estimation with 95% probability. This means that with 95% probability, the regression line will pass through this interval. The Prediction Interval takes into account possible future deviations of the predicted response from the mean. There is a 95% chance that the predicted value will lie within the Prediction interval.

For more information on the rest of the Linear Regression output, please see the Predicting Housing Prices using Multiple Linear Regression chapter that appears earlier in this guide.