PUBLIC

SAP BusinessObjects Predictive Analytics 3.1
2017-10-26

Expert Analytics User Guide



Content

1 About Expert Analytics. 6
1.1 Expert Analytics Overview. 6
1.2 New in Expert Analytics 3.1.1. 6
1.3 Document History. 7
1.4 Documentation Resources. 7
1.5 What this Guide Contains. 8
1.6 Target Audience. 9

2 Getting Started with Expert Analytics. 10
2.1 Basics of Expert Analytics. 10
2.2 Understanding Online and Agnostic Modes 11
2.3 Launching Expert Analytics. 12
2.4 Installing R and the Required Packages. 12
2.5 Configuring R. 14
2.6 Understanding Expert Analytics. 15
Designer View. 15
Results View. 16
2.7 Using Expert Analytics from Start to Finish. 16
2.8 Configuring Advanced Features of Expert Analytics. 17

3 Acquiring Data. 19
3.1 Data Acquisition Overview. 19
Viewing a data source connection and its associated documents. 20
Acquiring data from an Excel workbook. 20
Acquiring data from a text file. 22
Acquiring data copied to the clipboard. 24
Acquiring data from SAP HANA views. 25
Acquiring data from universes. 29
Acquiring data using Query with SQL. 31
Editing an acquired data set. 37
Refreshing data in a document. 38
Renaming a data set. 38
Objects hidden from the object list. 38
Editing the enrichment suggestions file. 38

4 Preparing Data. 41
4.1 Data Preparation Overview. 41
Prepare room—viewing, cleaning, and manipulating data. 41

Editing and cleaning data. . . . 47
Creating measures and hierarchies. . . . 49
Creating a calculated measure or dimension. . . . 54
Working with multiple data sets. . . . 70

5 Building Analyses. . . . 74
5.1 Creating an Analysis. . . . 74
Applying Preprocessing Components to Data. . . . 74
HANA Demand Forecasting Component Overview. . . . 75
Configuring the HANA Sentiment Analysis Component. . . . 87
Partitioning Data. . . . 88
Applying Algorithms. . . . 88
Optional: Storing Results of Analysis. . . . 90
5.2 Running the Analysis. . . . 90
5.3 Saving the Analysis. . . . 90
5.4 Deleting an Analysis from the Document. . . . 91
5.5 Viewing Results and Exporting an Analysis. . . . 91

6 Creating R Extensions and PAL Components. . . . 92
6.1 Creating an R Extension. . . . 92
Specifying properties with the R Extension Wizard. . . . 92
Creating an R Extension. . . . 95
Plotting Multiple Charts in R Extensions. . . . 98
6.2 Creating a PAL Component. . . . 100
Specifying Properties with the PAL Component Creation Wizard. . . . 101
Creating a PAL Component. . . . 103

7 Sharing and Consuming R Extensions. . . . 104
7.1 Sharing and Consuming R Extensions. . . . 104
7.2 Sharing an R Extension. . . . 104
7.3 Browsing for and Consuming R Extensions. . . . 106
7.4 Editing an R Extension. . . . 107

8 Viewing the Results of Analysis. . . . 110
8.1 Analyzing Data. . . . 110
8.2 Scatter Matrix Chart. . . . 111
8.3 Statistical Summary Chart. . . . 111
8.4 Parallel Coordinates. . . . 112
8.5 Decision Tree. . . . 113
8.6 Trend Chart. . . . 114
8.7 Cluster Chart. . . . 115
8.8 Apriori Tag Cloud Chart. . . . 116
8.9 Confusion Matrix. . . . 116

8.10 R Extensions Chart. 117

9 Visualizing Data. 118
9.1 Visualize room—creating charts. 118
Creating charts. 121
Data sorting in charts. 132
Filtering data in the Visualize room. 133
Hierarchical data. 136
Finding measures, dimensions, and data values. 138
Measures associated with dimensions. 138
Aggregation types supported. 139

10 Creating Stories. 140
10.1 Compose room—creating stories about visualizations. 140
Page Settings panel. 140
Creating a story. 144
Modifying a story. 148
Saving a story. 148
Refreshing data on an infographic page. 148
Exploring a visualization in a story. 148
Drilling through hierarchical data in a story. 149

11 Optimizing Data. 151
11.1 Optimizing Data. 151

12 Working with Models. 153
12.1 Creating a Model. 153
12.2 Sharing Models via PMML. 153
12.3 Sharing Models Using Spar Files. 154
12.4 Sharing Custom Components Using .spar Files. 154
12.5 Importing a Model. 155
12.6 Deleting a Model. 155

13 Comparing Models. 156
13.1 Overview. 156
13.2 Comparing Two Models. 160
13.3 Comparing Three or More Models. 162

14 Exporting Models and Analyses. 164
14.1 Exporting Models and Extensions. 164
14.2 Exporting a Single SAP HANA Model as a Stored Procedure. 165
14.3 Exporting a Single R Extension as a Stored Procedure from Expert Analytics. 166
14.4 Exporting an Analysis as a Stored Procedure. 168
14.5 Exporting a Chain of Trained Models. 169

14.6 Removing an Exported Stored Procedure from SAP HANA. 170

15 Component Properties. 172

15.1 Algorithms. 172

R Algorithms and Dependent Packages. 172

Association. 173

Classification. 178

Clustering. 191

Decision Trees. 198

Outliers. 206

Neural Network. 211

Regression. 215

Time Series. 235

15.2 Data Preparation Components. 245

Data Type Definition. 246

Filter. 246

Formula. 251

Model Compare. 256

Model Statistics. 258

Normalization. 263

Partition. 266

Sample. 267

HANA Data Type Definition. 269

15.3 Data Writers. 270

CSV Writer. 270

HANA Writer. 271

JDBC Writer. 271

15.4 Models. 272

1 About Expert Analytics

1.1 Expert Analytics Overview

Expert Analytics is a statistical analysis and data mining toolset that enables you to build predictive models to discover hidden insights and relationships in your data, from which you can make predictions about future events.

Expert Analytics is a toolset of the SAP BusinessObjects Predictive Analytics application.

With Expert Analytics, you can perform various analyses on the data, including time series forecasting, outlier detection, trend analysis, classification analysis, segmentation analysis, and affinity analysis. It enables you to analyze data using different visualization techniques, such as scatter matrix charts, parallel coordinates, cluster charts, and decision trees.

Expert Analytics offers a range of predictive algorithms, supports use of the R open-source statistical analysis language, and offers in-memory data mining capabilities for handling large volume data analysis efficiently.

With Expert Analytics you can connect to various data sources such as flat files, relational databases, in-memory databases, and SAP BusinessObjects universes. In addition, you can operate on different volumes of data from a small matrix of data in a CSV file to a very large data set in SAP HANA.

1.2 New in Expert Analytics 3.1.1

Learn about the new features in Expert Analytics for SAP BusinessObjects Predictive Analytics 3.1.1.

Schedule Model Chains from Expert Analytics in Predictive Factory

You can export a chain of trained models from Expert Analytics to a local file for consumption in Predictive Factory for scheduling retrain and apply tasks. Predictive Factory will enable you to monitor the evolution of such models over time. For a step guide to the process, see Exporting a Chain of Trained Models for Consumption via SAP HANA [page 169]. To view a video tutorial on our YouTube channel, go to How to Export a Model Chain from Expert Analytics to Predictive Factory .

Convert Data Types

You can use the HANA Data Type Definition component to convert one data type to another. For a step guide to the process, see HANA Data Type Definition [page 269].

1.3 Document History

View the previous updates made to this document.

Document Updates	Links
Video Tutorial on What's New in Expert Analytics 3.0	What's New in Expert Analytics 3.0.
New sections on how to consume Expert Model chains within SAP HANA	Exporting a Chain of Trained Models for Consumption via SAP HANA [page 169].

Document Updates	Links
Video Tutorial on What's New in Expert Analytics 2.5	What's New in Expert Analytics 2.5.
New sections on how to create, edit, export and import R Extensions	Adding R Extensions [page 104].

Document Updates	Links
Video Tutorial on What's New in Expert Analytics 2.4	What's New in Expert Analytics 2.4
New section on working with the SAP HANA Demand Forecasting Component	Configuring the HANA Demand Forecasting Component [page 83].
New section on working with the SAP HANA Sentiment Analysis Component.	Configuring the SAP HANA Sentiment Analysis Component [page 87].
New section on working with the SAP HANA Optimization Function	Configuring the SAP HANA Optimization Function [page 151].

1.4 Documentation Resources

The following table provides the list of guides available for SAP BusinessObjects Predictive Analytics:

What do you want to do?	Then go here...
Get instant help on using Expert Analytics, or find information on a feature or workflow.	The Online Help is available within Expert Analytics as follows: • Click the Help icon (*?*) on a dialog box or window. • Select menu *Help* > *Help*.

What do you want to do?	Then go here...
Get instant help on using Automated Analytics, or find information on a feature or workflow.	Contextual help for each panel is available within Automated Analytics. Either press F1 or select menu *Help* > *Get Help for this Panel*. Full, searchable online help for Automated Analytics is available: Select menu *Help* > *Open Full Searchable Help*.
Get instant help on using Predictive Factory.	From anywhere in the application • Click the Help icon (*?*) to activate the integrated In-Application Assistance. • Click the User Menu and select Documentation to open the Full Searchable Help.
Get complete documentation on using SAP BusinessObjects Predictive Analytics (English)	SAP BusinessObjects Predictive Analytics Home page
Get documentation on using SAP BusinessObjects Predictive Analytics in a different language. **Note** Documentation in languages other than English is only available for certain guides.	SAP All Products page Select a language, then select SAP BusinessObjects Predictive Analytics and the version required from the drop-down lists.
Get the latest information on database and software support for SAP BusinessObjects Predictive Analytics.	Go to SAP Product Availability Matrix and search for "SAP BusinessObjects Predictive Analytics"

1.5 What this Guide Contains

This guide provides:

- An overview of Expert Analytics
- How to acquire data from various data sources
- How to perform data manipulation, data cleansing, and semantic enrichment operations in the *Prepare* room
- Information on various algorithms and components available in Expert Analytics
- Information on how to create analyses and models
- Information on how to analyze data using predictive visualization techniques
- How to create story boards
- How to share charts and datasets

Note

Expert Analytics inherits data acquisition and data manipulation functionality from SAP Lumira. Therefore, for information about workflows not covered in this guide, see the *SAP Lumira User Guide* available at: http://help.sap.com/lumira.

> **Note**
>
> Information about how to install and configure the application is covered in the *SAP BusinessObjects Predictive Analytics Desktop Installation Guide* available at: http://help.sap.com/pa. .

1.6 Target Audience

This guide is intended for professional data analysts, business users, statisticians, and data scientists who want to use Expert Analytics to analyze and visualize data using predictive algorithms.

> **Note**
>
> To use Expert Analytics, you need to be familiar with statistical and data mining algorithms and have a basic understanding on how to use these algorithms.

2 Getting Started with Expert Analytics

2.1 Basics of Expert Analytics

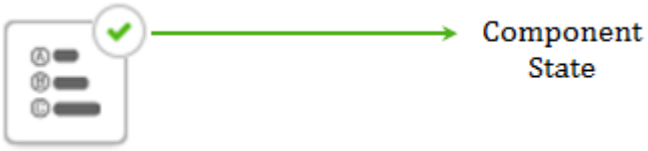
Important concepts that are relevant when using Expert Analytics.

Component

A component is the basic processing unit of Expert Analytics. Each component has one input and/or multiple output connection points. These connection points are used to connect components through connectors. After connecting components together, data is transmitted from predecessor components to their successor components.

Expert Analytics consists of the following components:

- Preprocessors
- Algorithms
- Data writers



You can access components from the Designer view of the *Predict* room. After you have added components to the analysis editor, the status icon of a component allows you to identify its state.

The following are the states of a component:

- No status icon: This state is displayed when you drag a component onto the analysis editor. It indicates that the component needs to be configured before running the analysis.
- (Configured): This state is displayed once all the necessary properties are configured for the component.
- (Success): This state is displayed after the successful execution of the analysis.
- (Failure): This state is displayed if this component causes the execution of the analysis to fail.

Analysis

An analysis is a series of different components connected together in a particular sequence with connectors, which define the direction of the data flow.

Model

A model is a reusable component created by training an algorithm using historical data.

In-Database (In-DB) working mode

In-Database (In-DB) is an analysis execution mode in which data processing is performed in the SAP HANA database with the use of data mining capabilities. In this mode, the SAP BusinessObjects Predictive Analytics desktop orchestrates execution on the SAP HANA side, and minimal data for reporting is downloaded to the client. This mode can be used to process large data sets. SAP HANA supports data mining through R integration, SAP Predictive Analysis Library (PAL) and SAP Automated Predictive Library (APL). This type of analysis is also referred to as online analysis.

> **Note**
>
> For information about sizing the SAP HANA database to perform In-DB analysis, see SAP Note 1514966.

In-Process (In-Proc) working mode

In-Process (In-Proc) is an analysis execution mode in which the data processing is performed by taking data out of the database into the predictive process space. In this mode, you cannot use SAP HANA PAL algorithms for analysis. However, you can work with R and SAP algorithms. This type of analysis is also referred to as Out-DB or offline analysis.

> **Note**
>
> For information about hardware requirements needed to perform In-Proc analysis, see the Product Availability Matrix at SAP Product Availability Matrix

2.2 Understanding Online and Agnostic Modes

Expert Analytics operates in SAP HANA online or agnostic (offline) modes.

In SAP HANA online mode, algorithms and functions for analyzing data are grouped into an Application Function Library (AFL). Expert Analytics libraries include Predictive Analytical Library (PAL), Automated Predictive Library (APL), and Unified Demand Forecast (UDF). The data science programming language R is also supported in this mode. Data processing executes on the SAP HANA/server side and minimal data is transferred back to the desktop client for reporting.

In agnostic mode, you download data from the data source. For example, to download an SAP HANA data source in Expert Analytics, you select *File* > *New*, click *New Document* and choose *Download from SAP*

HANA. The algorithms and functions are local to the desktop, including those for R and non-SAP HANA data sources. In this mode, data processing executes on the desktop and is limited by the capacity of available resources on the local computer.

2.3 Launching Expert Analytics

To launch Expert Analytics, choose *Start* > *All Programs* > *SAP Business Intelligence* > *SAP BusinessObjects Predictive Analytics Desktop* > *SAP BusinessObjects Predictive Analytics* > *Expert Analytics*.

2.4 Installing R and the Required Packages

R is an open-source programming language and software environment for statistical computing.

Expert Analytics supports the following R algorithms and dependent packages for both SAP HANA and agnostic platforms. For more information about supported platforms and technologies, click to access the Product Availability Matrix page and search for "Predictive Analytics", http://service.sap.com/sap/support/pam.

R Algorithm	Dependent Packages
HANA R-Apriori	arules
HANA R-CNR Tree	rpart
HANA R-Multiple Linear Regression	stats
HANA R-Triple Exponential Smoothing	stats
HANA R-Bagging Classification	adabag, rpart
HANA R-Boosting Classification	adabag, rpart
HANA R-Random Forest Classification	randomForest
HANA R-Random Forest Regression	randomForest

R Algorithm	Dependent Packages
R-CNR Tree	rpart
R-Apriori	arules
R-K-Means	stats

R Algorithm	Dependent Packages
R-Linear Regression	stats
R-Multiple Linear Regression	stats
R-Exponential Regression	stats
R-Geometric Regression	stats
R-MONMLP Neural Network	monmlp
R-NNet Neural Network	nnet
R-Single Exponential Smoothing	stats
R-Double Exponential Smoothing	stats
R-Triple Exponential Smoothing	stats
R-Bagging Classification	adabag, rpart
R-Boosting Classification	adabag, rpart
R-Random Forest Classification	randomForest
R-Random Forest Regression	randomForest

Prerequisites:

To use open-source R algorithms in your analysis, you need to install the R environment and configure it with the application.

SAP BusinessObjects Predictive Analytics provides an option to install and configure the current version of R and the required packages from within the application. Ensure that you are connected to the internet while installing R.

Before installing R (and corresponding required packages), ensure that the following requirements are met:

- The existing R is uninstalled and the registry entries and the R installation folder are removed from the machine.
- The R environment variables (R_LIBS, R_HOME) and R path variables are removed.

Installation:

To install the R environment and the required packages, perform the following steps:

1. Launch SAP BusinessObjects Predictive Analytics.
2. Open Expert Analytics.
3. From the *File* menu, choose *Install and Configure R*.
4. Select *Install R*.
5. Read the open-source R license agreement, important instructions, and select *I agree to install R using the script*.
6. Select *Ok*.

Note

If you have already installed R, you can use this procedure to install the required R packages, or manually install the following required packages through your R console:

2.5 Configuring R

After you have installed R, configure the R environment to enable R algorithms in the application. If you have already installed R-2.15.x or R-3.0.x, R-3.1.0 or R-3.1.2 and the required packages, you can skip the R installation step and directly configure R.

To configure R, perform the following steps:

1. Launch SAP BusinessObjects Predictive Analytics.
2. Open Expert Analytics.
3. From the *File* menu, choose *Install and Configure R*.
4. On the *Configuration* tab, select *Enable Open-Source R Algorithms*.
5. Choose *Browse* to select the R installation folder.

   For example, `C:\Users\Public\R-3.1.2`.
6. Choose *Ok*.
   The "User Account Control" dialog box appears with a warning message.
7. Choose *Yes* in the confirmation prompt.

Recommendation

- When installing R packages, check that the folder containing the installed R packages exists under the file path in *File* > *Install and Configure R* > *Configuration*.
  You can specify the exact location where you want an R package installed using the following command in R studio: `install.packages("PackName",lib="PATH")`
  For example: `install.packages("recomandable",lib="C:/Users/Public/R-3.1.2/library")`. After the installation, check the subfolder in the R file path. You should see a folder called *recomandable*.
- Alternatively, if you have R packages installed in multiple locations, make sure that the Microsoft Windows® environment variable, *R_LIBS*, is pointing to each location. Add the folder locations to the *R_LIBS* variable, separating each one by a semicolon so they can be found by Expert Analytics.
  For example: `R_LIBS=%R_HOME%\library;%YOUR_HOME_DIRECTORY%\Documents\R\win-library\3.1`
  If the *R_LIBS* variable is not already available, you need to create this.

  #### Note

  You can use the *.libPaths()* function to display all locations where R packages are installed.

2.6 Understanding Expert Analytics

When you launch Expert Analytics, the home page appears. The home page contains information that helps you get started.

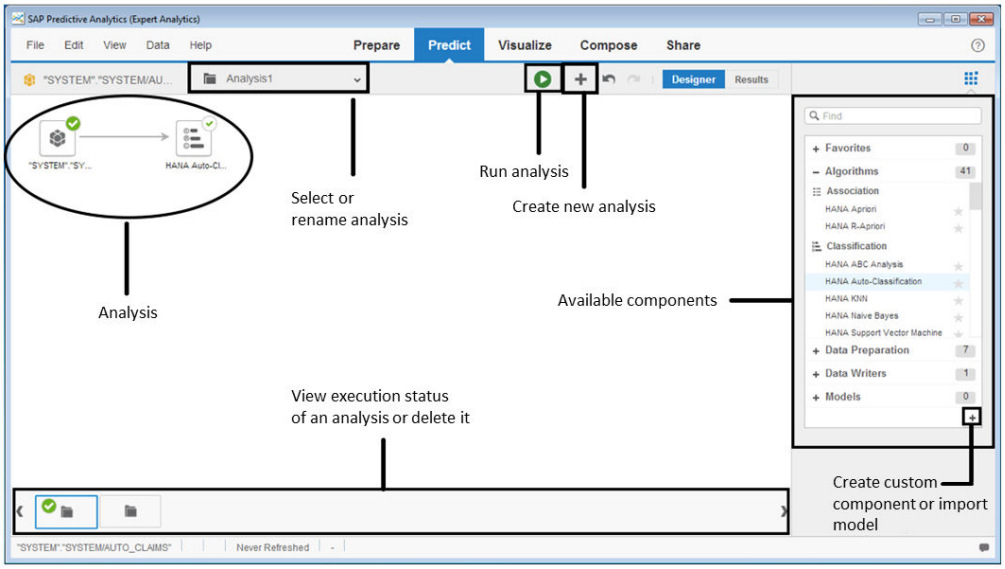
It also displays the *Try with Samples* option. This option allows you to try out the functions of Expert Analytics using sample datasets. You can also view the Expert Analytics sample documents in SAP Lumira using your SAP BusinessObjects Predictive Analytics trial license key.

To start analyzing data using Expert Analytics, you need to perform the following tasks:

- Connect to the data source and acquire data for analysis
- Prepare data for analysis by applying data manipulation and data cleansing functions
- Analyze data by applying data mining and statistical analysis algorithms
- Share datasets and charts with external collaborators

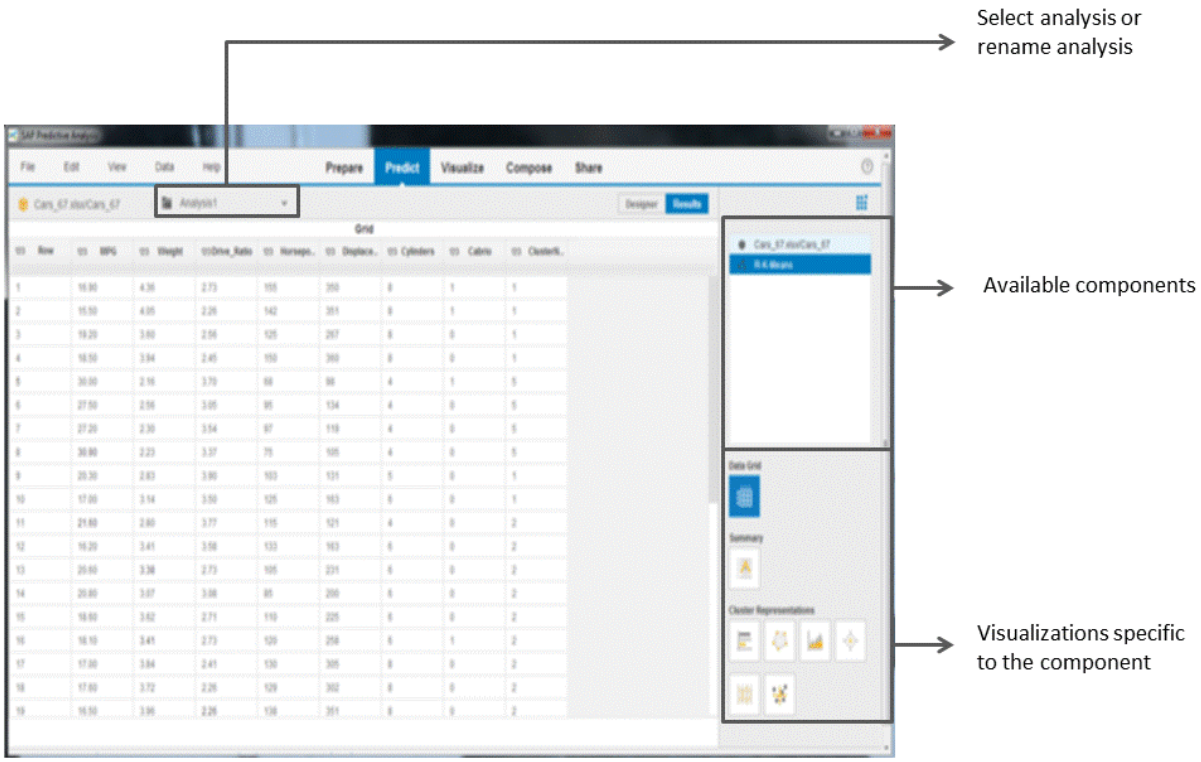
2.6.1 Designer View

The *Designer* view in the *Predict* room enables you to design and run analyses, and to create predictive models.



2.6.2 Results View

The *Results* view in the *Predict* room enables you to understand data and analysis results by using various visualization techniques and intuitive charts.



2.7 Using Expert Analytics from Start to Finish

The following is an overview of the process you can follow to build a chart based on a dataset. The process is not a linear one, and you can move from one step back to a preceding step to fine-tune your chart or data.

Steps to work with your data	Description
Connect to your data source	If your data source is: • RDBMS: Enter your credentials, connect to the database server, browse and select a data source; for example, if you are connecting to SAP HANA, you select a view and cube to create an analysis. • Flat file: Choose the columns to be acquired, trimmed, or shown and hidden.

Steps to work with your data	Description
View and organize the columns and dimensions	You can view the data acquired as columns or as facets. You can organize the data display to make chart building easier by doing the following: • Create filters and hide unneeded columns • Create measures, time hierarchies, and geography hierarchies • Clean and organize the data in columns using a range of manipulation tools • Create columns with formulas using a wide selection of available functions **Note** Please note that several filters are available in the *Predict* room, but not all. In cases where filters are not available, you can use the Filter and Formula components to create the desired effect.
Analyze your data using predictive algorithms	When you have acquired the relevant data in the *Prepare* room, switch to the *Predict* room and create an analysis to find patterns in the data and predict the future outcomes. In the *Predict* room, you can do the following: • Create an analysis • Build predictive models • View analysis results • View model visualizations
Save your analysis	Name and save the analysis that includes your charts. Analyses are saved in a document with the .lums file format in the application folder under *Documents* in your profile path - `C:\Users\<userid>\Documents\SAP Predictive Analytics Documents`. Note that results and visualizations can be viewed only in Expert Analytics.
Open an existing analysis	If you open an existing analysis that is saved in .lums file format, the following is true: • If SAP Lumira was installed on the machine before SAP BusinessObjects Predictive Analytics, the .lums document opens in Predictive Analytics. • If SAP BusinessObjects Predictive Analytics is installed on the machine without any SAP Lumira instance, the .lums document opens in SAP BusinessObjects Predictive Analytics • If SAP Lumira was installed on the machine after SAP BusinessObjects Predictive Analytics, the .lums document opens in SAP Lumira You can change this behavior by right-clicking the .lums file and associating it with the preferred tool, for example, SAP BusinessObjects Predictive Analytics

2.8 Configuring Advanced Features of Expert Analytics

You can configure the advanced features of Expert Analytics such as performance optimization and datatype support enablement for PAL algorithms using the SAPPredictiveAnalysis.ini file.

1. Close the SAP BusinessObjects Predictive Analytics application.

2. Navigate to `<SAPPA_INST_DIR>\Desktop`.
3. Open the SAPPredictiveAnalysis.ini file.
4. Set the values for the following parameters to true to enable the corresponding feature. Set the value to false to disable the feature.

Parameter	Description	Default Value
-Dpa.batch.sql	This parameter optimizes the performance of Expert Analytics using the batch execution of SQLs.	True
-Dpa.decimal.enabled	This parameter enables the decimal datatype support for PAL algorithms. The decimal datatype support is available from SAP HANA 71 and above.	False

5. Save and close the SAPPredictiveAnalysis.ini file.
6. Relaunch SAP BusinessObjects Predictive Analytics.

3 Acquiring Data

3.1 Data Acquisition Overview

Expert Analytics supports multiple data sources on which you can build your analyses.

Data source	Description
Microsoft Excel	Loads an Excel worksheet as a dataset
Text file	Loads a text file (`.csv` or `.txt`) as a dataset
Clipboard	Creates a dataset from data that was copied to the clipboard
SAP HANA	Datasets can be downloaded from SAP HANA for the analysis on your local machine
	Use SAP HANA as a platform to run your analyses on the SAP HANA server, with minimal data downloaded for reporting purposes
SAP BusinessObjects universe	Downloads data from SAP BusinessObjects universe files (`.unv` and `.unx`)
Query with SQL	Runs freehand SQL on a database, to download a dataset

When acquiring data, the application displays a preview of it, parses the data, and analyzes the columns to determine their data type and uniqueness.

> **Note**
>
> Your input table for Expert Analytics must contain unique column names. The application does not consider columns with same names, but different letter cases, as unique.

Objects representing columns are proposed as either dimensions or measures. You can manually hide some types of columns, based on the column name and data properties.

> **Note**
>
> When not connected to SAP HANA, the maximum number of cells that can be acquired is determined by the capacity of your computer. A warning message displays when an acquisition includes 30 million cells for 64-bit operating systems.

Depending on the data source, data can be adapted before acquisition to include or remove columns, dimensions, measures, variables, and input parameters. Some data sources have additional options, such as formatting data, naming and trimming columns, and specifying column-name prefixes.

Related Information

Acquiring data from a text file [page 22]
Acquiring data from SAP HANA views [page 25]
Acquiring data from an Excel workbook [page 20]
Acquiring data using Query with SQL [page 31]
Connecting to a universe data source [page 29]
Editing an acquired data set [page 37]
Objects hidden from the object list [page 38]

3.1.1 Viewing a data source connection and its associated documents

You can view all connections defined for the application, and the documents associated with each connection, and change the target data source for locally defined connections.

1. Close any open documents.
2. From the left-side menu, select *Connections*.
   The *CONNECTIONS* pane appears on the right and lists all available data source connections. Select a connection to display a list of documents associated with it. The *DOCUMENT FOR* pane appears to the right of the *CONNECTIONS* pane and lists the documents associated with each connection.
3. Select a local connection in the list to display its target data source.
   You can select another data source if required.
4. (Optional) To change the data source connection for a document, perform the following actions:
   a. Select the connection to change.
   b. Select the document to change the data source for in the *DOCUMENT FOR* pane.
   c. Select a new data source for the document in the *CONNECTIONS* pane, and select *Apply*.

3.1.2 Acquiring data from an Excel workbook

1. On the *Home* page, select *Acquire Data*.
2. In the *Add new dataset* dialog, select *Microsoft Excel*, and select *Next*.
3. Choose one or more Excel files, and select *Open*.
   Data from the Excel files is previewed in the *Add new dataset* dialog.
4. (Optional) Modify the Excel options for acquiring data.
5. Select *Create*.

The *Visualize* room opens so you can start building charts and analyzing the data. If you want to modify the dataset first, switch to the *Prepare* room.

3.1.2.1 Add new data set dialog options for Excel

You can acquire data from one or multiple Microsoft Excel workbooks. You choose which rows and columns to acquire. You can also acquire data from cross tables.

Option	Description
Dataset Name	Enter a name for the new data set.
File(s)	Select the Excel workbooks that will be the data source for the new data set.
Sheet	When an Excel workbook contains multiple worksheets, select the worksheet to acquire for the data set.
Append all sheets	Select this check box to add all worksheets in the workbook to the data set. Common columns are appended, and different columns are added as new columns.
Set first row as column names	Select this check box to set the first row values in the worksheet as column names in the data set.
Table Header Type	Select *Standard Table (No Transformations)* or *Cross Table*.
Select All	Select this check box to add all columns in the worksheet to the data set.
Show record count	Select this check box to show the number of columns and the number of rows in the data set.
Advanced Options > *Show hidden columns*	Select this check box to display hidden worksheet columns as column headers in the data set.
Advanced Options > *Show hidden rows*	Select this check box to display hidden worksheet rows in the data set.
Advanced Options > *Detect merged cells*	Select this check box to highlight merged worksheet cells in the data set.
Advanced Options > *Range Selection*	When a worksheet contains one or more named ranges, select the range to apply to columns acquired for the data set. A data set is restricted to the columns defined in this range.
Advanced Options > *Column*	For cross tables, specify the number of columns to use for the left header.
Advanced Options > *Row*	Specify the number of rows to use for the top header.

3.1.2.2 Acquiring data from multiple Excel workbooks

When acquiring data from multiple Excel workbooks, the data format and data type must be the same in all of the workbooks.

1. On the *Home* page, select *Acquire Data*.
2. In the *Add new dataset* dialog, select *Microsoft Excel*, and select *Next*.
3. Choose one or more Excel files, and select *Open*.

   Data from the Excel files is previewed in the *Add new dataset* dialog.
4. (Optional) In the *Dataset Name* box, enter a name for the data set.
5. Beside *Files(s)*, select *Add Files*, and browse to and select the Excel spreadsheet to acquire data from.

   You can use wild cards to search for a spreadsheet name. By default, the first file in the path is considered the reference file to which data will be appended from other spreadsheets acquired.

   For example, enter **`C:\data\monthly updates\*.xls(x)`** to find all `.xls(x)` files in the path.
6. In the *Sheet* list, select a worksheet.

   This worksheet is the reference sheet that data from other worksheets will be appended to. The count of records is updated to reflect the number of records from all acquired data. A "Source file" column is added to the data set, listing each data source name. If you selected the *Append all sheets* check box, all worksheets in the Excel spreadsheet are added to the data set.

   Data from the worksheet appears in the preview pane of the *Add new dataset* dialog.
7. (Optional) To display hidden worksheet rows or columns in the data set, select *Advanced Options*.
8. (Optional) To display hidden worksheet columns in acquired data, select the *Show hidden columns* check box, and enter the column range to display in the *Range Selection* list.
9. (Optional) To display hidden worksheet rows in acquired data, select the *Show hidden rows* check box, and enter the row range to display in the *Range Selection* list.
10. Select *Create*.

The data is acquired and appears in the *Prepare* room.

3.1.3 Acquiring data from a text file

You can acquire data from one or more text files, if the data is stored with delimiters or in fixed-width columns. An example of a text file using delimiters is a comma-separated value (`.csv`) file.

A `.csv` file stores numbers and text in plain-text format. Each record consists of fields usually separated by a comma or a tab, and records are separated by line breaks. Here is an example of a `.csv` file, with data separated by commas:

```
"Product","Country","Year","Quantity","Margin"
"Skis","Italy","2013","1,297","1,929"
"Computers","China","2014","609","10,659"
```

Acquiring data from this `.csv` file results in five columns in the data set: "Product," "Country," "Year," "Quantity," and "Margin." Column 2, in this example, would contain the values "Country", "Italy", and "China".

Here is an example of a text file with the data stored in fixed-width columns:

```
Product   Country   Year      Quantity  Margin
Skis      Italy     2013      1,297     1,929
Computers China     2014      609       10,659
```

You can acquire data from multiple-file data sources. The files must have the same format and data type.

Option	Description
Dataset Name	The name of the data set.
File(s)	The file or files that contain the data for the new data set. You can import data from one or multiple files. To specify multiple files, separate the file paths in the *File(s)* field with semicolons, or select *Add Files* and choose one or more files to add to the selection.
Separator	Choose whether data in your files is separated by delimiters or is entered in fixed-width columns. Delimiters are symbols, such as commas, tabs, or spaces, that separate fields in the data source and that will specify columns in the data set in SAP Lumira.
Set first row as column names	Select this check box to use the first row of data as column names in the data set. Clear this check box to use the default column names ("Column1", "Column2", and so on).
Advanced Options > *Number format*	The format for numeric columns in the data set.
Advanced Options > *Date format*	The format for date columns in the data set.
Advanced Options > *Break Column*	When acquiring data stored as fixed-width columns, analyze the data file and suggest column widths (in characters) for separating data into columns in the data set. If the suggested widths aren't suitable, you can change the widths by entering values separated by commas. For example, if your data is in three columns and the column widths are five, 10, and 15 characters, you would enter `5,10,15` in the *Break Column* box, and select *Apply* to see a preview of the resulting data set.
Advanced Options > *Trim leading spaces*	Select this check box to remove leading and trailing values from numbers and text in the data set so that column headers do not appear as empty fields. For example, if a "Product" entry has a leading space (" Product"), the space is removed and "Product" appears as the column header.

1. On the *Home* page, select *Acquire Data*.
2. In the *Add new dataset* dialog, select *Text*, and select *Next*.
3. Choose one or more text files, and select *Open*.
   Data from the files is previewed in the *Add new dataset* dialog.
4. (Optional) Adjust the data set options in the dialog as needed.
5. Select *Create*.

The *Visualize* room opens, and you can start building charts and analyzing data. If you want to modify the data set first, switch to the *Prepare* room.

3.1.4 Acquiring data copied to the clipboard

Text-based data can be copied to the clipboard from a text-based file (for example, from Microsoft Excel) or from a web page.

Option	Description
Dataset Name	The name of the data set
Separator	Choose whether data on the clipboard is separated by delimiters or is entered in fixed-width columns. Delimiters are symbols, such as commas, tabs, or spaces, that separate fields in the data source and that will specify columns in the data set in the application.
Set first row as column names	Select this check box to use the first row of data as column names in the data set. Clear this check box to use the default column names ("Column1", "Column2", and so on).
Advanced Options > *Number format*	The format for numeric columns in the data set.
Advanced Options > *Date format*	The format for date columns in the data set.
Advanced Options > *Break Column*	When acquiring data stored as fixed-width columns, analyze the data file and suggest column widths (in characters) for separating data into columns in the data set. If the suggested widths aren't suitable, you can change the widths by entering values separated by commas. For example, if your data is in three columns and the column widths are five, 10, and 15 characters, you would enter `5,10,15` in the *Break Column* box, and select *Apply* to see a preview of the resulting data set.
Advanced Options > *Trim leading spaces*	Select this check box to remove leading and trailing values from numbers and text in the data set so that column headers do not appear as empty fields. For example, if a "Product" entry has a leading space (" Product"), the space is removed and "Product" appears as the column header.

> **Note**
>
> The Microsoft Internet Explorer (IE) web browser has a known issue when copying text to the clipboard. If you encounter this issue, use a different supported browser instead.

1. Copy text to the clipboard.
2. On the SAP Lumira *Home* page, select *Acquire Data*.
3. In the *Add new data set* dialog, select *Copy from Clipboard*, and select *Next*.

The text you copied is pasted in the dialog.

4. (Optional) Select *Trim Spaces* to remove leading and trailing spaces from numbers and text in the data set.
5. (Optional) Select *Trim Row* to remove blank lines from the data set.
6. Select *Proceed*.
   Data from the files is previewed in the *Add new dataset* dialog.
7. (Optional) Adjust the data set options in the dialog as needed.
8. Select *Create*.

The *Visualize* room opens, and you can start building charts and analyzing data. If you want to modify the data set first, switch to the *Prepare* room.

3.1.5 Acquiring data from SAP HANA views

You can acquire data from SAP HANA analytic or calculation views.

Data in an SAP HANA database is accessible in a "view"—a predefined virtual grouping of table columns that enables data access for a particular business requirement. Views are specific to the type of tables that are included, and to the type of calculations that are applied to columns. For example, an attribute view is built on dimension tables, an analytic view is built on a fact table and attribute views, and a calculation view executes a function on columns when the view is accessed.

You can connect to SAP HANA views in two ways:

- By downloading data from SAP HANA
  Data is copied locally and can be manipulated and edited before being visualized in charts.
- By viewing data in SAP HANA
  Data is read-only (that is, you cannot edit it), but you can visualize it in charts.

After connecting to a view, data is presented as columns, facets, measures, dimensions, and hierarchies in the application.

Related Information

Downloading data from SAP HANA [page 27]
Connecting to SAP HANA [page 26]
Accessing SAP BW data in SAP HANA views [page 28]
Specifying values for SAP HANA variables and string input parameters [page 28]

3.1.5.1 Restrictions for SAP HANA connections

Restriction	Description
Only one level is available for geographical hierarchies.	Only one attribute at a time can be used when creating a geographical hierarchy.
Measures with numeric or string dimensions cannot be created.	Measures are detected from the SAP HANA analytic view. They must be created in the SAP HANA view before the application can automatically acquire them.
Datasets cannot be published to SAP HANA.	
Some functions are not supported.	The following SAP HANA functions are not supported: • *AddMonthToDate* • *AddYearToDate* • *LastDayOfMonth* • *DayOfYear* • *Week* • *LastWord* • *ExceptLastWord*
Some features are not available for analytic views that use a calculation view.	When an analytic view uses a calculation view (for example, when an attribute view within the analytic view has a calculated measure or one or more calculated columns): • The grid view is not available in the *Prepare* room. • When a measure is selected in the *Prepare* room, facets show no values. • Sorting on a measure is not possible in the *Visualize* room.
Other data sources cannot be used along with a Connect to SAP HANA data source.	When a Connect to SAP HANA data source is used in an SAP Lumira document, it is not possible to acquire data from other data sources.

3.1.5.2 Connecting to SAP HANA

While connected to SAP HANA, you can view data and create visualizations from an SAP HANA cube.

You must know your SAP HANA server name, port number, user name, and password. For more information, contact your SAP HANA administrator.

1. On the *Home* page, select *Acquire Data*.
2. In the *Add new dataset* dialog, select *Connect to SAP HANA One*, and select *Next*.
3. Select the server to log on to in the *Server* list.
4. Enter the port number to log on to in the *Instance/Port* box.

5. Connect to the SAP HANA server:

Option	Description
If you connect by entering credentials	Enter your user name and password, and select *Connect*.
If you connect via single sign on (SSO)	Select the *Authenticate by Operating System (SSO)* check box, and select *Connect*.

6. Select *Next*.
   An *Add new dataset: SAP HANA views* dialog appears, displaying available SAP HANA views.
7. Expand the SAP HANA view that contains the data to view, and choose the cube that contains the data.
8. Choose which data to acquire:
   - To acquire particular dimensions and measures in the data, select *Next*, choose the dimensions and measures, and select *OK*.
   - To acquire all data, select *Create*.

The *Visualize* room opens, and you can start building charts and analyzing the data. If you want to modify the data set first, switch to the *Prepare* room.

Related Information

Specifying values for SAP HANA variables and string input parameters [page 28]
Restrictions for SAP HANA connections [page 26]

3.1.5.3 Downloading data from SAP HANA

You can manipulate data and create visualizations from an SAP HANA cube.

You must know your SAP HANA server name, port number, user name, and password. For more information, contact your SAP HANA administrator.

1. On the *Home* page, select *Acquire Data*.
2. In the *Add new dataset* dialog, select *Download from SAP HANA One*, and select *Next*.
3. Select the server to log on to in the *Server* list.
4. Enter the port number to log on to in the *Instance/Port* box.
5. Connect to the SAP HANA server:

Option	Description
If you connect by entering credentials	Enter your user name and password, and select *Connect*.
If you connect via single sign on (SSO)	Select the *Authenticate by Operating System (SSO)* check box, and select *Connect*.

6. Select *Next*.
   An *Add new dataset: SAP HANA views* dialog appears, displaying available SAP HANA views.

7. Expand the SAP HANA view that contains the data to view, and choose the cube that contains the data.
8. Choose which data to acquire:
    - To acquire particular dimensions and measures in the data, select *Next*, choose the dimensions and measures, and select *OK*.
    - To acquire all data, select *Create*.

The *Visualize* room opens, and you can start building charts and analyzing the data. If you want to modify the data set first, switch to the *Prepare* room.

Related Information

Acquiring data from SAP HANA views [page 25]
Specifying values for SAP HANA variables and string input parameters [page 28]

3.1.5.4 Accessing SAP BW data in SAP HANA views

You can access SAP Business Warehouse (BW) data that is available in SAP HANA analytic or calculation views.

In an SAP BW-on-SAP HANA system, you can use the SAP HANA modeler to import SAP BW models (for example, SAP HANA-optimized cubes, Data Store Objects (DSO), and BW Query Snapshots) as analytic views and calculation views. Once the models are activated, the application can consume them by connecting to an SAP HANA cube.

For information about implementing an SAP BW-on-SAP HANA system and about making data available to SAP HANA, see the *FAQ: BW on HANA* document at http://www.experiencesaphana.com/community/solutions/net-weaver-bw/bwonhanafaq.

Related Information

Downloading data from SAP HANA [page 27]

3.1.5.5 Specifying values for SAP HANA variables and string input parameters

You are prompted to enter a value for an SAP HANA variable or a string input parameter when acquiring an analytic view in *Download from SAP HANA One* mode and when creating a document on an analytic view in *Connect to SAP HANA One* mode.

Each SAP HANA variable defines a filter on a dimension of a view. You enter a value for each dimension before data is acquired, and the value appears as a facet row after acquisition.

You enter a value for each SAP HANA input parameter when acquiring data, and SAP Lumira passes the value to a calculation, such as a formula for a calculated measure. When entering a value for a string input parameter, you must enter an SQL statement, using single quotes to indicate the beginning and end of the statement string. For example, enter `BUKRS='CALP'` to search for *CALP*.

1. Connect to an SAP HANA instance in *Download from SAP HANA One* or *Connect to SAP HANA One* mode.
2. Choose which data to acquire:
   - To acquire particular data, select an analytic view, select *Preview and select data*, select *Select*, choose the dimension values and measures, and select *Edit Variables*.
     When no variables or input parameters are defined in a view, the *Edit Variables* button is not available.
   - To acquire all data available in an analytic view, choose the view, and select *Create*.

   A *HANA Variables* box appears, listing the variables and input parameters defined for the analytic view. Variables are prefixed by "VAR" and input parameters are prefixed by "IP."
3. Choose a variable or an input parameter.
   The dimension or input parameter value appears in the right pane.
4. Choose one or more values, and select *Add*.

   To choose multiple individual values, press and hold Ctrl and select each value. To choose a range of values, press and hold Shift and select the first and last value in the range.

   The selected values appear in the bottom pane.
5. Select *OK*.
6. If you are acquiring data through *Preview and select data*, select *Create* to start the data acquisition.
   Data appears in the *Prepare* room, and each variable appears as a facet with the selected prompt values.

The *Variables* button appears at the top of the facets pane. Select the button to view the values you chose for SAP HANA variables.

3.1.6 Acquiring data from universes

You can acquire data from SAP BusinessObjects universe files.

Universe (`.unx`) files for SAP NetWeaver BW access are created with the Information Design Tool that is installed with SAP BusinessObjects Business Intelligence platform. For information about the Information Design Tool, see http://help.sap.com/businessobject/product_guides/sbo41/en/sbo41sp1_info_design_tool_en.pdf.

For information about data federation, see http://help.sap.com/businessobject/product_guides/sbo41/en/sbo41_dfat_guide_en.pdf.

For a complete list of supported universes and database access options, see the *Product Availability Matrix* on the SAP Service Marketplace at SAP Product Availability Matrix.

3.1.6.1 Connecting to a universe data source

1. On the *Home* page, select *Acquire Data*.

2. In the *Add new dataset* dialog, select *Universe*, and select *Next*.
3. In the *Universe credentials* pane:
   a. Enter the name or IP address of the server that hosts your Central Management Server (CMS).

      If you are connecting to a CMS that belongs to a different network domain, make sure the `hosts` file located at `C:\Windows\System32\drivers\etc` has the corresponding host name entry (for example, `x.x.x.x<NameOfMachineHostingCMS>`).
   b. Enter the CMS user name, password, and authentication type.
   c. To use Windows AD authentication to connect to the CMS, append the following entries in the`SAPLumira.ini` file, located at `<LumiraInstallDir>\SAPLumira\Desktop`:

      ```
      -Djava.security.auth.login.config=<Path_to_bscLogin>\bscLogin.conf
      -Djava.security.krb5.conf=<Path_to_kbr5>\krb5.ini
      ```

      ```
      -Djava.security.auth.login.config=C:\Windows\bscLogin.conf
      -Djava.security.krb5.conf=C:\Windows\krb5.ini
      ```
   d. Select *Connect*.

   A list of universes available in the CMS appears.
4. Choose a universe, and select *Select*.
5. Choose the required objects in the universe tree, and select *Acquire*.

   Before acquiring data, you can preview the data and apply filters by selecting the *Preview and Select Data* option. If a query contains contexts or prompts, you must respond to them before data can be acquired. When creating a query, you can set the following query properties:
   - *Max Rows Retrieved*: The maximum number of rows to be retrieved by the query.
   - *Max Retrieval Time*: The maximum amount of time a query can run (in seconds).
   - *Retrieve Duplicate Rows*: Select to retrieve duplicate rows.

The *Visualize* room opens, and you can start building charts and analyzing the data. If you want to modify a data set first, switch to the *Prepare* room.

3.1.6.2 Troubleshooting messages about universe data connections

You may encounter these messages and possible causes while working with the universe data source.

Message	Cause
`Could not connect to Central Management Server (CMS)`	• The CMS is unresponsive. • Your user name or password is incorrect. • The authentication type is incorrect. • A network issue has occurred.
`Could not load the selected universe`	• The universe is corrupted. • The CMS is unresponsive. • The universe connection is not configured properly.

Message	Cause
`Could not validate the query`	• There is an issue with the database connection. • The data types do not match the object. • A result returned from the server has reached the limit set for the `Maximum Character Stream Size (MB)` configuration parameter in the SAP Web Intelligence Report Server. • One or more universe objects are not configured properly.
`Query returns no row sets`	A query for the object returned no data.

3.1.7 Acquiring data using Query with SQL

You can create a data provider by manually entering the SQL for a target data source. You can specify the source tables, columns, and functions used to acquire data.

For a complete list of database middleware that Query with SQL can access, see the SAP Product Availability Matrix.

Supported database middleware	How to obtain the driver
Apache	Amazon EMR and Apache Hive Simba drivers are included with SAP Lumira.
Cloudera	The Cloudera Impala Simba driver is included with SAP Lumira.
IBM DB2	Go to the IBM DB2 connectivity download page at https://www.ibm.com/account/profile/us?page=reghelpdesk. Choose the appropriate driver for your database, save the compressed installation file to your computer, extract the compressed file (`db2jcc.jar`) to a local directory, and run the installer from your computer. For versions earlier than 9.5, you must extract `db2cc.jar` and `db2jcc_license_cu.jar` instead. Before you can download a driver, you must register using a free IBM-recognized user email address as the account name. If you do not know which version of the driver to use, both drivers for DB2 version 10.1 [DB2 version 10.1 FP0 (GA) and version 10] are suitable for all versions later than DB2 version 9.5. For more information, contact your database administrator.
IBM Netezza	See your Netezza administrator.

Supported database middleware	How to obtain the driver
Microsoft SQL Server	Go to the SQL Server 2005, 2008, and 2012 Microsoft Drivers download center page at http://www.microsoft.com/en-us/download/driver.aspx?q=driver . Choose the appropriate driver for your database, save the installation file to your computer, and run the installer from your computer. If you don't know which version of the driver to use, Microsoft JDBC Driver 4.0 for SQL Server is suitable for all supported SQL server versions. If you are installing JDBC Driver 4.0 for SQL Server, the driver is `sqljdbc_4.0.2206.100_enu.exe` for a Windows operating system. The `sqljdbc4.jar` driver file is extracted to `\sqljdbc_4.0\enu\`, in the specified extraction folder.
Oracle	Go to the Oracle JDBC Driver Downloads page at http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html Before you can download a driver, you must create a free user account. If you don't know which version of the driver to use, `ojdbc14.jar` is suitable for any supported version of Oracle 10 and 11.
Sybase	The Sybase driver (`jconn4.jar`) is installed by default; you do not need to install it. It is located at `\\<InstallDir>\Program Files\SAP Lumira\Desktop\plugins\com.businessobjects.connectionserver.standalone_3.1.3.v20120603-0404\ConnectionServer\jdbc\drivers\IQ15`.
Teradata	Go to the Teradata connectivity download page at http://downloads.teradata.com/download/connectivity/jdbc-driver . Choose the appropriate driver for your database, save the compressed installation file to your computer, extract the compressed file to a local directory, and run the installer from your computer. Before you can download a driver, you must create a free user account. If you don't know which version of the driver to use, the Teradata JDBC Driver 14 is suitable for all supported Teradata versions. For Windows, use `TeraJDBC__indep_indep.14.00.00.14.zip`. Once extracted, the driver files are `tdgssconfig.jar` and `terajdbc4.jar`.

Database middleware	JDBC driver available
Oracle	`ojdbc14.jar`
Microsoft SQL Server	`sqljdbc4.jar`
Teradata	`terajdbc4.jar` and `tdgssconfig.jar`
Sybase	`jconn4.jar`
IBM DB2	`db2jcc.jar` or `db2cc.jar` and `db2jcc_license_cu.jar` for versions earlier than 9.5
IBM Netezza	`nzjdbc.jar`

Related Information

Installing data access drivers [page 33]
Connecting to a Query with SQL data source [page 34]
Query with SQL connection parameters [page 36]

3.1.7.1 Installing data access drivers

The Sybase IQ 15 data access driver is automatically installed with the application. For other databases, you may need to install the JDBC data access driver for your database middleware, before using Query with SQL.

- You must be familiar with your database and with the SQL language.
- The correct data access driver must be installed for your database middleware. A data access driver is the software provided by a database vendor that allows a client application to connect to middleware and to access data in a database. You copy the data access driver for your middleware from your database vendor support web site to a local folder, and then you can select the driver in the application and connect to the database.

> **Note**
>
> Installing data access drivers from a vendor site can be problematic due to the variety of driver versions and file formats. If you are unfamiliar with your database version or the vendor web site, contact your database administrator.

Follow these general steps to obtain a data access driver:

1. Download the data access driver (a `.jar` file) from the database vendor site, and copy the file to a local folder.
2. Register the driver path by selecting the driver in the application.
3. Select a Query with SQL data source on the *SQL Drivers* tab in the application preferences.
   You can select an installed SQL driver or install the required driver.

1. Select *File* > *Preferences* > *SQL Drivers*.
   The *Driver Installation* page lists database middleware names and the status of drivers:
   - When the status check mark is green, the driver is correctly installed and you can start using Query with SQL.
   - When the status check mark is red, the driver is not installed for that middleware and you must install it.
   - When the status check mark is yellow, a compatible driver is available for the middleware, but the application must be restarted before it is available. Once the software has restarted, you can use Query with SQL.
2. Choose a data source, and perform one of the following actions:

Option	Description
If the data source middleware has a green check mark	Select *Next*, enter the middleware connection information, and select *Create*.

Option	Description
	The data access drive is installed. You do not need to perform the remaining steps in this task.
If the data source middleware has a yellow check mark	Restart the application, and repeat step 1.
If the data source middleware has a red check mark	Go to step 3.

3. If the middleware driver is not configured, select the *Install* button, choose the database driver, and select *Install Drivers* at the top of the database list.
4. In the selection box for locally available middleware:

Option	Description
If the correct `.jar` file is listed	Go to step 6.
If the correct `.jar` file is not listed	Download the driver from the vendor support site, install the driver, select *Cancel* to close the driver selection box, and then download and install the correct `.jar` file.

   You must access the web page that lists JDBC data access drivers for the middleware vendor. Depending on the database, different types of driver files are available; usually a compressed file containing the drivers or an executable file to install the drivers automatically. For the application, download only the compressed file.

5. On your vendor's support web site, download the compressed JDBC driver file (for example, a `.tar`, `.gz`, or `.zip` file) for your database middleware version.
6. On your computer, select the folder that contains the extracted JDBC driver files for your database middleware.

   A complete list of supported JDBC drivers is included in the Product Availability Matrix, available on the SAP Service Marketplace site at https://support.sap.com/pam.
7. Restart the application.
   The list of available database middleware drivers is updated.

When you use Query with SQL to create a new document in the application, the target database middleware is listed with a green check mark, indicating that the driver is available to access the database.

3.1.7.2 Connecting to a Query with SQL data source

You can connect directly to a database to specify the data to acquire and to set parameters to optimize the database connection.

- You must be familiar with your database and with the SQL language.
- The correct data access driver must be installed for your database middleware. A data access driver is the software provided by a database vendor that allows a client application to connect to middleware and to access data in a database. You copy the data access driver for your middleware from your database vendor support web site to a local folder, and then you can select the driver in the application and connect to the database.

> **Note**
>
> Installing data access drivers from a vendor site can be problematic due to the variety of driver versions and file formats. If you are unfamiliar with your database version or the vendor web site, contact your database administrator.

You need to install a JDBC data access driver for your database middleware before using Query with SQL. The data access driver is a `.jar` file that you download from a database vendor site and copy to the driver folder in the application installation path. Refer to the Related Information about finding and installing the correct data access driver for your database middleware.

1. On the *Home* page, select *Acquire Data*.
2. In the *Add new dataset* dialog, select *Query with SQL*, and select *Next*.
   A list of database middleware appears.
   - When a green check mark appears next to the middleware name, the middleware is installed and you can start using Query with SQL.
   - When a red cross appears next to the middleware name, the data access driver for the middleware is not installed; you must install it.
3. Choose the database middleware for the target database:

Option	Description
If the middleware is available	Select the middleware, and select *Next*.
If the middleware is not available	Select *Install*, and install the middleware.

4. Enter your logon credentials on the *Login* tab and, if necessary, enter driver parameters on the *Advanced* tab.
5. Select *Create*.
   The SQL editor opens.
6. Enter the SQL to fetch the required tables, preview the SQL query, and select *Create*.
   The *Visualize* room appears, and you can start building charts and analyzing data. If you want to modify the data set first, switch to the *Prepare* room.

Related Information

Query with SQL connection parameters [page 36]
SQL editor options for Query with SQL [page 36]
Installing data access drivers [page 33]

3.1.7.2.1 SQL editor options for Query with SQL

Use an SQL editor to write SQL and create a Query with SQL data source, based on a connected database. The SQL editor is accessed from the Query with SQL connection option when you create a new document.

Only the `SELECT` statement is authorized in the SQL editor to acquire data from database tables. Use these SQL editor options to select tables for the data source:

Option	Description
Catalog	The accounts available to the connected database. Expand each node to see the tables available. Double-click a table to add the table to the SQL query.
Query	The `SELECT` query to fetch tables. (Only `SELECT` is supported.) You can add table names by double-clicking the table in the account node in the left pane.
SQL History	Keep a log of the `SELECT` statements used in the query pane. Choose a statement to include it in the query.
Preview data	Select this option to preview the tables that are acquired by `SELECT`.
Select All/None	Choose all or no columns, or choose individual columns for acquisition.

Related Information

Connecting to a Query with SQL data source [page 34]

3.1.7.3 Query with SQL connection parameters

You can create your own data provider by manually entering the SQL for a target data source to acquire table data. When using Query with SQL, you must enter connection information for the target database, and you can specify connection parameters to optimize the fetching of data.

Parameter	Description
`User name`	The user name that you use to connect with the target database
`Password`	The password that you use to connect with the target database
`Server (<host>:<port>)`	The name and port of the server hosting the database
`Database`	The database name

Parameter	Description
`Connection pool mode`	If using a connection pool, use to keep the connection pool mode connection active.
`Pool timeout`	If the connection pool mode is set to *Keep the connection active for*, the length of time in minutes to keep the connection open.
`Array fetch size`	The maximum number of rows authorized with each fetch from the database. For example, if you enter 20, and your query returns 100 rows, the connection retrieves the data in five fetches of 20 rows each. To deactivate array fetch, enter an array fetch size of 1. Data is retrieved row by row. Deactivating the array fetch size can increase the efficiency of retrieving your data, but it slows server performance. The greater the value in the array fetch size, the faster your rows are retrieved. However, ensure that you the client system has adequate memory.
`Array bind size`	Size of the bind array before it is transmitted to the database. Generally, the larger the bind array, the more rows (n) can be loaded in one operation, and performance will be optimized.
`Login timeout`	The number of minutes before a connection attempt times out and a message appears.
`JDBC driver properties`	Values for JDBC driver properties. You can define the value of more than one property, separated by commas. For example, the `oracle.jdbc.defaultNChar=true,defaultNChar=true` value for JDBC driver properties sets the `oracle.jdbc.defaultNChar` and `defaultNChar` driver properties.

3.1.8 Editing an acquired data set

After a data set has been acquired, you can edit it.

You can edit this information in acquired data sets:

- Add new columns that were removed from the data source when it was originally acquired
- Remove columns that were included in the original data source
- Change values selected for SAP HANA variables and input parameters

1. Open a data set that is already acquired in the application.
2. Select *Data* > *Edit Data Source*.
3. Perform one or more of the following actions:
    - Select a column name check box to add a new column.
    - Clear a column name check box to remove a column.
    - Select or clear check boxes to add or remove dimensions and measures.
    - To change SAP HANA variables and input parameters, select *Edit Variables*, enter or delete values for variables or input parameters, and select *OK*.
4. Select *OK*.

The data set is updated with the added or removed columns, dimensions, measures, or variables.

3.1.9 Refreshing data in a document

The data that is saved with a document can become stale or invalid. Refresh the document to get fresh data from the data source.

With a document open, select *Data* > *Refresh document*.

3.1.10 Renaming a data set

You can rename a data set in the *Prepare*, *Visualize*, and *Share* rooms.

1. Open a data set.
2. Perform one of these actions:
   - In the *Prepare* or *Visualize* room, select the data set name.
   - In the *Share* room, select the cogwheel beside the data set name and select *Rename*.
3. Change the name of the data set.
4. Press `Enter`, or select an area outside the data set name field.

3.1.11 Objects hidden from the object list

Use the enrichment suggestions file to prevent specific columns from being proposed as measures in the application when data is acquired.

To prevent specific columns from being proposed as measures when data is acquired, the application uses the `enrichment_suggestions.<VersionNumber>.txt` file to identify columns that should not be proposed as measures.

By default, column names in the enrichment suggestions file are in English. However, you can define names in other languages, specify column names to hide from the objects list, and prevent objects from being considered time or geographical objects. The enrichment will be processed if you selected automatic detection of enrichments in the application preferences.

When you upgrade the application, a new version of the enrichment suggestions file is saved, without overwriting the original file. You can use the original file as a reference to modify the new suggestions file. The application will use the file name that corresponds with the installed version of the application.

3.1.12 Editing the enrichment suggestions file

Edit the enrichment suggestions file to identify which objects should not be proposed as measures or as time or geographic attributes on the *Measures and Dimensions* panel.

You can define as many rules as you require in the `enrichment_suggestions.<VersionNumber>.txt` file. For each rule, you must declare four properties. The syntax is Java regex and metadata is not case-sensitive.

Required property	Description
`objectName`	Pattern matching on the object name (column header). Any character can be used. When `.*DAY.*` is used, any object containing the string `DAY` is included in the rule (MONDAY, TUESDAY, and so on).
`dataType`	List of data types. Recognized data types are: • `integer` • `biginteger` • `double` • `string` • `date` • `boolean` Any column name, with any data type, will be considered for exclusion from the proposal pane. If no `dataType` property is declared, all data types are considered.
`enrichment`	Prevents objects from appearing. The values are `MEASURE` or `TIME` (time hierarchy objects) or `GEO` (geographic hierarchy objects).
`rule`	Defaults to `hide`. Do not change this value.

The following example shows the default enrichment file:

```
{
 "version":"1.0",
 "policies":{
    },
    "suggestionRules":[
    {
      "objectName":"(?i).*year.*|.*month.*|.*quarter.*
                   |.*week|.*day|.*semester.*|.*hour|.*minute|.*second",
      "dataTypes":["integer", "biginteger", "double"],
      "enrichment":"MEASURE",
      "rule":"hide"
    },
    {
      "objectName":"(?i).*zip.*",
      "dataTypes":["integer", "biginteger", "double"],
      "enrichment":"MEASURE",
      "rule":"hide"
    },
    {
      "objectName":"(?i).*_id\\d*",
      "dataTypes":["integer", "biginteger", "double"],
      "enrichment":"MEASURE",
      "rule":"hide"
    },
    {
      "objectName":"(?i).*key.*",
      "dataTypes":["integer", "biginteger", "double"],
      "enrichment":"MEASURE",
      "rule":"hide"
    },
    {
      "objectName":"(?i).*zip.*",
```

```
      "dataTypes":["integer", "biginteger", "double"],
      "enrichment":"TIME",
      "rule":"hide"
    },
    {
      "objectName":"(?i).*_id\\d*",
      "dataTypes":["integer", "biginteger", "double"],
      "enrichment":"TIME",
      "rule":"hide"
    },
    {
      "objectName":"(?i).*key.*",
      "dataTypes":["integer", "biginteger", "double"],
      "enrichment":"TIME",
      "rule":"hide"
    }
    ]
}
```

1. Open the `enrichment_suggestions.<VersionNumber>.txt` file in a text editor.
2. For each object, define the data type(s), enrichment, and other properties as needed.

   You must keep `"rule"` set to `"hide"`.
3. Save the file with the same name.

4 Preparing Data

4.1 Data Preparation Overview

When data is first acquired by the application, it is raw data that is often formatted inconsistently and is not easily interpreted by business users. Before creating charts to visualize your data, it is often necessary to prepare the data so that it is presentable and understandable.

Data preparation can be done in either *Grid* or *Facets* view, using the *Manipulation Tools* panel at the right side of the *Prepare* room.

Editing tasks can be applied to all values in a column or to selected values.

Related Information

Editing and cleaning data [page 47]
Converting data to another type [page 49]
Creating a geography or time hierarchy [page 50]
Creating a measure from a column or dimension [page 53]
Adding a data set [page 71]
Switching to another data set [page 71]
Merging data sets (JOIN) [page 71]

4.1.1 Prepare room—viewing, cleaning, and manipulating data

Before creating charts, use the tools in the *Prepare* room to view and prepare data.

The *Prepare* room displays data for the connected data source and is divided into three areas.

Area	Description
Data pane	The central area that displays data in rows and columns (*Grid* view) or in facets (*Facets* view). It is where you view data and can apply the following tools (when available) to column values: • *Change Aggregation* • *Sort* • *Filter* • *Display Formatting* • *Convert To Number* • *Convert To Date* • *Convert To Text* • *Create a measure* • *Create a time hierarchy* • *Create a geographic hierarchy* • *Create a custom hierarchy* • *Rename* • *Remove* • *Merge the column* • *Hide column* • *Fit to content* • *Create Calculated Dimension* • *Duplicate*
Measures and Dimensions panel	A panel located to the left of the data area that lists the measures and dimensions the application detected in the data. Use tools on the *Measures and Dimensions* panel to define and to edit measures and to create time and geography hierarchies.
Manipulation Tools panel	A panel located to the right of the data area, where you can edit text and convert values in a cell or column of data, create new columns with formulas, and rename, duplicate, and remove columns.

Related Information

Measures and Dimensions panel [page 43]
Data pane [page 43]
Manipulation Tools panel [page 45]
Data Preparation Overview [page 41]

4.1.1.1 Measures and Dimensions panel

The *Measures and Dimensions* panel is located to the left of the Chart Canvas. It lists the measures, dimensions, hierarchies, and inferred dimensions in a data set.

Object	Description
Measures	A map to aggregated data in a column or calculation. You use measures to get a calculated result when columns are combined. For example, a measure called Sales Revenue would represent the column Sales Revenue that contains the summed revenue for sales. Measures are automatically detected and listed.
Dimensions	A data object that represents categorical data in a data set.
Hierarchies	A reference to more than one related column in a data set; the columns have hierarchical relationships. For example, an object Time could include Year, Quarter, and Month columns arranged in a hierarchical structure under the top object Time.
Attributes	Maps to a column in a data set.
Inferred dimensions	One or more columns created from geography or time data that is available to the application (to support a hierarchy).

Related Information

Data Preparation Overview [page 41]
Creating a geography or time hierarchy [page 50]
Creating measures [page 53]
Creating measures and hierarchies [page 49]

4.1.1.2 Data pane

The *Data* pane is the central pane that shows your data in the *Prepare* room.

Use the *Data* pane to view, organize, edit, and prepare datasets for visualizations.

Option	Description
Data source selection	Lists the data sources connected in the current session. You can use the list to toggle between datasets and to add datasets.
Data filters	Lists the filters that are applied to column data. You can edit or remove the filters in the list.

Option	Description
Grid	Select  to display data in columns and rows. All rows are displayed.
Facets	Select  to display only unique values in data, in columns. (Repeated values in columns are not shown.) Using facets can be helpful when a data set includes many repeated values.
Show/Hide columns	Select  to show or hide columns in a data set.
Calculation	Select  to add calculated dimensions or measures.
Combine	Select  to merge or append data to a data set. You can merge data from multiple datasets into the current data set, but the data must be compatible. You can append another data set to the current one. Data in common columns is appended to the current data set, and data in unique columns is added in new columns.
Refresh the document data	Select  to refresh the data set(s) used in the document.
Undo	Select  to reverse the last action.
Redo	Select  to repeat the last action.

Related Information

Editing and cleaning data [page 47]
Filtering data [page 47]
Adding a data set [page 71]
Merging data sets (JOIN) [page 71]
Switching to another data set [page 71]
Creating a calculated measure or dimension [page 54]

4.1.1.3 Manipulation Tools panel

The *Manipulation Tools* panel is located on the right side of the *Prepare* room and contains tools for editing and formatting values.

Depending on the data type of the selected column, you can use the panel to perform the following tasks:

- Duplicate, rename, and remove columns
- Create calculated dimensions
- Find, replace, and change string values
- Fill in prefixes and suffixes
- Convert, trim, and group values
- Edit text strings

4.1.1.3.1 Data actions for columns

Data actions for columns containing characters, dates, and/or numbers are listed on the *Manipulation Tools* panel. The actions that are available depend on the type of data in the column.

To show the data actions available for a column, select the  icon next to the column name or right-click the name.

Option	Description	Available from menu in column header	Available for		
			Characters	Dates	Numbers
Duplicate	Inserts a new column that is a copy of this column.	Yes	Yes	Yes	Yes
Rename	Changes the name of this column to a specified name.	Yes	Yes	Yes	Yes
Split	Divides this column after a specified split point and moves all string values after that point to a new column. The split can be a punctuation mark (for example, a comma) or a text string.	No	Yes	No	No
Remove	Removes this column.	Yes	Yes	Yes	Yes
Convert Case	Converts text in this column to lowercase or uppercase.	No	Yes	No	No
Replace	Finds a specified string in this column and replaces it with another specified string.	No	Yes	No	No

Option	Description	Available from menu in column header	Available for		
			Characters	Dates	Numbers
Fill	Prefixes or suffixes a specified string with a specified character, to a specified length.	No	Yes	No	No
Convert to Text	Converts all values in this column to text.	Yes	No	No	Yes
Convert to Number	Converts all values in this column to numbers.	Yes	Yes	No	No
Convert to Date	Converts all values in this column to dates in the selected format.	Yes	Yes	No	No
Trim	Removes characters in this column before or after a specified punctuation mark or character.	No	Yes	No	No
Group by Selection	Creates a group for the values selected in this column.	No	Yes	Yes	Yes
Group by Range	Creates a group for a specified range of values in this column.	No	Yes	Yes	Yes
Create Calculated Dimension	Creates a new column and applies a specified function to values in the new column. For example, a "Floor" function can be applied to a "Margin" column to create a new column of margin values, rounded down to the nearest whole number.	No	No	Yes	Yes
Cell inner selection	In the *Grid* or *Facets* view: • Removes text in a specified word or range of characters in this column • Replaces text in a specified word or range of characters in this column • Creates a new column with data copied from this column and cleaned (for example, with "resort" removed from the data) • Moves specified text to the beginning of each row value in this column	No	Yes	No	No
Concatenate	Joins two or more columns, with an optional specified separator and name for the merged column. *Concatenate* options become available when you select two or more columns.	No	No	No	No

4.1.2 Editing and cleaning data

You use the *Manipulation Tools* panel to edit and format values in a column. The panel is available in the *Grid* and *Facets* views.

Some data actions on the *Manipulation Tools* panel are also accessible by selecting the  icon in a column name or by right-clicking the column name.

1. Perform one of the following actions:

Option	Description
To edit all values in a single column	Select the column header.
To edit all values in multiple columns	Press `Ctrl` and select each column header.
To edit an individual value	Select a cell.
To edit multiple values in a column	Press `Ctrl` and select each cell.
To edit a range of characters or a word within a cell (cell inner selection)	(Character values only) Double-click in the *Grid* view or slow double-click in the *Facets* view, and select a range of characters or a word.

2. Open the *Manipulation Tools* panel to the right of the *Data* pane.
   Unique column values appear in a *Values* box at the top of the panel. You can select one or more values to edit in this box, or enter a search string in the *Find* box. Selections in the editor panel override the value selections made directly in a column. The data actions available for a column depend on the data type of the column and on whether a column, cell, or range of characters within a cell is selected.
3. Select an editing option on the *Manipulation Tools* panel, modify the values as needed, and select *Apply*.

4.1.2.1 Filtering data

A filter is a restriction imposed on a data set to limit the values displayed. You create filters by choosing values or ranges of values from a dimension to include or exclude.

You can filter data in an entire data set or in a single visualization. Filters applied to a data set affect any chart that uses the data. However, filters applied to a visualization affect only the current chart (not the entire data set).

In the *Prepare* room, you can add or edit data set filters. All of the filters that are defined on the data set appear in the filter bar at the top of the *Data* pane.

In the *Visualize* room, you can work with filters applied to the data set as well as filters on the current visualization.

Example

If a data set includes data on revenue for products sold over years between 1995 and 2012 and you only want to analyze revenue data for the years 2010 to 2012, you can create a data set filter on the dimension Year to limit the values shown to this period.

Related Information

Filtering data in the Visualize room [page 133]

4.1.2.1.1 Using the filter dialog in the Prepare room

You use the filter dialog in the *Prepare* room to define a filter on the data set and all of the visualizations based on it.

The filter dialog does not support SAP Lumira formula language or regular expressions (regex). You may need to add a calculated measure or dimension to create a more complex filter.

1. To open the filter dialog in the *Prepare* room, select the *Options*  icon in a column header or for a dimension, and select *Filter*.

   You can also select the token for an existing data set filter to edit it in the filter dialog.
2. On the filter dialog, choose an operator from the list.
3. Select or type the values to filter:
   - For filters that use the *Between* operator, type a beginning value and an end value.
   - For filters that use the *In List* or *Not In List* operator, select values from the list in the filter dialog.

   > **Note**
   >
   > You can hold SHIFT while clicking values to select a range of values.
   >
   > You can also select the *Options*  icon to change the filter dialog settings, including displaying the number of times that each record occurs in the data set, and sorting the data by value or by number of occurrences.
   >
   > When filtering an alphanumeric dimension, you can select the *Find*  icon to search for a member by name.
4. Select *Apply*.

The data is filtered and a token representing the filter is added above the *Data* pane.

You can edit the filter by selecting the token in the *Prepare* room or *Visualize* room, or remove it by selecting the *Delete*  icon.

Related Information

Creating a calculated measure or dimension [page 54]

4.1.2.2 Formatting numbers and dates

You can choose a display format for number and date columns.

For example, numbers can be displayed as percentages or in scientific notation, and negative numbers can be displayed in parentheses. You can also add a custom symbol or character string as a prefix or suffix to the displayed values.

1. Open the display format dialog from the *Measures and Dimensions* panel or from the Data pane in the *Prepare* room:
   - On the *Measures and Dimensions* panel, select the *Options*  icon beside a measure or dimension, and select *Display Formatting*.
   - In the *Data* pane in the *Prepare* room, select the *Options*  icon in the header of a number or date column, and select *Display Formatting*.
2. Select formatting options and select *OK*.

   If you're adding a character string prefix or suffix, the maximum number of characters is 256.

4.1.2.3 Converting data to another type

You can convert data from one type to another. For example, you can convert text to dates or numbers to text.

1. Perform one of the following actions:
   - On the *Measures and Dimensions* panel, select the *Options*  icon next to a dimension.
   - In the *Data* pane, select the  icon in a column heading.
2. In the data conversion dialog, select options as needed, and select *OK*.

4.1.3 Creating measures and hierarchies

You enrich data by adding measures and time and geography hierarchies. Measures allow easy manipulation of calculations, and hierarchies enable you to use a natural grouping of related columns.

When acquiring data, SAP Lumira detects hierarchies and potential measures. Detected measures are displayed on the *Measures and Dimensions* panel, and dimensions identified as potential hierarchies are flagged with a  icon. You can select  next to a dimension to manually create hierarchies.

Related Information

Creating a geography or time hierarchy [page 50]

Creating measures [page 53]

4.1.3.1 Creating a geography or time hierarchy

Time, geography, and custom hierarchies enrich a data set. Time and geography hierarchies are detected automatically when a data set is acquired, but you can create hierarchies at any time.

Time hierarchies can be created on number and date columns. Geography hierarchies can be created only on columns containing values that are compatible with geography data values in the NAVTEQ database used by the application. You use the contextual menus of column headers and dimensions to create hierarchies.

1. Select the *Options*  icon next to a dimension, and select *Create a geographic hierarchy* > *By Names* or *Create a time hierarchy*.
   Column or dimension enrichment options appear for the hierarchy.
2. Choose which columns to map to the hierarchy:
   - For time hierarchies, select the columns to map for the *Year*, *Quarter*, *Month*, and *Day* levels.
   - For geography hierarchies, select the columns to map for some or all of the *Country*, *Region*, *Sub-Region*, and *City* levels. These columns are checked for matches with the internal geography database used by the application.
     Select *Detected columns* to display columns detected as possible matches in drop-down lists for the level. If no columns are detected, the lists are empty. Select *All columns* to include all columns in drop-down lists for the level.

   For time hierarchies, the new columns selected as levels are automatically added to the data set, and the new time hierarchy appears on the *Hierarchies* semantic tab.

   For geography hierarchies, the *Geographical Data* dialog appears, showing columns that matched the internal database in green, columns that were an ambiguous match in orange, and columns that did not match in red.

   For inferred dimensions, columns are created for the hierarchies.
3. (Optional for geography hierarchies) In the *Geographical Data* dialog, for each proposed match, select the proposition row and select *Choose* to accept the location or *Not found* to remove the row, and select *Confirm*.
   The new geography columns selected as levels are automatically added to the data set, and the new geography hierarchy appears on the *Measures and Dimensions* panel.

You can modify the matched levels of a geography hierarchy at any time. (Select the *Options*  icon next to the hierarchy name, select *Edit reconciliation*, change the proposed matches for a level, and select *Confirm*.)

Related Information

Creating a geography hierarchy with latitude and longitude data [page 51]
Creating a custom hierarchy [page 52]

4.1.3.2 Creating a geography hierarchy with latitude and longitude data

When a data set contains latitude and longitude data, you can customize a geography hierarchy to use the data. The application creates a measure for each column during data acquisition.

Latitude and longitude data must be numeric. If data is not numeric, you must convert column values using a formula (for example, `ToNumber()`). If columns are not numeric, you must define the numeric converted dimensions as measures.

The application automatically calculates hierarchical levels above and below a selected geography dimension. You can accept the calculated levels in your hierarchy or replace them with levels that you define based on your latitude and longitude data.

Level property	Description
`Category`	Definition of the level, either automatically calculated based on the latitude/longitude data or user-defined (you select the column to base a level on)
`Column`	For a user-defined level, select the column to use for the level.
`Latitude`	For a user-defined level, select the latitude data.
`Longitude`	For a user-defined level, select the longitude data.
`Level type`	Name of the level in the hierarchy

> **Note**
>
> The application does not support creating geography hierarchies with latitude and longitude data from SAP HANA data sources.

1. Check whether measures have been created for latitude and longitude columns.

   If measures have been created, go to step 4.
2. If measures have not been created, convert the latitude and longitude columns to a numeric data type:

   a. Select the *Options*  icon in the header of the latitude dimension or column, and select *Create Calculated Dimension*.
   The *New Calculated Dimension* dialog appears, with the column name already in the formula (for example, `{column_1}`).

   b. Double-click the `ToNumber(<param>)` function to insert it in the formula.

   c. Move the column name that appeared in step b to the `ToNumber()` function.
   The entire formula should be `ToNumber({column_1})`.

   d. Enter a name for the calculated dimension, and select *OK*.

   e. Repeat steps a-d for the longitude dimension or column.
3. Define the new numeric latitude and longitude dimension as measures:

a. Select the *Options*  icon for the new numeric latitude dimension, and select *Create a measure*.

b. Select the *Options*  icon for the new numeric longitude dimension, and select *Create a measure*.

4. Select the *Options*  icon in the geography or dimension column heading to base the hierarchy on, and select *Create a Geographic hierarchy* > *By Latitude/Longitude*.
   The *Geographical Data* dialog appears. The *Level Name* pane lists the hierarchy levels calculated by the application. The original dimension is shown in red and the calculated hierarchy levels are shown in green. You can accept the proposed hierarchy based on the latitude/longitude data, or you can customize the levels of the hierarchy. Select a level to display its properties in the left pane.
5. To accept the proposed calculated levels, select *OK*, and go to step 7.
6. To choose the columns to base levels on, for each level you want to define in the hierarchy, select *User Defined* in the *Category* list, and select properties for the level in the other lists.
   Use the arrows to the left of a level to move it up or down in the hierarchy. Add a level to the hierarchy by selecting *Add Level*, or remove a level by selecting the *X* icon to the left of the level.
7. Select *OK*.

The geography hierarchy is added to the *Hierarchies* category on the semantic pane. You can change the levels of a hierarchy at any time. (Select the *Options*  icon next to the hierarchy name, select *Edit reconciliation*, select the column to base the level on, and select *OK*.)

Related Information

Creating a geography or time hierarchy [page 50]
Creating a calculated measure or dimension [page 54]

4.1.3.3 Creating a custom hierarchy

You can create a hierarchy using any combination of the available dimensions.

1. Select the *Options*  icon next to the dimension to use as the basis for the hierarchy.
2. Select *Create a custom hierarchy*.
   The *Create Hierarchy* dialog appears. The dimensions available on the *Measures and Dimensions* panel are listed in the left pane. You can enter a search string to find a dimension (for example, the first letters of a dimension name).
3. Add dimensions to the hierarchy in the right pane.

   > **Tip**
   >
   > You can double-click a dimension to move it between the panes.

4. (Optional) Use the arrows beside the hierarchy list to move a selected dimension up or down in the hierarchy.

5. Enter a name for the hierarchy, and select *Create*.
   The new custom hierarchy appears on the *Measures and Dimensions* panel. New columns are created for each level of the new hierarchy.

4.1.3.4 Creating measures

Measures enrich data sets. You can manually create them at any time directly from a column or dimension or by using the formula language to create a calculated measure, or you can allow the application to detect them automatically on numeric column data types when a data set is acquired.

> **Note**
>
> When using a *Connect to SAP HANA* data source, it is not possible to create a measure with a numeric or string dimension. Measures in *Connect to SAP HANA* data sources are detected directly from the SAP HANA Analytic view. Measures must be created in the SAP HANA view, before being acquired automatically in the application.

> **Note**
>
> When using a *Connect to SAP HANA* data source, it is not possible to change the aggregation type of a measure.

Related Information

Creating a measure from a column or dimension [page 53]
Creating a calculated measure or dimension [page 54]

4.1.3.4.1 Creating a measure from a column or dimension

You can create a measure from almost any column or dimension.

These exceptions apply:

- When the column data type is *Numeric*, any aggregate function can be used for the measure.
- When the column data type is *Date* or *String*, neither *Sum* nor *Average* can be used.
- Aggregation is performed when the measure is used in the *Facets* view. It is not available in the *Grid* view.

Function	Description
Sum	Returns the sum of a measure
Min	Returns the smallest value in a set of values

Function	Description
Max	Returns the largest value in a set of values
Count (Distinct)	Returns the number of distinct values in a set of values
Count (All)	Returns the number of values in a set of values
Average	Returns the average value of a measure
None	Allows a numeric dimension to be used as a measure, without aggregation. This type of measure enables each value to be visualized in a graph, which is useful for certain types of graphs. For example, for a scatter plot that displays margin and quantity-sold values, this option displays all points on the scatter plot and shows the spread of individual values that would not be apparent using an aggregation function. **Note** The aggregation type *None* is not supported when using a *Connect to SAP HANA* data source.

1. Select the  icon on a column heading or next to the dimension to use as the basis for the measure, and select *Create a measure*.
   A measure is created in the *Measures* section of the *Measures and Dimensions* panel.
2. Select the  icon next to the new measure, select *Change Aggregation*, and select an aggregate function.

Switch to the *Facets* view to see the measure applied to data in a data set. Select the measure to see changes to data values caused by aggregation.

4.1.4 Creating a calculated measure or dimension

You can create calculated measures and dimensions using the formula language.

The following features are supported in the formula editor:

- Combining any two columns in a dataset
- Applying functions from a predefined set of numeric, date, and text functions
- Using "if," "then," "else" clauses
- Using automatic completion to improve editing speed
- Using a calendar picker for date parameters
- Copying text and syntax to a function definition

1. Select the *Calculation* button, and select *New Calculated Dimension* or *New Calculated Measure*.

> **Tip**
>
> You can create a calculated measure or dimension with the *Options*  icon next to a measure or dimension, or by selecting *Create Calculated Dimension* on the *Manipulation Tools* panel.

2. Enter a name for the measure or dimension.
3. Double-click one or more measures or dimensions and functions to add them to the *Formula* syntax box.
4. Enter parameters for the function and associated information based on the function task.

   You must enter the names of columns used in the formula. Automatic completion will suggest a column name after you start entering the first letter.
5. If you are inputting calendar information, select the *Select a Date* button at the bottom of the functions list, and use the date picker to select dates.
6. Select *OK* to apply the formula.
   A measure or dimension is created.

> **Example**
>
> Suppose you want to create a dimension that multiplies the values in the `<margin_gross_percent>` column by 100 and rounds up to the next integer.
>
> 1. Select the *Calculation* button, and select *New Calculated Dimension*.
> 2. In the *New Calculated Dimension* dialog, double-click a dimension object or function to insert the dimension or function in the *Formula* box. For example, double-click `Ceil(num)`.
> 3. Edit the formula and add other dimension objects as needed. A new column with a default formula name is created. For example, add `Ceil(margin_gross_percent )*100` to the formula to create a column called `Ceil(margin_gross_percent )*100`.
> 4. Enter a name for the new calculated dimension column, and select *OK*. The new column is created in the *Data* pane and appears on the *Measures and Dimensions* panel.

4.1.4.1 Functions reference

You can define which functions will be available on the formula bar.

Category	Description
Character	Manipulates character strings
Aggregate Aggregate functions are implemented in the definition of a measure.	Aggregates data (for example, by summing or averaging a set of values)
Date and Time	Returns date or time data
Numeric	Returns numeric data

Category	Description
Logical	Returns `true` or `false`
Miscellaneous	Functions that do not fit in any other category

Related Information

Creating a measure from a column or dimension [page 53]
Character functions [page 57]
Date and time functions [page 62]
Numeric functions [page 65]
Logical functions [page 68]
Miscellaneous functions [page 70]

4.1.4.1.1 Character functions

Use a character function to manipulate character strings in a formula. The input is a column of your data sets, and functions are applied to cell contents.

Function	Syntax	Description
`<matchExpr> like <pattern>`	• `matchExpr`: The string expression to search • `pattern`: The pattern string constant to search for	Determines whether a character string matches a specified pattern. The search is not case-sensitive. The pattern can include regular characters and the following special characters: • `"_"` matches a single character • `"%"` matches zero to many characters Before you can use a special character as a regular character, you must escape it, using a backslash ("\"). **Note** `"["`, `"^"`, `"-"`, and `"]"` are reserved for future use. For example: `"Hiking is fun" like "H % is _un"` returns `true`
`Concatenate(str1, str2)`	• `str1`: First string • `str2`: Second string	Concatenates two strings into a single string. The operator + can also concatenate strings. For example: `Concatenate("Mr", "Brown")` returns `"MrBrown"`

Function	Syntax	Description
`Contain(whereStr, whatStr)`	• `whereStr`: String in which a search is conducted • `whatStr`: Substring that is the object of the search	Returns occurrences of a string within another string. The search is not case-sensitive. For example: `Contain("Cats are grey", "aRe")` returns `true`
`ExceptFirstWord(str, sep)`	• `str`: Input string • `sep`: A separator	Returns a copy of a string, with the first word removed. For example: `ExceptFirstWord("Level 3, Standford Street", ", ")` returns `"Standford Street"`
`ExceptLastWord(str, sep)`	• `str`: Input string • `sep`: A separator	Returns a copy of a string, with the last word removed. For example: `ExceptLastWord("james.brown@company.com", "@")` returns `"james.brown"`
`FirstWord(str, sep)`	• `str`: Input string • `sep`: A separator	Returns the first word of a string. For example: `FirstWord("Senior Developer", " ")` returns `"Senior"`
`LastWord(str, sep)`	• `str`: Input string • `sep`: A separator	Returns the last word of a string. For example: `LastWord("Red/Purple", "/")` returns `"Purple"`

Function	Syntax	Description
`Length(str)`	`str`: Input string	Returns the length of a string. For example: `Length("How long")` returns `8`
`LowerCase(str)`	`str`: Input string	Returns a copy of a string, with all characters converted to lowercase. For example: `LowerCase("GOOD JOB")` returns `"good job"`
`Lpad(str, length, pad)`	• `str`: Input string • `length`: Expected length • `pad`: Character sequence to add	For example:Returns a copy of a string, padded with leading characters to the specified total length. `Lpad("Incomplete field", 20, "#")` returns `"####Incomplete field"`
`Replace(str, target, replacement)`	• `str`: Input string • `target`: String to be replaced • `replacement`Returns a copy of a string, padded with leading characters to the specified total: String value to insert	Returns a string, with all occurrences of a specified string replaced with another specified string. For example: `Replace("hyperthermia", "ert", "ot")` returns `"hypothermia"`
`Rpad(str, length, pad)`	• `str`: Input string • `length`: Expected length • `pad`: Character sequence to add	Returns a copy of a string, padded with trailing characters to the specified total length. For example: `Rpad("Incomplete field", 20, "#")` returns `"Incomplete field####"`

Function	Syntax	Description
`SubString(str, start)`	• `str`: String from which a substring is computed • `start`: Start position in the input substring	Returns a substring of a string. For example: `SubString("Wong", 3)` returns `"ng"`
`SubString(str, start, length)`	• `str`: String from which a substring is computed • `start`: Start position in the input substring • `length`: Length of the substring to return	Returns a substring of a string. For example: `SubString("Wong", 2, 2)` returns `"on"`
`ToText(param)`	`param`: Parameter to convert	Converts a parameter to a string. All parameters are valid, and numbers are truncated to zero decimal places.
`Trim(str, toTrim)`	• `str`: Input string • `toTrim`: Character to be removed	Returns a copy of the string, with the leading and trailing repetitions of a character removed. This function is case-sensitive. For example: `Trim("Aurora", "a")` returns `"Auror"`
`TrimLeft(str, toTrim)`	• `str`: Input string • `toTrim`: Character to remove	Returns a copy of the string, with the leading occurrence of a character removed. This function is case-sensitive. For example: `TrimLeft("Above", "A")` returns `"bove"`
`TrimRight(str, toTrim)`	• `str`: Input string • `toTrim`: Character to be removed	Returns a copy of a string, with trailing repetitions of a character removed. This function is case-sensitive. For example: `TrimRight("Laura", "a")` returns `"Laur"`

Function	Syntax	Description
`UpperCase(str)`	`str`: Input string	Returns a copy of a string, with all characters converted to uppercase. For example: `UpperCase("Little Boy")` returns `"LITTLE BOY"`
`ToDate(string, format)`	The date format is a combination of the following reserved tokens, separated by delimiters: • `d` or `dd`: Day of month (1-31) • `M` or `MM`: Month of year (1-12) • `y` or `yy`: Abbreviated year without century (00-99) `yyyy`: Year with century (1956, 2012, 2014, and so on) All other sequences are considered delimiters. • `string`: Input string to convert • `format`: Date format string constant	Converts an input string in a data sets to a date in a specified format, when the dates in a column of an original data source are in string format. For example: `ToDate(Obj, 'yyyy/dd/MM')` converts a string in the format `yyyy/dd/MM` to a date

Name	Trimmed string
Aurora	Auror
Auror	Auror
auror	uror
aurora	uror
uror	uror

This formula returns `"Auror"`: `Trim("Aurora", "a")`.

4.1.4.1.2 Date and time functions

Date and time functions return date or time data. Note that you may need to convert the format of your source data in the application.

Function	Syntax	Description
`AddMonthToDate(#date#,periods)`	• `#date#`: Original date • `periods`: Number of periods to add	Returns a date that is produced by adding a specified number of month(s) to a specified date. For example: `AddMonthToDate(#2012-01-01#,1)` returns `2012-02-01`
`AddWeekToDate(#date#,periods)`	• `#date#`: Original date • `periods`: Number of periods to add	Returns a date that is produced by adding a specified number of week(s) to a specified date. For example: `AddWeekToDate(#2012-01-01#,1)` returns `2012-01-08`
`AddYearToDate(#date#,periods)`	• `#date#`: Original date • `periods`: Number of periods to add	Returns a date that is produced by adding a specified number of year(s) to a specified date. Use negative numbers to remove a year. For example: `AddYearToDate(#2012-01-01#,1)` returns `2013-01-01`
`CurrentDate()`		Returns the current date as a date. For example: `CurrentDate()` returns `<CurrentDate>`

Function	Syntax	Description
`DateDiffInDays(#start#,#end#)`	• `#start#`: Start date of the interval • `#end#`: End date of the interval	Returns the number of days between two dates. For example: `DateDiffInDays(#2012-03-23#,#2012-01-30#)` returns `-53`
`DateDiffInMonths(#start#,#end#)`	• `#start#`: Start date of the interval • `#end#`: End date of the interval	Returns the number of months between two specified dates. For example: `DateDiffInMonths(#2013-02-01#,#2014-01-01#)` returns `11`
`Day(#date#)`	`#date#`: A date	Returns the day of the month as a number from 1 to 31. For example: `Day(#2012-03-23#)` returns `23`
`DayOfWeek(#date#)`	`#date#`: A date	Returns the day of the week as a number from 1 (Sunday) to 7 (Saturday). For example: `DayOfWeek(#2012-03-23#)` returns `6`
`DayOfYear(#date#)`	`#date#`: A date	Returns the day of the year as a number. For example: `DayOfYear(#2012-03-23#)` returns `83`

Function	Syntax	Description
`LastDayOfMonth(#date#)`	`#date#`: A date	Returns the date produced by computing the last day of the month of a specified date. For example: `LastDayOfMonth(#2012-03-23#)` returns the date `2012-03-31`
`LastDayOfWeek(#date#)`	`#date#`: A date	Returns the date produced by computing the last day of the week of a specified date. For example: `LastDayOfWeek(#2012-03-23#)` returns the date `2012-03-24`
`MakeDate(year,month,day)`	• `year`: Number that represents a year • `month`: Number that represents a month • `day`: Number that represents a day of the month	Returns a date that is built from a specified year, month, and day. For example: `MakeDate(2011,6,12)` returns the date `2011-06-12`
`Month(#date#)`	`#date#`: A date	Returns the month of the year as a number from 1 to 12. For example: `Month(#2012-03-23#)` returns `3`
`Quarter(#date#)`	`#date#`: A date	Returns a number that represents the quarter of a specified date. For example: `Quarter(#2012-03-23#)` returns `1`

Function	Syntax	Description
`Week(#date#)`	`#date#`: A date	Returns a number that represents the week of a specified date. For example: `Week(#2012-03-23#)` returns `12`
`Year(#date#)`	`#date#`: A date	Returns the year of a specified date. For example: `Year(#2012-03-23#)` returns `2012`

4.1.4.1.3 Numeric functions

Use numeric functions to return numeric values in a formula.

Function	Syntax	Description
`Ceil(num)`	`num`: A number	Returns the smallest integer that is greater than or equal to a specified number. For example: `Ceil(14.2)` returns `15`
`Floor(num)`	`num`: A number	Returns the largest integer that is not greater than a specified number. For example: `Floor(14.8)` returns `14`

Function	Syntax	Description
`Log(num)`	`num`: A number	Returns the natural logarithm of a specified number. For example: `Log(100)` returns `4.605`
`Log10(num)`	`num`: A number	Returns the base 10 logarithm of a specified number. For example: `Log10(100)` returns `2`
`Mod(num, divisor)`	• `num`: A number • `divisor`: The divisor	Returns the remainder of the division of a number by another number. For example: `Mod(15,2)` returns `1`
`Power(num, exponent)`	• `num`: A number • `exponent`: The exponent	Raises a number to a power. The operator ^ (caret) can be used instead of this function. For example: `Power(2,3)` returns `8`
`Round(num, digits)`	• `num`: A number • `digits`: The number of decimal places to round off to	Returns a numeric value, rounded to a specified number of decimal places. For example: `Round(14.81, 1)` returns `14.8`

Function	Syntax	Description
`Sign(num)`	`num`: A number	Returns -1 if a specified number is negative, 0 if the specified number is zero, or +1 if the specified number is positive. For example: `Sign(-2)` returns `-1`
`ToText(num, digits)`	• `num`: A number • `digits`: Number of decimal places to use. This parameter is optional, and its default value is 0.	Converts a specified number to a string. The number is truncated to the specified number of decimal places. For example: `ToText(12.1451, 2)` returns `12.14`
`Truncate(num, digits)`	• `num`: A number • `digits`: Number of decimal places to truncate	Returns a numeric value, truncated at a specified number of decimal places. For example: `Truncate(12.281, 1)` returns `12.200`

Temperature	Text
-2.01	-2.0
-1.06	-1.1
0.08	0.1
1.07	1.1
2.08	2.1
3.99	4.0
5.00	5.0

This formula returns `12.14`: `ToText(12.1451, 2)`.

Temperature	Truncated
-2.01	-2.00
-1.06	-1.00
0.08	0.00
1.07	1.00

Temperature	Truncated
2.08	2.00
3.99	3.90
5.00	5.00

This formula returns `12.200`: `Truncate(12.281, 1)`.

4.1.4.1.4 Logical functions

You can use logical functions in a formula to return `true` or `false`.

Function	Syntax	Description
`IsNotNull(obj)`	`obj`: User object (column)	Returns a Boolean value that indicates whether a supplied field does not contain a null value. When a field contains a null value, the function returns `false`. For all other values, the function returns `true`.
`IsNull(obj)`	`obj`: User object (column)	Returns a Boolean value that indicates whether the supplied field contains a null value. When a field contains a null value, the function returns `true`. For all other values, the function returns `false`.
`<left> and <right>`	• `left`: Left operand • `right`: Right operand	Returns the logical conjunction of its Boolean inputs. This function returns `false`: `true and false`.
`<left> or <right>`	• `left`: Left operand • `right`: Right operand	Returns the logical disjunction of its Boolean inputs. This function returns `true`: `true or false`.
`if<cond> then <alt1> else <alt2>`	• `cond`: Boolean condition to test • `alt1`: Alternative 1 • `alt2`: Alternative 2	Chooses between two alternatives, based on a Boolean condition. The second alternative is optional and evaluates to `null` when missing.

Function	Syntax	Description
`<testExpr> in <candidateList>`	• `testExpr`: Expression to be tested • `candidateList`: List of match candidates	Use to determine whether a first input matches a value in a second input list. For example: `3 in [2, 4, 6]` returns `false`
`not<bool>`	`bool`: A Boolean	Use to negate a Boolean input. For example: `not false` returns `true`

Left	Right	Result of {Left} and {Right}
True	True	`true`
True	False	`false`
False	True	`false`
False	False	`false`

This function returns `false`: `true and false`.

Left	Right	Result of {Left} or {Right}
True	True	true
True	False	true
False	True	true
False	False	false

This function returns `true`: `true or false`.

4.1.4.1.5 Miscellaneous functions

These functions can be used in a formula, but they do not fit into a standard category for function families.

Function	Syntax	Description
`GroupValues(column, ListOfValues, newValue)`	• `column`: User object to apply the grouping to • `ListOfValues`: List of values to be grouped • `newValue`: Value that will replace the grouped values	Groups a list of values. For example: `GroupValues(CountryColumn, ["USA", "India", "France"], "My Countries")` returns `"My Countries"` when the `CountryColumn` column contains `"USA"`, `"India"`, or `"France"`
`ToNumber(param)`	`param`: Parameter to convert	Converts any type of parameter to a numeric value. Numbers are truncated to zero decimal places.

4.1.5 Working with multiple data sets

You can add a data set to the available datasets, move between datasets, and merge or append two datasets.

When combining datasets, two datasets are merged using a JOIN operator, and two matched datasets are merged using a UNION operator. Appended datasets are compatible and have an equivalent number of columns in the merged table.

Related Information

Adding a data set [page 71]
Switching to another data set [page 71]
Merging data sets (JOIN) [page 71]
Appending data sets (UNION) [page 72]
Removing a data set [page 73]

4.1.5.1 Adding a data set

You can open multiple data sets in the same document, and you can add a data set to a document.

1. In the datasets list at the top of the *Data* pane, select *Add dataset*.
2. In the *Add new dataset* dialog, select a data source in the *Select a Source* or *All Recently Used* pane, and select *Next*.
3. Enter connection information for the data set, and select *Create*.

Data from the data set is acquired in the document.

4.1.5.2 Switching to another data set

You can have multiple data sets open in a document at the same time and switch from one data set to another, which is useful when preparing a merge between two datasets.

In the data sets list at the top of the *Data* pane, select the data set to switch to.

The data set you selected is now the active data set.

4.1.5.3 Merging data sets (JOIN)

Use the JOIN operator to merge two data sets.

- The merging data set must have a key column.
- Only columns with the same data type can be merged.
- The merge process combines all columns.

Columns in the second data set are matched to a key column in the original data set. The application proposes potential column matches and the probability of each match.

> **Note**
>
> Once a data set has been merged with another data set, the datasets are a unit. You cannot remove either data set.

1. Select the *Combine*  icon, and select *Merge*.
2. In the *Merge Data* dialog, select the key column to use as the identifying column for matching.
3. Perform one of the following actions:

Option	Description
If the data set to merge is already available in the document	Select the data set in the list above the right pane.

Option	Description
If the data set to merge is not open	Select *Add New Dataset*, and select the data source to merge.

Columns that can be matched, based on the key column in the original data set, are listed under *Lookup Dataset*.

4. Select *Merge*.

Columns in the second data set are added to the original data set.

Related Information

Appending data sets (UNION) [page 72]

4.1.5.4 Appending data sets (UNION)

Use the UNION operator to append two data sets.

Both tables in the union must contain an equivalent number of columns and compatible data types. Only a data set that is compatible with the target data set can be appended.

Once a data set has been merged with another data set, the data sets become a unit, and you cannot separate them.

1. Select the *Combine*  icon, and select *Append*.
2. In the *Append Data* dialog, perform one of the following actions:

Option	Description
If the data set to append is already available in the document	Select the data set in the list above the right pane.
If the data set to append is not open	Select *Add New dataset*, and select the data source to acquire and append.

If the data set to append is compatible with the original data set, dimension columns are listed under *Lookup dataset* on the right side of the pane. A sample of distinct values for each selected dimension appears in the *Sample of Distinct Values* column.

3. To select a different source dimension for the union with the matching target dimension, select another dimension in the list.

   If the selected dimension contains a compatible data type, the dimension can be appended. If a `The union is not possible` message appears in red, the selected dimension didn't contain a compatible data type and you must select a compatible dimension.
4. Select *Append*.
   The two data sets are combined. The combined data set retains the column names of the target data set.

4.1.5.5 Removing a data set

You can remove a data set from a document, if it has not been appended or merged with another data set.

1. In the datasets list at the top of the *Data* pane, select the data set to remove.
2. Select the *Remove Dataset*  icon next to the data set.

The data set and any visualizations based on it are deleted.

5 Building Analyses

5.1 Creating an Analysis

You can use Expert Analytics to perform data mining and statistical analysis by running data through a series of components. The series of components are connected to each other with connectors, which define the direction of the data flow. This process is referred to as analysis.

A document is your starting point when using Expert Analytics. You create a new document to start analyzing your data and building new analysis. You can open locally stored saved documents to view or modify existing analysis and data sets.

Each document is a file that contains:

- Connection parameters for the data source if the source is an RDBMS.
- Data set: The column data used to create charts.
- Analyses and models, and their results.
- Charts built on the data and saved as visuals.

To create an analysis, perform the following steps:

1. Acquire data from a data source.
2. Optional: Prepare the data for analysis (for example, by filtering the data).
3. Apply algorithms.
4. Optional: Store the results of the analysis for further analysis.

To add multiple analyses to the document, choose  (Add Analysis) in the analysis toolbar.

> **Note**
>
> To get the latest version of SAP HANA and associated libraries, go to the SAP Product Availability Matrix.

5.1.1 Applying Preprocessing Components to Data

You can apply preprocessing components to your data, such as filters, before performing an analysis.

This is an optional step.

In many cases, the raw data from the data source may not be suitable for analysis. For accurate results, you may need to prepare and process the data before analysis. You can find data manipulation functions in the *Prepare* room and data preparation functions in the *Predict* room. In the *Prepare* room, you can work on the static data or raw data that is imported into Expert Analytics. In the *Predict* room, you can work on the transient data using preprocessor components.

Data preparation involves checking data for accuracy and missing fields, filtering data based on range values, sampling the data to investigate a subset of data, and manipulating data. You can process data using data preparation components.

1. In the *Predict* room, double-click the required preprocessor component under the list of components on the right.
   The preprocessor component is added to the analysis editor and an automatic connection is created to the data source component.
2. From the contextual menu of the preprocessor component and choose *Configure Properties*.
3. In the component properties dialog box, enter the necessary details for the preprocessor component properties.
4. Choose *Done*.
5. To view the results of the analysis, choose  (Run Analysis).

5.1.2 HANA Demand Forecasting Component Overview

The HANA Demand Forecasting component runs an algorithm on HANA to produce sales predictions for a set period in the future. The component functionality is a subset of Unified Demand Forecast (UDF), a module in SAP Customer Activity Repository (CAR). A primary focus of the component is to forecast Consumer Demand. As well as providing forecast and forecast interval information, the algorithm also provides data on price elasticity for all products in the workflow. Described below are the configurable properties, results grid, and algorithm summary of the HANA Demand Forecasting component.

> **Note**
>
> For a guide to configuring the component properties, see Configuring the HANA Demand Forecasting Component [page 83].

Component Properties Settings

Section Name	Property Descriptions
Forecast Horizon	Set the prediction period *Start* and *End* dates.

Section Name	Property Descriptions
Variables	Set the following Variable properties: *Product ID*: Select the string-only column from the input table that contains the product identifier code, which can be up to 60 characters long. *Location ID*: Select the string-only column from the input table that contains the location identifier code, which can be up to 60 characters long. *Transaction Timestamp*: Select the column from the input table that contains the transaction timestamp, which must be in date or timestamp format. *Unit Sales*: Select the numeric-only column from the input table that contains the unit sales figure. *Revenue*: Select the numeric-only column from the input table that contains the revenue figure.
Holidays (Optional)	Contains information about public holidays in particular locations. Set the following Holidays properties: *Schema*: Select the schema for the input table from the list in the HANA database. *Tables*: Select a table from the schema. *Views*: Select a view from the schema. *Time Stream ID*: Select the string-only column from the input table that contains the time stream identifier code, which can be up to 10 characters long. *Public Holiday Key*: Select the string-only column from the input table that contains the public holiday key, which can be up to 3 characters long. *Operational Status*: Select the integer-only column from the input table that contains the operational status. *Timestamp*: Select the column from the input table that contains the transaction timestamp, which must be in date or timestamp format.

Section Name	Property Descriptions
Locations to Holiday Mapping (Optional)	Set the following Locations to Holiday Mapping properties: *Schema*: Select the schema that contains the table with information about mapping locations to public holidays. *Tables*: Select a table from the schema. *Views*: Select a view from the schema. *Location ID*: Select the string-only column from the input table that contains the location identifier code, which can be up to 60 characters long. *Holiday ID*: Select the string-only column from the input table that contains the holiday identifier code, which can be up to 10 characters long.
Demand Influencing Factors (Optional)	Set the following Demand Influencing Factors properties: *Schema*: Select the schema that contains the table with information about Demand Influencing Factors. *Tables*: Select a table from the schema. *Views*: Select a view from the schema. *Product ID*: Select the string-only column from the input table that contains the product identifier code. *Location ID*: Select the string-only column from the input table that contains the location identifier code. *DIF Attribute*: Select the string-only column from the input table that contains the Demand Influencing Factor (DIF) attribute, which can be up to 32 characters long. *Timestamp From*: Select the date-only column from the input table that contains the date that the timestamp begins. *Timestamp To*: Select the date-only column from the input table that contains the date that that the timestamp ends. *DIF Value*: Select the numeric-only column from the input table that contains the Demand Influencing Factor (DIF) value.

Section Name	Property Descriptions
Expected Future Prices (Optional)	Set the following Expected Future Prices properties: *Schema*: Select the schema that contains the table with information about Expected Future Prices. *Tables*: Select a table from the schema. *Views*: Select a view from the schema. *Product ID*: Select the string-only column from the input table that contains the product identifier code. *Location ID*: Select the string-only column from the input table that contains the location identifier code. *Timestamp From*: Select the date-only column from the input table that contains the date that the timestamp begins. *Price*: Select the numeric-only column from the input table that contains the price.

Component Advanced Settings (Optional)

Section Name	Property Descriptions
Configuration Parameters (Optional)	In the Configuration Parameters section, you have the option to configure the following settings: *Damping factor* (FC_TREND_DAMP): Defines the damping factor for the trend regressor. The range is value >= 0.00000. *Total regress mass* (MOD_HDM_NEAR_HOLIDAY_DENSITY): Sets the proportion of total regressor mass to the right (POST: left) of the half-way date. Note that there are two groups of HDM-regressors, SYS:CAL:YR:HDM:PRE:* and SYS:CAL:YR:HDM:POST:*. The PRE-regressors define the ramp-up before the holiday, the POST-regressors the ramp-down after the holiday. The range is 0.50000 <=value < 1.00000.

Section Name	Property Descriptions
Time Delay Effect (Optional)	In the Time Delay Effect section, configure the following settings: *Observation weight* (MOD_TIME_WEIGHT): Sets the weight of a one-year-old observation in modeling, compared to an observation taken today. This way, the variable helps to decide whether to give equal importance (or weight) to all the records irrespective of their timestamps. For example, when building the model, setting the parameter value to 1 gives equal importance to all the records irrespective of the time that they were recorded. Whereas, setting a value to the parameter less than 1 enables the user to give less importance to the records that have older timestamps, as compared to the records that have recent timestamps. The range is 0.50000 <=value < 1.00000. *Lower Boundary on weight* (MOD_TIME_WEIGHT_MIN): Defines a lower boundary below which the weight will not fall. The range is 0.00001 <=value <= 1.00000.
Out of Stock Detection (Optional)	In the Out of Stock Detection section, configure the following settings: *Zero sales period*(MOD_OOSD_MIN_LEN): Sets the minimum length of the continuous zero sales to be considered for an out-of-stock period evaluation. The range is value >= 1.00000. *Probability threshold* (MOD_OOSD_THRSHLD): Sets the threshold for the probability score to determine whether an item is out-of-stock. The probability score for each item is derived based on the occurrence of zero sales for a period greater than the values specified in the parameter, MOD_OOSD_MIN_LEN. The range is value >= 1.00000.
Time Series Decomposition (Optional)	Select the appropriate Time Series Decomposition checkboxes to decompose and clearly see the influence on your results of the following factors: *Seasonality*: Seasonality can impact the results at different times of the year, such as the start or end of the year, or every alternate month. *User Promotion*: A user promotion can impact the results with sudden spike in sales. *Holidays*: National holidays such as Christmas Day and Thanksgiving can impact sales. De-select the checkboxes if you do not want to consider the impact of these factors in your results.

Section Name	Property Descriptions
Outlier Detection (Optional)	In the Outlier Detection section, configure the following settings: *Configure outlier detection*: Checkbox to switch on or off the outlier detection. *Acceptable distance from mean* (MOD_OUTLIER_MEAN_FACTOR): Defines the outlier detection factor to determine how far away from the mean is acceptable. *Min. non-zero observations* (MOD_OUTLIER_STD_DEV_FACTOR): Defines the outlier detection minimum number of non-zero observations, regular and promotional, counted before 0-filling. *Standard deviation factor*(MOD_OUTLIER_STD_DEV_FACTOR): Defines the outlier detection factor to determine how many deviations away from the mean is acceptable.

Component General Settings (Optional)

Section Name	Property Descriptions
Basic (Optional)	*Component Name*: Not configurable. In the Outlier Detection section, you can configure the following settings: *Alias Name*: An alias for the component name. *Description*: The purpose of the component.

Component Results Grid

Column	Description
PROD_ID	Product ID
LOC_ID	Location ID
TSTMP_FR	Timestamp From
TSTMP_TO	Timestamp To
ACTUAL_UNIT_SALES	Actual Unit Sales

Column	Description
FC_CONF_INDEX	Forecast Confidence Index (FCI)
FC_UNIT_SALES	Forecasted Unit Sales
INTERCEPT	Intercept of the time series decomposition component.
TREND	Trend of the time series decomposition component.
SEASONALITY	Seasonality of the time series decomposition component.
DAY_OF_WEEK	Day-of-week of the time series decomposition component.
HOLIDAY	Holiday of the time series decomposition component.
SALES_PROMOTION	Sales Promotion of the time series decomposition component.
PRICE	Product-location specific future price on a daily basis. Historical price calculated based on sales and unit price.
PRICE_ELASTICITY	Price Elasticity. Measures the responsiveness of the quantity demanded of a good or service to a change in its price.
FORECAST_INFO_MSG	Provides additional information that explains the Forecast Confidence Index (FCI).
FORECAST_INFO_DIF_DESC	Provides additional information that explains the demand influencing factor for that impacts the forecast.

Algorithm Summary

Category	Number of Product Location Combinations	Criteria	Descriptions
Perfectly Inelastic Demand	0	E = 0	The category is an extreme case because the quantity demanded is unaffected by any price change. The quantity is fixed and modifications to the price have no effect on the result.

Category	Number of Product Location Combinations	Criteria	Descriptions
Inelastic	0	-1 < E < 0	UDF constrains price elasticity to be less than zero and above -10 in the default parameterization. Therefore, a price elasticity above -1 is called an inelastic demand. This means that changes in price have a relatively small effect on the quantity of the good or service demanded. In contrast, if price elasticity is below -1, changes in price have a relatively large effect on the quantity of the good or service demanded, which is an elastic demand.
Unit Elastic	0	E = -1	Any change in price induces an equal relative change in quantity. For example, a 20% change in price includes a 20% change in quantity demanded. Unit Elastic is the dividing line between the elastic and inelastic ranges.
Relatively Elastic Demand	1	E < -1	The quantity demanded is extremely responsive to price because relatively small changes in price cause relatively large changes in quantity. For example, a 2% change in price leads to more than a 20% change in quantity demanded (perhaps more than 40%).

HANA Demand Forecasting (UDF)

The HANA Demand Forecasting component is a subset of Unified Demand Forecast (UDF) for SAP Retail applications on SAP HANA. It is part of the SAP Customer Activity Repository (CAR). UDF uses all necessary near-real time input data automatically out of this platform. Thus UDF relies on the Demand Data Foundation component of CAR. This component can provide for tasks such as NW-based job scheduling, batch job parallelization framework, exception workbench and configuration IMG-screen.

The unified forecasting engine represents a combination of the scientific forecasting expertise and methodologies from several sources. These include the SAP acquisitions SAF AG (SAP Forecasting and Replenishment) and Khimetrics (Demand Management Foundation).

UDF empowers business analysts with the knowledge of the impact that each Demand Influencing Factor (DIF) had on consumer demand in the past. For example, DIFs can be price changes, promotions, seasonality or a trend. The decomposed values can be used to forecast future demand to support consuming applications in Retail and Consumer Products. What's more, UDF learns from new demand data. This means that it automatically adapts the demand model to strengthen the forecast as more data is introduced.

The capabilities of the in-memory database technology are leveraged as much as possible. Therefore you can take full advantage of the opportunities provided by Big Data. As a result, you can model and forecast large amounts of data to enable new business scenarios. This enables you to support a high volume of data, to perform near real-time processing and to push detailed granular insight into demand data.

> **Note**
>
> For further information, see the SAP Help page, Unified Demand Forecast (UDF).

Related Information

Configuring the HANA Demand Forecasting Component [page 83]

5.1.2.1 Configuring the HANA Demand Forecasting Component

You can configure the properties of the HANA Demand Forecasting component.

Prerequisites:

SAP HANA system and the corresponding version of the Unified Demand Forecast Application Function Library (UDF AFL). To get the latest version of SAP HANA and associated libraries, go to the SAP Product Availability Matrix.

Process:

When configuring the HANA Demand Forecasting component, it is mandatory to map information about schema, table and column names from your HANA sales table. You perform the mapping in the Variables section of the component's Properties tabbed page.

You have the option to configure the settings in the remaining sections of the Properties tabbed page, and the configurations settings in the Advanced and General tabbed pages.

To configure the component, take the following steps:

1. In Expert Analytics, connect to a *HANA Data Source*. This data source is your sales table.
2. Navigate to the *Predict Room*.
3. In the *Predict Room*, from the Component List select *Time Series* - *HANA Demand Forecasting*. Drag-and-drop the *HANA Demand Forecasting* component to the analysis editor. Alternatively, double-click the *HANA Demand Forecasting* component. Click *OK*.

4. To open the configuration settings, double-click the *HANA Demand Forecasting* component. Alternatively, on the component click the *Settings* icon  and, from the context menu, select *Configure Settings*.
5. In the Properties panel of the HANA Demand Forecasting dialog box, the Forecast Horizon section enables you to set the prediction period. Set the *Start* and *End* dates.
6. In the Variables section, you map information from your sales table to the component. Configure the following settings:
    a. *Product ID*: Select the string-only column from the input table that contains the product identifier code, which can be up to 60 characters long.
    b. *Location ID*: Select the string-only column from the input table that contains the location identifier code, which can be up to 60 characters long.
    c. *Transaction Timestamp*. Select the column from the input table that contains the transaction timestamp, which must be in date or timestamp format.
    d. *Unit Sales*. Select the numeric-only column from the input table that contains the unit sales figure.
    e. *Revenue*: Select the numeric-only column from the input table that contains the revenue figure.
7. Optionally, in the Holidays section, configure the following settings:
    a. *Schema*: Select the schema for the input table.
    b. *Tables*: Select a table from the schema.
    c. *Views*: Select a view from the schema.
    d. *Time Stream ID*: Select the string-only column from the input table that contains the time stream identifier code, which can be up to 10 characters long.
    e. *Public Holiday Key*: Select the string-only column from the input table that contains the public holiday key, which can be up to 3 characters long.
    f. *Operational Status*: Select the integer-only column from the input table that contains the operational status.
    g. *Timestamp*: Select the column from the input table that contains the transaction timestamp, which must be in date or timestamp format.
8. Optionally, in the Locations to Holiday Mapping section, configure the following settings:
    a. *Schema*: Select the schema that contains the table with information about mapping locations to public holidays.
    b. *Tables*: Select a table from the schema.
    c. *Views*: Select a view from the schema.
    d. *Location ID*: Select the string-only column from the input table that contains the location identifier code, which can be up to 60 characters long.
    e. *Holiday ID*: Select the string-only column from the input table that contains the holiday identifier code, which can be up to 10 characters long.
9. Optionally, in the Demand Influencing Factors section, configure the following settings:
    a. *Schema*: Select the schema that contains the table with information about Demand Influencing Factor..
    b. *Tables*: Select a table from the schema.
    c. *Views*: Select a view from the schema.
    d. *Product ID*: Select the string-only column from the input table that contains the product identifier code.
    e. *Location ID*: Select the string-only column from the input table that contains the location identifier code.

f. *DIF Tag*: Select the string-only column from the input table that contains the demand influencing factor tag, which can be up to 32 characters long.
g. *DIF Attribute*: Select the string-only column from the input table that contains the demand influencing factor attribute, which can be up to 32 characters long.
h. *Timestamp From*: Select the date-only column from the input table that contains the date that the timestamp begins.
i. *Timestamp To*: Select the date-only column from the input table that contains the date that the timestamp ends.
j. *DIF Value*: Select the numeric-only column from the input table that contains the Demand Influencing Factor (DIF) value.

10. Optionally, in the Expected Future Prices section, configure the following settings:
    a. *Schema*: Select the schema that contains the table with information about Expected Future Prices.
    b. *Tables*: Select a table from the schema.
    c. *Views*: Select a view from the schema.
    d. *Product ID*: Select the string-only column from the input table that contains the product identifier code.
    e. *Location ID*: Select the string-only column from the input table that contains the location identifier code.
    f. *Timestamp From*: Select the date-only column from the input table that contains the date that the timestamp begins.
    g. *Price*: Select the numeric-only column from the input table that contains the price.
11. Optionally, navigate to the Advanced tabbed page. In the Configuration Parameters section, configure the following settings:
    a. *Damping factor* (FC_TREND_DAMP): Defines the damping factor for the trend regressor. The range is value >= 0.00000.
    b. *Total regress mass* (MOD_HDM_NEAR_HOLIDAY_DENSITY): Sets the proportion of total regressor mass to the right (POST: left) of the half-way date. Note that there are two groups of HDM-regressors, SYS:CAL:YR:HDM:PRE:* and SYS:CAL:YR:HDM:POST:*. The PRE-regressors define the ramp-up before the holiday, the POST-regressors the ramp-down after the holiday. The range is 0.50000 <=value < 1.00000.
12. Optionally, in the Time Delay Effect section, configure the following settings:
    a. *Observation weight* (MOD_TIME_WEIGHT): Sets the weight of a one-year-old observation in modeling, compared to an observation taken today. This way, the variable helps to decide whether to give equal importance (or weight) to all the records irrespective of their timestamps. For example, when building the model, setting the parameter value to 1 gives equal importance to all the records irrespective of the time that they were recorded. Whereas, setting a value to the parameter less than 1 enables the user to give less importance to the records that have older timestamps, as compared to the records that have recent timestamps. The range is 0.50000 <=value < 1.00000.
    b. *Lower Boundary on weight* (MOD_TIME_WEIGHT_MIN): Defines a lower boundary below which the weight will not fall. The range is 0.00001 <=value <= 1.00000.
13. Optionally, in the Out of Stock Detection section, configure the following settings:
    a. *Zero sales period*(MOD_OOSD_MIN_LEN): Sets the minimum length of the continuous zero sales to be considered for an out-of-stock period evaluation. The range is value >= 1.00000.
    b. *Probability threshold* (MOD_OOSD_THRSHLD): Sets the threshold for the probability score to determine whether an item is out-of-stock. The probability score for each item is derived based on the occurrence of zero sales for a period greater than the values specified in the parameter, MOD_OOSD_MIN_LEN. The range is value >= 1.00000.

14. Optionally, select the appropriate Time Series Decomposition checkboxes to decompose and clearly see the influence on your results of *Seasonality*, *User Promotion* and *Holidays*. De-select the checkboxes if you do not want to consider the impact of these factors in your results.
15. Optionally, in the Outlier Detection section, configure the following settings:
    a. *Configure outlier detection*: Defines whether to switch on or off outlier detection.
    b. *Acceptable distance from mean* (MOD_OUTLIER_MEAN_FACTOR): Defines the outlier detection factor to determine how far away from the mean is acceptable.
    c. *Min. non-zero observations* (MOD_OUTLIER_STD_DEV_FACTOR): Defines the outlier detection minimum number of non-zero observations, regular and promotional, counted before 0-filling.
    d. *Standard deviation factor*(MOD_OUTLIER_STD_DEV_FACTOR): Defines the outlier detection factor to determine how many deviations away from the mean is acceptable.
16. When you have configured the necessary settings, click *Done*.
17. Optionally, in the General tabbed page, set properties in the Basic section, such as a component *Alias Name* and *Description*.
18. Click the *Run Analysis*  icon.
19. When the analysis executes, click *OK* on the notification message.
20. Click the *Results* tab to view the results.

View the Results Grid:

In the *Results* tabbed page, the results grid is shown by default. For a description of each column in the grid, see Results Grid [page 80].

View the Algorithm Summary:

Click *Summary* to see an overview from the algorithm that describes the behavior of the product location combinations in relation to price elasticity. For a detailed description, see Algorithm Summary [page 81].

View the Graph

Click *Model Representation* to see a graph that displays data points for both the historical data and the forecast range. You can zoom-in on the graph to isolate any portion. For example, to zoom-in on the predicated values for the Forecasted Sales (which are graphed in yellow lines at the end of the graph), select the final portion of the Slider control bar under the graph. The graph will change focus to display the Forecasted Sales.

Export the Analysis:

You can export a Demand Forecast analysis as a stored procedure. For a detailed description, see Exporting an Analysis as a Stored Procedure [page 168].

You can now configure HANA Demand Forecasting component to forecast future sales.

Related Information

HANA Demand Forecasting Component Overview [page 75]

5.1.3 Configuring the HANA Sentiment Analysis Component

The HANA Sentiment Analysis component enables you to analyze a complex stream of text (for example, the opinions of Twitter users about a product or service). The component analyses the opinion contained in each unit of text and relays whether the sentiment is positive or negative. This way, you can transform your unstructured data into a series of easily understandable categories to discover influencing factors. From there, you can generate insights to better run your business.

Prerequisites:

SAP HANA system with Predictive Analytics Library (PAL), Automated Predictive Library (APL) and R configured. To get the latest version of SAP HANA and associated libraries, go to the SAP Product Availability Matrix.

1. In Expert Analytics, connect to a *Data Source*. For example, for an analysis of Twitter user opinions on a product or service, you could use a table called TwitterFeed.
2. In the *Predict Room*, from the Component List select *Data Preparation* - *Preprocessors* - *HANA Sentiment Analysis*. Drag-and-drop the *HANA Sentiment Analysis* component to the analysis editor. Alternatively, double-click the *HANA Sentiment Analysis* component. Click *OK*.
3. Double-click the *HANA Sentiment Analysis* component to work with its configuration settings.

   Alternatively, on the component click the *Settings*  icon and from the context menu, select *Configure Settings*.
4. In the HANA Sentiment dialog box, in the Properties panel select a *Target Variable* from the menu. Note that it is filtered to list only text columns of the following types: TEXT, BINTEXT, VARCHAR, NCLOB, CLOB or BLOB.
5. Add a *Sentiment Column Name* which is the output column name. In the example of Twitter, this is the column name into which the sentiments are written for each tweet.
6. In the Advanced panel, take the following actions in the Behavior section:
   a. Select the languages of the text for analysis. By default, it will analyse all supported languages but this can be optimized by specifying the languages contained in the data set.
   b. Select the MIME type to choose the type of text contained in your target variable. By default, it will analyse all supported MIME types but this can be optimized by specifying the MIME types contained in the data set.
   c. Choose whether or not to report the number of profanities in the analysis via the *Enable Profanities* checkbox.
   d. Map the sentiments that you are interested in for analysis. In the same section, name the sentiments for use in analysis and reporting. In the example of Twitter, you can map each sentiment as either **good** or **bad**. That way, you can work with a two-class problem. Click *Done*.
7. When configured, you can use the sentiments for analysis. For example, the analysis can be completed via a decision tree which you can add to the analysis chain from the Algorithms section of Components List panel.

> **Note**
>
> The analysis is available for display in visual support tools such as a decision tree.

8. Click the *Run Analysis*  icon. Please allow time for the analysis to complete, because during the execution a full text index is created, which can extend the execution time depending on the amount of text that is tokenized and analyzed.
9. Click the *Results* tab to view the Summary of the results.

The Summary includes the Total input records, Records with sentiments and Records without sentiments, and a breakdown of your mapped sentiments. In the Twitter example, the Summary includes a percentage of good and bad sentiments and the number of unique tokens.

You can now configure HANA Sentiment Analysis component and use it as a pre-processing step in a complex analysis.

5.1.4 Partitioning Data

The Partition component splits data sets into Train, Validate and Test partitions. It also provides flexibility by enabling you to configure the percentage of data required for each partition.

The best way to build predictive analytics models is to build the models on training (or Train) data. This way, you can tune the parameters of the algorithms while evaluating the performance of the model using the validation data set.

The models are fitted to the training data. The tuning of model parameters is based on the performance of the model on the Validate data set.

After the model parameters are optimized for best performance, the test data is used to pick the model that has the best performance on a completely unseen data set called the Test data set.

The Partition component is use on all algorithms in Expert Analytics, including PAL, APL and R in both agnostic and HANA models.

You work with the Partition component in the *Predict* room. Double-click the *Partition* component under the Data Preparation list of components on the right-hand panel. The preprocessor component is added to the analysis editor and an automatic connection is created to the data source component. From the contextual menu of the preprocessor component and choose *Configure Properties*. In the component properties dialog box, enter the necessary percentages for the Train, Test and Validate data sets and click *Done*. To view the results you click  (Run Analysis).

5.1.5 Applying Algorithms

Once you have the relevant data for analysis, you need to apply appropriate algorithms to determine patterns in the data.

Determining an appropriate algorithm to use for a specific purpose is a challenging task. You can use a combination of a number of algorithms to analyze data. For example, you can first use time series algorithms to smooth data and then use regression algorithms to find trends.

The following table provides information on which algorithm to choose for specific purposes:

Purpose	Algorithm
Performing time-based predictions	Time Series Algorithms • Single Exponential Smoothing • Double Exponential Smoothing • Triple Exponential Smoothing
Predicting continuous variables based on other variables in the data set	Regression Algorithms • Linear Regression • Exponential Regression • Geometric Regression • Logarithmic Regression • Multiple Linear Regression • Polynomial Regression • Logistic Regression
Finding frequent itemset patterns in large transactional data sets to generate association rules	Association Algorithms • Apriori • AprioriLite
Clustering observations into groups of similar itemsets	Clustering Algorithms • K-Means
Classifying and predicting one or more discrete variables based on other variables in the data set	Decision Trees • HANA C 4.5 • R-CNR Tree • CHAID
Detecting outlying values in the data set	Outlier Detection Algorithms • Inter Quartile Range • Nearest Neighbor Outlier • Anomaly Detection • Variance Test
Forecasting, classification, and statistical pattern recognition	Neural Network Algorithms • R-NNet Neural Network • R-MONMLP Neural Network

If you did not find a relevant algorithm, you can create your own component using R script within Expert Analytics which are known as R Extensions. After which, you can perform analysis on your acquired data. For more information on adding a custom component see: Specifying properties with the R Extension Wizard [page 92]

1. In the *Predict* room, double-click the required algorithm component under the list of components on the right.
   The algorithm component is added to the analysis editor and is connected to the previous component in the analysis.
2. From the contextual menu of the algorithm component and choose *Configure Properties*.

3. In the component properties dialog box, enter the necessary details for the algorithm component properties.
4. Choose *Done*.
5. To view the results of the analysis, choose  (Run Analysis).

5.1.6 Optional: Storing Results of Analysis

You can store the results of the analysis in flat files or databases for further analysis using data writer components. Only the table view is stored in the data writer component.

1. In the *Predict* room, double-click the required data writer component under the list of components on the right.
   The data writer component is added to the analysis editor and is connected to the previous component in the analysis.
2. From the contextual menu of the data writer component and choose *Configure Properties*.
3. In the component properties dialog box, enter the necessary details for the data writer component properties.
4. Choose *Done*.
5. To view the results of the analysis, choose  (Run Analysis).

5.2 Running the Analysis

Once you have prepared your data and applied the necessary algorithms, you can run an analysis.

- To run the analysis, choose  (Run Analysis) in the analysis editor toolbar.
- If your analysis is very large and complex, you can run the analysis, component-by-component and analyze the data. To run a part of the analysis, choose *Run up to Here* from the contextual menu of the component until which you want to run.

5.3 Saving the Analysis

After creating an analysis, you can save it for reusing it in the future. In Expert Analytics, you need to save the document to save the analyses you create. The saved document contains data set, analyses, results, and visualizations. The document is saved in the .lums file format.

To save an analysis in a document, perform the following steps:

1. Choose *File* > *Save*.

2. Enter a name for the document.
3. Choose *Save*.

If you create multiple analyses using the same data set, all the analyses are saved in the same document. You can access all the analyses in a document through the Analysis dropdown list.

5.4 Deleting an Analysis from the Document

You can delete an analyis if it is no longer needed.

To delete an existing analysis from the document, hover on the analysis' image in the analysis bar, and choose ✕

5.5 Viewing Results and Exporting an Analysis

To view the results of components in an analysis, after running the analysis, switch to the *Results* view or from the contextual menu of the component, select *View Results*.

For a detailed description of how to export a chain as a stored procedure, see Exporting an Analysis as a Stored Procedure [page 168].

6 Creating R Extensions and PAL Components

6.1 Creating an R Extension

As an expert user, you can create and add a component using R scripts.

R is a software programming language and environment for statistical computing and graphics. Expert Analytics provides an environment for you to use R scripts (within a valid R function format) and create components called an R Extension. You can use R Extensions for analysis as with any other existing component.

While creating an R Extension you give it a unique name. It appears in the Predict Room under the classification, *Algorithms* > *R Extensions*.

The R Extension can be classified as an algorithm, a preprocessor component, or a data writer. From version 3.1.2. of R, you can use the libraries such as algorithms, visualization, data manipulation, and preparation. What's more, you can share one or more R Extension by using .spar files.

The R Extension has an expanding window and keyword highlighting to make script easier to read in the component. You use the Expand window button is used to activate the window:



Read on to learn how to create and use R Extensions.

6.1.1 Specifying properties with the R Extension Wizard

You can specify properties for an R Extension through a wizard.

R Extension Creation Wizard Properties

Note: Predictive Composer R Extension Wizard properties are marked with an asterisk *.

Property	Description
Extension Name*	Enter a name for the extension. **Note** You cannot rename an existing extension.

Property	Description
Extension Type*	Select the type of the extension.
Extension Category	Select a category under which to group your new extension.
Extension Description*	Enter a description of the extension, which will appear as the tooltip for the created extension.

Property	Description
Load R Script	Click to load an R script.
Script Editor*	Copy and paste or write the R script in the text box.
Primary Function Name*	Select the name of the function that you want to execute.
Input DataFrame*	Select the Input DataFrame from the list of parameters.
Output DataFrame*	Enter a name for the variable that you want to use as OutputDataFrame.
Model Variable Name*	Enter a name for the variable that you want to use as model variable.
Show Visualization*	To display the results of the R Extension execution in chart format, select this option.
Show Summary*	To display the algorithm summary after the R Extension execution, select this option.
Option to save the model*	To allow the *Save as Model* option for the R Extension, select this checkbox. **Note** If you select *Option to save the model*, the *Model Variable Name* field is enabled, and *Model Scoring Function Details* appears.
Option to Export as PMML	To allow the *Export as PMML* option for the R Extension, select this checkbox. **Note** The *Option to Export as PMML* checkbox is only enabled, if you select the *Option to save the model*.
Model Scoring Function Name*	Select the name of the model scoring function that you want to execute. **Note** Model Scoring Function fields are only visible if you selected the *Option to save the model* checkbox.
Input DataFrame*	Select the Input DataFrame from the list of parameters.
Output DataFrame*	Enter a name for the variable that you want to use as Output DataFrame.

Property	Description
Input Model Variable Name*	Select the Input Model Variable Name from the list of parameters.

Property	Description
Primary Function - Output Table Definition*	
Consider all columns from previous component (or consider none)*	Select to include or exclude respectively the predicted column of the parent component in the output of the extension.
Data Type*	Select the *Data type* for the predicted column of the extension.
New Predicted Column Name*	Enter a name for the predicted column, which is the output column of the extension.
Primary Function - Property View Definition*	
Function Parameters*	Defined parameters.
Property Display Name*	Enter a name for the *Independent Column* and the *Dependent column*, which will appear in the property view of the extension.
Control Type*	Select the *Control Type* for the *Independent Column* and the *Dependent column*.
Model Scoring - Output Table Definition*	
Consider all columns from previous component (or consider none)*	Select to include or exclude respectively the predicted column of the parent component in the output of model scoring.
Data Type*	Select the *Data type* for the predicted column of model scoring.
New Predicted Column Name*	Enter a name for the predicted column, which is the output column of model scoring.
Model Scoring - Property View Definition*	
Function Parameters*	Defined parameters.
Property Display Name*	Enter a name for the column that appears in the property view of the saved model.
Control Type*	Select the *Control Type* for the *Independent Column* and the *Dependent column*.

Related Information

Creating an R Extension [page 95]

6.1.2 Creating an R Extension

How to create an R Extension for use in analyses.

Before creating an R Extension, ensure that the following requirements are met:

- The R script is written in a valid R function format.
- The R script executes in the R GUI console.
- The R script has at least one main function.
- Install the packages required to run the R script either on your machine or on the SAP HANA server.
- The R script written for In-Database analysis returns a DataFrame.

Following are the best practices you should consider while writing the R script:

- The R script written for In-Proc analysis returns a DataFrame.
- Type conversion of the output is recommended; for example, if a column has numeric values, mention it as *as.numeric(output)*
- For categorical variables used in the R script, specify the variable using *as.factor* command.

1. In the *Predict* room, under the list of components on the right-hand side of the screen, select the + icon and in the resulting menu, select *R Extension*.
   The *Create New R Extension* wizard displays.
2. On the *General* page, enter the following information:
   a. In the *Component Name* text box, enter **`My component`**.
   b. From the *Component Type* dropdown list, select *Algorithms*.
   c. In the *Component Description* text box, enter **`R component for Simple Linear Regression`**.
3. Choose *Next*.
   The *Script* page appears.
4. On the *Script* page, choose *Load Script* to select a file to upload.

> **Note**
>
> You can write or copy and paste the following sample R script in the text box.

> **Note**
>
> Refer to the comments in the following R function format to help you understand and write your own R script.

```
#The following is a sample script for a simple linear regression component.
#You must write the script in a valid R function format.
#Note that the function name and variable name in R script can be user-
defined, and are supported in R.
#The following is the argument description for the primary function SLR:
#InputDataFrame: Dataframe in R that contains the output of the parent
component.
#The following two parameters are received from the user through the
property view:
#IndependentColumn: Column name that you want to use as independent
variable for the component.
#DependentColumn; Column name that you want to use as a dependent variable
for the component.
SLR<-function(InputDataFrame, IndependentColumn, DependentColumn)
```

```
{
  finalString<-paste(paste(DependentColumn,"~" ), IndependentColumn);
#Formatting the final string to
#pass to "lm" function
slr_model<-lm(finalString); # calling the "lm" function and storing the
output model in "slr_model"
#To get the predicted values for the Training data set, call the "predict"
function with this model and
#input dataframe, which is represented by "InputDataFrame".
result<-predict(slr_model, InputDataFrame); # Storing the predicted values
in the "result" variable.
output<- cbind(InputDataFrame, result); # combining "InputDataFrame" and
"result" to get the final table.
plot(slr_model); #Plotting model visualization.
#returnvalue: function must always return a list that contains results
"out", and model variable
#"slrmodel", if present.
#The output variable stores the final result.
#The model variable is used for model scoring.
return (list(slrmodel=slr_model, out=output))
}
#The following is the argument description for the model scoring function
"SLRModelScoring":
#InputDataFrame: Dataframe in R that contains the output of the parent
component.
#IndependentColumn: Column name to be used as independent variables for the
component.
#Model: Model variable that is used for scoring.
SLRModelScoring <- function (InputDataFrame, IndependentColumn, Model)
{
#Calling "predict" function to get the predictive value with "Model " and
"InputDataFrame".
predicted<-predict(Model, data.frame(InputDataFrame[,IndependentColumn]),
level=0.95);
# Combining "InputDataFrame" and "predicted" to get the final table.
output <- cbind(InputDataFrame, predicted);
#returnvalue: function should always return a list that contains the result
("model result"),
#The output variable stores the final result
return(list(modelresult=output))
}
```

Two examples of converting an R script to a valid R function format, recognized by Expert Analytics are given below:

R script	R function format (recognized by Expert Analytics)

`dataFrame<-read.csv("C:\\CSVs\ \Iris.csv")`
`attach(dataFrame)`
`set.seed(4321)`
`kmeans_model<-`
 	
`kmeans(data.frame(`SepalLength`,`SepalWidth`,`PetalLength`,`PetalWidth`),`
 	
`centers=5,iter.max=100,nstart=1,algorithm=`
`"Hartigan-Wong")`
`kmeans_model$cluster`
	`kmeansfunction<-`
	`function(dataFrame,independent,`
	
	`Clustersize,Iterations,algotype,numberofinitialdsets)`
	`{`
	`set.seed(4321)`
	`kmeans_model<-`
	`kmeans(data.frame(dataFrame[,independent]),`
	
	`centers=Clustersize,iter.max=Iterations, nstart=numberofinitialdsets,`
	`algorithm= algotype)`
	`output<- cbind(dataFrame,`
	`kmeans_model$cluster);`

R script	R function format (recognized by Expert Analytics)
	boxplot(output); return (list(out=output)); }
dataFrame<- read.csv("C:\\Datasets\\cnr\ \Iris.csv") attach(dataFrame) library(rpart) cnr_model<-rpart (Species~PetalLength+PetalWidth +SepalLength+ SepalWidth, method="class") library(rpart) predict(cnr_model, dataFrame,type = c("class"))	cnrFunction<- function(dataFrame,IndependentColumns,dep) { library(rpart); formattedString<- paste(IndependentColumns, collapse = '+'); finalString<-paste(paste(dep, "~"), formattedString); cnr_model<- rpart(finalString, method="class"); output<- predict(cnr_model, dataFrame,type=c("class")); out<- cbind(dataFrame, output); return (list(result=out,modelcnr=cnr_model)); } cnrFunctionmodel<- function(dataFrame,ind,modelcnr,type) { output<- predict(modelcnr,data.frame(dataFrame[,ind]),type=type); out<- cbind(dataFrame, output); return (list(result=out));

> **Note**
>
> Declare parameters for the model scoring function in the primary function, except for Input Dataframe and Input Model Variable Name, which you select from the dropdown lists.

5. In the *Primary Function Details* section, enter the following information:
   a. From the *Primary Function Name* dropdown list, select *SLR*.
   b. From the *Input DataFrame* dropdown list, select *InputDataFrame*.
   c. In the *Output DataFrame* box, enter **`out`**.
   d. Select the *Option to save the model* checkbox.
      The *Model Variable Name* field is enabled, and *Model Scoring Function Details* appears.
   e. In the *Model Variable Name* field, enter **`slrmodel`**.
   f. Select the *Show Summary* and *Option to export as PMML* checkboxes.
6. In the *Model Scoring Function Details* section, enter the following information:
   a. From the *Model Scoring Function Name*, select *SLRModelScoring*.
   b. From the *Input DataFrame* dropdown list, select *MInputDataFrame*.
   c. In the *Output DataFrame* field, enter **`modelresult`**.
   d. From the *Input Model Variable Name* dropdown list, select *Model*.
7. Choose, *Next*.

The *Settings* page appears.

8. In the *Output Table Definition* section of *Primary Function Settings*, perform the following substeps:
   a. Choose *Consider None*.
   b. From the *Data Type* dropdown list, select *Integer*.
   c. In the *New Predicted Column Name* box, enter **`Predicted column`**.
9. In the *Property View Definition* section, perform the following substeps:
   a. In the *Property Display Name*, in the *Independent column* box, enter **`Independent Column`**.
   b. From the *Control Type* dropdown list, select *Column Selector (Single)* as the control type for the *Independent column*.
   c. In the *Property Display Name*, In *Independent column* box, enter **`Dependent Column`**.
   d. From the *Control Type* dropdown list, select *Column Selector (Single)* control type for *Dependent column*.
10. In the *Output Table Definition* section of *Model Scoring Settings*, choose *Consider all columns from previous component*.
11. From the *Data Type* dropdown list, select *Integer*.
12. In the *New Predicted Column Name*, enter *Output Column*.
13. In the *Property View Definition* section, perform the following substeps:
    a. In the *Property Display Name*, enter **`Independent column`**.
    b. From the *Control Type* dropdown list, select *Column Selector (Single)* as the control type for the *Independent column*.
14. Choose *Finish*.

Depending on the type of analysis performed, you can create a model just like any other component.

Related Information

Specifying properties with the R Extension Wizard [page 92]
Models [page 272]
Creating a Model [page 153]

6.1.3 Plotting Multiple Charts in R Extensions

Multiple charts in R Extensions can be plotted in non-HANA scenarios by using the `multiplot` feature.

You can now plot multiple charts in R Extensions. The feature does not incur any changes in the procedure and the UI components for creating an R Extension. However, there are some rules that you must follow when writing a custom R function that uses the `multiplot` feature. The rules are listed below, followed by a custom R function as an example to illustrate the writing principles:

- Load `ggplot2` package in custom R function and use the `ggplot2` plotting functions for `multiplot`, for example, `qplot` and `ggplot`.
- Set Expert Analytics to the `multiplot` mode using `pa.config("multiplot","true")` in the custom R function.

- Assign each chart to a variable and return all chart variables as a list in the `return` statement.

The following R code shows how to plot multiple charts in the custom R function. For the purpose of demonstrating the `multiplot` feature, no predictive algorithms are used in this R function, but you can add predictive algorithms as indicated in the inline comments.

> **Note**
>
> This sample R function is hard-coded to use the `mtcars` data set. To run this sample R function, you need to load the `mtcars` data set first.

```
myanalysisfunction <- function(mydataframe){
# Load ggplot2 library  library(ggplot2)
# Set PA to multiplot mode  pa.config("multiplot","true")
########################################
#   Put predictive algorithms here
########################################
# Plot a histogram chart using qplot and assign the chart to variable my_p1
my_p1 <- qplot(mydataframe$hp, geom="histogram")
# Plot a scatterplot chart and assign the chart to variable my_p2
my_p2 <- qplot(mydataframe$wt, mydataframe$hp)
# Plot a simple linear model and assign the chart to variable my_p3
my_model <- lm(mydataframe$wt ~ mydataframe$hp)
my_p3 <- qplot(hp, wt, data = mydataframe) + geom_abline(intercept=coef(my_model)
[1],
slope=coef(my_model)[2])
# plot a scatterplot chart with a linear smooth line and assign the chart to
variable my_p6
my_p5 <- ggplot(mydataframe, aes(x = wt, y=mpg), .~cyl)+ geom_point()
my_p6 <- my_p5 + geom_smooth(aes(group=cyl),method="lm")
# plot a pie chart and assign the chart to variable my_p7
my_p7 <- ggplot(data=mydataframe, aes(x=factor(1), fill= factor(cyl))) +
geom_bar(width=1)+
coord_polar(theta="y")
# plot a time series chart and assign it to chart variable my_p8
year <- as.numeric(unlist(strsplit("1998 1999 2000 2001 2002 2003 2004 2005 2006
2007", "\\s+")))
revenue <- as.numeric(unlist(strsplit("10   6     13   14    12    8     10
10    6     9", "\\s+")))
mydataframe2 <- data.frame(year, revenue)
my_p8 <- ggplot(mydataframe2, aes(year, revenue)) + geom_bar(stat="identity",
fill="white",
colour="black") + geom_line(colour="red") + stat_smooth(se=F, size=3)
# Return all chart variables and set metadata for each chart
return(list(out=mydataframe, charts=list(list(chart=my_p1, type="bar",
name="chart 1"),
list(chart=my_p2, type="scatter plot", name="chart 2"), list(chart=my_p3,
type="line", name="chart 3"),
list(chart=my_p8, type="time series", name="chart 8"), list(chart=my_p7,
type="pie",
name="chart 7"), list(chart=my_p6, type="line", name="chart 6"))))
}
```

You must return the charts to be plotted in the `return` statement as elements of the charts list. The `return` statement is a high-dimensional list that looks like this:

```
return(list(…, charts=list(…, list(chart=chart_variable_1,name=”chart_name_1”,
type=”chart_type_1”), list(chart=chart_variable_2,name=”chart_name_2”,
type=”chart_type_2”), …), …))
```

Each element of the charts list is also a list which contains information about a chart to be plotted, including chart variable (chart element), chart name (name element), and chart type (type element). Chart name and

chart type provide information for users to distinguish charts from each other on the result window. Different icons are used for different chart types and the chart name displays as a tooltip when the cursor hovers over the icon. You do not have to set chart name and chart type to plot multiple charts, but it is highly recommended.

Expert Analytics currently supports the following chart types:

- `bar`
- `horizontal bar`
- `bubble`
- `confusion matrix`
- `gain`
- `lift`
- `line`
- `model accuracy`
- `parallel coordinates`
- `scatter plot`
- `time series`

For other chart types that are not included in the list above, a default icon will be used for the chart.

The `multiplot` feature does not change the existing (single) plot and there is no need to change the existing custom R function if only (single) plot is needed.

6.2 Creating a PAL Component

As an expert user, you can create a new SAP HANA Predictive Analysis Library (PAL) component. This action enables other users to add the PAL components to their analyses.

The PAL component is made available in the *Predict* room by clicking the plus (+) sign in the list of components in the right-hand panel.

You create the component by using algorithms from the SAP HANA Predictive Analysis Library. The following PAL algorithms are currently supported in Expert Analytics:

Algorithm	Corresponding Function
ABC Analysis	ABC
Agglomerate Hierarchical Clustering	HCAGGLOMERATE
ARIMA	ARIMATRAIN
Binning	BINNING
C4.5 Decision Tree	CREATEDT
Exponential Regression	EXPREGRESSION
FP-Growth	FPGROWTH
K-Medoids	KMEDOIDS

Algorithm	Corresponding Function
Naive Bayes	NBCTRAIN
Scaling Range	SCALINGRANGE

You can share one or more PAL components by using .spar files.

> **Note**
>
> You have the option to save C4.5 Decision Tree and Exponential Regression components as a model.

> **Note**
>
> For more information about the PAL algorithms and the uses of each algorithm, see *SAP HANA Predictive Analysis Library (PAL) Reference* at http://help.sap.com/hana_one

6.2.1 Specifying Properties with the PAL Component Creation Wizard

You can specify properties for the SAP HANA Predictive Analysis Library (PAL) component through a wizard.

PAL Component Creation Wizard Properties

Property	Description
Component Name	Enter a name for the component. **Note** You cannot rename an existing component.
Component Type	Select the type of the component. **Note** You cannot edit the component type.
Component Description	Enter a description of the component, which will appear as the tooltip for the new component.

Property	Description
Area	Library area name. This defaults to AFLPAL.

Property	Description
Function	Select one of the functions from the dropdown list to set parameters for that function: • ABC • HCAGGLOMERATE • ARIMATRAIN • BINNING • CREATEDT • EXPREGRESSION • FPGROWTH • KMEDOIDS • NBCTRAIN • SCALINGRANGE
Input Parameters	**Note** Parameters are dependent on the function that is selected.
Parameter Name	**Note** Parameter names are dependent on the function that is selected. Parameter names are not editable.
Display Name	Enter an alternative name to be shown for the parameter.
Default Value	Enter a default value for the parameter. Depending on the data type of the parameter, this can be a numeric or text value. **Note** Default values cannot be entered when the parameter is dependent on the input data.
Data Type	This can be string, integer, or double, depending on the parameter. Data types are not editable.
Include Parameter	• For mandatory parameters: This checkbox is selected by default and cannot be deselected. • For optional parameters: This checkbox is selected by default, but you can deselect it for parameters you do not want to include in the component.
Model Parameters	**Note** Model parameters can only be specified for certain functions. The properties are the same as for the Input Parameters: Parameter Name, Display Name, Default Value, Data Type, and Include Parameter.

6.2.2 Creating a PAL Component

How to create a SAP HANA Predictive Analysis Library (PAL) component for use in analyses.

To create a PAL component, you need to be connected to a SAP HANA data source.

1. In the *Predict* room, under the list of components on the right, choose  > *PAL Component*
2. On the *General* page, enter the following information:
   a. Enter a unique *Component Name*.
   b. Select *Component Type*.

      This is defaulted to *Algorithms*.
   c. Enter an optional *Component Description*.

      This appears as a tooltip when you hover over the created component in the list of components on the right.
3. Choose *Next*.

   The *Function Settings* page appears.
4. Select a *Function* from the dropdown list.

   > **Note**
   >
   > The *Area* is defaulted to AFLPAL.
5. Optional: Enter a *Display Name* for any of the listed parameters.
6. Optional: Enter a *Default Value* for any of the listed parameters.

   > **Note**
   >
   > The default value can be a numeric or text value depending on the *Data Type* specified for the parameter. *Default Value* field is greyed out for parameters that are dependent on input data.
7. For optional parameters, you can deselect the *Include Parameter* checkbox.

   > **Note**
   >
   > The *Include Parameter* checkbox cannot be deselected for mandatory parameters.
8. For functions that include model parameters, you can specify the following for each parameter:
   a. Display Name
   b. Default Value
   c. Include Parameter
9. Click *Finish*.

   The component is listed under *Algorithms* > *Custom PAL Components* in the list of components on the right. It can be added to an analysis like any other component. You can configure the component when it is added to an analysis by selecting *Configure Settings* from the context menu.

You can edit the PAL component if you need to make changes to the component details.

7 Sharing and Consuming R Extensions

7.1 Sharing and Consuming R Extensions

You can enable and publish a partner extension with R script.

R Extensions can help to solve predictive problems by providing the means to prepare and enrich data, as well as to create algorithms, visualizations, reports and more. You can write a new extension and share it with partners in R. In addition, you can edit a saved self-authored R Extension before exporting it.

To protect your custom R scripts and extension metadata, the files are encrypted both as .spar files and artefacts on your local file system. When the author saves the R Extension, it is encrypted and updated on the local disk. The script is decrypted before its handed over to R for execution, in memory.

After an author creates the extension, its name is marked with an EXT identifier in the Component List under the appropriate section.

You can share and consume R Partner Extensions at the SAP Analytics Extensions Directory. Here you can browse and import custom predictive extensions and models created for SAP BusinessObjects Predictive Analytics.

Related Information

Sharing an R Extension [page 104]
Browsing for and Consuming R Extensions [page 106]
Editing an R Extension [page 107]

7.2 Sharing an R Extension

You can enable a partner extension with R script and publish it to the SAP Analytics Extensions Directory, where you can also consume R Extensions written by others.

The following are the system requirements and configuration process to create and export an R Extension.

Prerequisites:

SAP HANA system with Predictive Analytics Library (PAL), Automated Predictive Library (APL) and R configured.

Configuration Process:

To create or edit an R Extension, take the following steps:

1. Load a data set to Expert Analytics.
2. Go to the **Predict** room.
3. Below the Components List on the right-hand side of the window, click the + icon to reveal the context menu. From the menu, select *R Extension* to launch the editor to create a custom R Extension.
4. The Create New R Extension dialog box opens at the *General* tabbed page. Configure the following settings:
    a. Add an *Extension Name*.
    b. Add an *Extension Type*. Choose *Data Writer* (for agnostic only) and *Data Prepare* (for SAP HANA and agnostic).
    c. If you chose Algorithms as the Extensions Type, select an algorithm *Category* to which the extension is added. Choose from the following predefined categories: *Association*, *Classification*, *Clustering*, *R Extensions*, *Decision Trees*, *Neural Networks*, *Outliers*, *Regression* and *Time Series*. Note that in Expert Analytics version 2.5, for the Data Writer and Data Prepare extension types, the Category field is disabled and the only option is *R Extensions*.
    d. Optionally, enter a description to the text box, *Extension Description*.
    e. To enable recipients to edit, select the checkbox, *Make editable when shared*.
5. Click *Next*.
6. In the *Script* tabbed page, choose from the following options:
    a. Add/Write R script to the *Script Editor*.
    b. Set the input parameters as necessary. They include *Primary Function Name*, *Input DataFrame*, *Output DataFrame* and *Model Variable Name*.
    c. Select the checkboxes as necessary. They include *Show the Visualization*, *Show Summary* and *Option to save the model*.
7. Click *Next*.
8. In the Settings tabbed page, configure the output parameters to appear in the configuration panel at a future time when the extension is consumed in a chain. Select from the following options:
    a. In the Output Table Definitions, select one of the following checkboxes: *Consider all columns from previous component* or *Consider none*. The component generates one predicted column and your selection will affect the Results Tab. For example, suppose the previous component contains five columns. If you consider all columns from the previous component, the Results tab will display the five previous component columns plus the newly-generated predicted column. In this example, that means six columns. If you consider no columns from the previous component, the results tab will display only the newly generated column.
    b. Choose a *Data Type* for each output column name.
    c. Enter a *New Predicted Column Name* for each output column name.
9. Configure the input parameters to appear in the configuration panel at a future time when the extension is consumed in a chain. In *Property View Definitions*, select from the following options:
    a. *Function Parameters*: The parameters of the primary function or the model score function.
    b. *Property Display Name*: Optionally, enter a property display name for any function a parameter to appear in the configuration window. Use to help users make a more intuitive selection.
    c. *Control Type*: Enables you to decide how you want users to input this parameter. The input methods include text box, dropdown box, combo box, single column selector and multiple column selector.

d. *Settings* icon : Use to set a mandatory field and choose the default input data type from the following list: *String*, *Integer*, *Double* and *R-Literal*.

10. Click *Finish*. The R Extension is encrypted and saved on the local file system. The component list in Expert Analytics is updated with the new R Extension name under the appropriate category.
11. The updated R Extension appears in the Component List under the appropriate section. (For example, if you create Data Writers, it appears in the Data Writers section.) Take any or all of the following actions:
    a. Identify the R Extension in the menu via the EXT identifier.
    b. View the R Extension description by clicking the extension name.
    c. *Export* to a .spar file for sharing.
    d. *Save* the file via File Explorer.
    e. *Share* it via the Portal at this address: http://www.sap.com/bi-partner-extensions.
    f. *Edit* in the R Extension editor window. Option, available only to the author.
    g. *Delete*. This option cannot be undone. The associated files are deleted from the system immediately.

You have enabled partner extensions with R script and seen how easily -- and securely -- extensions can be shared with partners.

Related Information

http://www.sap.com/bi-partner-extensions?product=predictive
Browsing for and Consuming R Extensions [page 106]
Editing an R Extension [page 107]
Exporting a Single R Extension as a Stored Procedure from Expert Analytics [page 166]

7.3 Browsing for and Consuming R Extensions

You can search for and download R Extensions, and models, from SAP Analytics Extensions Directory. You can download extensions to your local drive and access them through Expert Analytics.

Take the following steps to import an extension or model from the directory:

1. Browse to the *Predictive Analytics* page of the SAP Analytics Extensions Directory.
2. Search for the R Extension that you want and use the download link provided to export its .spar file to your local file directory.
3. Open Expert Analytics. In *Predict* > *Components List* >, click the + button to open the context menu.
4. From the context menu, click *Import Model/Extension* and select a downloaded R Extension in a .spar file from your local file system.
5. In the resulting *Import Model/Extension* dialog box that displays in Expert Analytics, confirm the R Extension that you want to import and click *Finish*.
6. The R Extension is imported into Expert Analytics. Access it from the Components List under the appropriate category type; for example, *Algorithms - Classification*. Note that Imported R Extensions are identified by the EXT indicator.

You can add your imported R Extension to an analysis chain in the Predict Room workspace. Editing of the imported R Extension is possible if the author enabled it (by selecting *Make editable when shared* on the Create New R Extension dialog box).

Related Information

Sharing an R Extension [page 104]
Editing an R Extension [page 107]

7.4 Editing an R Extension

Authors can edit an R Extension that have not yet been exported from Expert Analytics.

Authors can edit their own R extensions before they are shared. For this purpose, the author's R extension script is securely encrypted in a designated location on their own file system. This means that the author must edit the extension only in their own workspace. However, when the extension has been exported to the SAP Analytics Extensions Directory, the author can no longer edit that version. This is true even if the author imports their own .spar file in an attempt to edit it. To change the R Extension after export, an author can update the stored extension on their local drive and re-export the new version. A consumer can import an R Extension and use it in a an analysis chain, but not edit the extension.

Take the following steps to edit an R Extension:

1. In Expert Analytics, go to the **Predict** room.
2. Below the Components List on the right-hand side of the window, click the + icon to reveal the context menu. From the menu, select an extension denoted by the EXT identifier.
3. Click *Edit*.
4. The Edit New R Extension dialog box opens at the *General* tabbed page. Configure any of the following settings:
    a. Change the *Extension Name*.
    b. Add an *Extension Type*. Choose *Data Writer* (agnostic only) and *Data Prepare* (HANA and agnostic).
    c. If you chose Algorithms as the Extension Type, select an algorithm *Category* to which the extension is added. Choose from the following categories: *Association*, *Classification*, *Clustering*, *R Extensions*, *Decision Trees*, *Neural Networks*, *Outliers*, *Regression* or *Time Series*. Note that for the Data Writer and Data Prepare extension types, in Expert Analytics version 2.5 only, the Category field is disabled and the only option is *R Extensions*.
    d. Optionally, change or add an *Extension Description*.
    e. Optionally, enable recipients to edit by selecting the checkbox, *Make editable when shared*.
5. Click *Next.*
6. In the *Script* tabbed page, edit any of the following options:
    a. Change the R script in the *Script Editor*.
    b. Change the parameters as necessary, which include *Primary Function Name*, *Input DataFrame*, *Output DataFrame* and *Model Variable Name*. Declare parameters for the model scoring function in the

primary function, except for Input Dataframe and Input Model Variable Name, which you select from drop-down lists

c. Change the checkboxes as necessary, which include *Show the Visualization*, *Show Summary* and *Option to save the model*.

7. Click *Next*.
8. In the Settings tabbed page, configure any of the output parameters to appear in the configuration panel at a future time when the extension is consumed in a chain. Select from the following options:
   a. In the Output Table Definitions, select one of the following checkboxes: *Consider all columns from previous component* or *Consider none*. The component generates one predicted column and your selection will affect the Results Tab. For example, suppose the previous component contains five columns. If you consider all columns from the previous component, the Results tab will display the five previous component columns plus the newly-generated predicted column. In this example, that means six columns. If you consider no columns from the previous component, the results tab will display only the newly generated column.
   a. Choose a *Data Type* for each output column name.
   b. Enter a *New Predicted Column Name* for each output column name.
9. Configure any of the input parameters to appear in the configuration panel at a future time when the extension is consumed in a chain. In *Property View Definitions*, select from the following options:
   a.
   b. *Property Display Name*: Optionally, enter a property display name for any function a parameter to appear in the configuration window. Use to help users make a more intuitive selection.
   c. *Control Type*: Enables you to decide how you want users to input this parameter. The input methods include text box, dropdown box, combo box, single column selector and multiple column selector.
   d. *Settings* icon : Use to set a mandatory field and choose the default input data type from the following list: *String*, *Integer*, *Double* and *R-Literal*.
10. Click *Finish*. The R Extension is updated with your latest changes. The script is encrypted and saved on the local file system. The component list in Expert Analytics is updated with the new R Extension.
11. The updated Extension appears in the Component List under the appropriate section. For example, if you create Data Writers, it appears in the Data Writers section. Take any or all of the following actions:
    a. Identify the extension in the menu via the EXT identifier.
    b. View the extension description by clicking the R Extension name.
    c. *Export* to a .spar file for sharing.
    d. *Save* the file via File Explorer.
    e. *Share* it via the Portal at this address: http://www.sap.com/bi-partner-extensions.
    f. *Edit* in the R Extension editor window. Option, available only to the author.
    g. *Delete*. This option cannot be undone. The associated files are deleted from the system immediately.

You have edited your R Extension.

Related Information

http://www.sap.com/bi-partner-extensions?product=predictive

Sharing an R Extension [page 104]

Browsing for and Consuming R Extensions [page 106]

8 Viewing the Results of Analysis

8.1 Analyzing Data

After you have run the analysis, the result of each component in the analysis is represented using different visualization charts.

To analyze data, perform the following steps:

1. After running an analysis, switch to the Results view by choosing the *Results* button in the toolbar.
2. To view the visualization for a component, choose the required component in the analysis from the list of components on the right.

By default, the result of the component is displayed in the *Table* view.

The following table summarizes components and their supported visualization charts.

Components	Visualization Charts
Data Sources and Preprocessors	Scatter Matrix Chart, Statistical Summary Chart, Parallel Coordinates
Clustering Algorithms	Cluster Representation Charts and Algorithm Summary
Decision Trees	Decision Tree, Algorithm Summary, Confusion Matrix
Time Series Algorithms	Trend Chart, Algorithm Summary
Regression Algorithms	Trend Chart, Algorithm Summary
Association Algorithms	Apriori Tag Cloud Chart, Algorithm Summary

The following table summarizes the supported data points for visualizations:

> **Note**
>
> If the input dataset exceeds the interactivity data point limit, the charts are rendered without interactivity. If the input dataset exceeds the maximum data point limit, the data above the limit is not shown in the chart.

Charts	Maximum Number of Data Points Supported	
	With Interactivity	Without Interactivity
Trend Chart	4000	6000
Scatter Matrix Chart	500	1000
Parallel Coordinate Chart	60000	75000

8.2 Scatter Matrix Chart

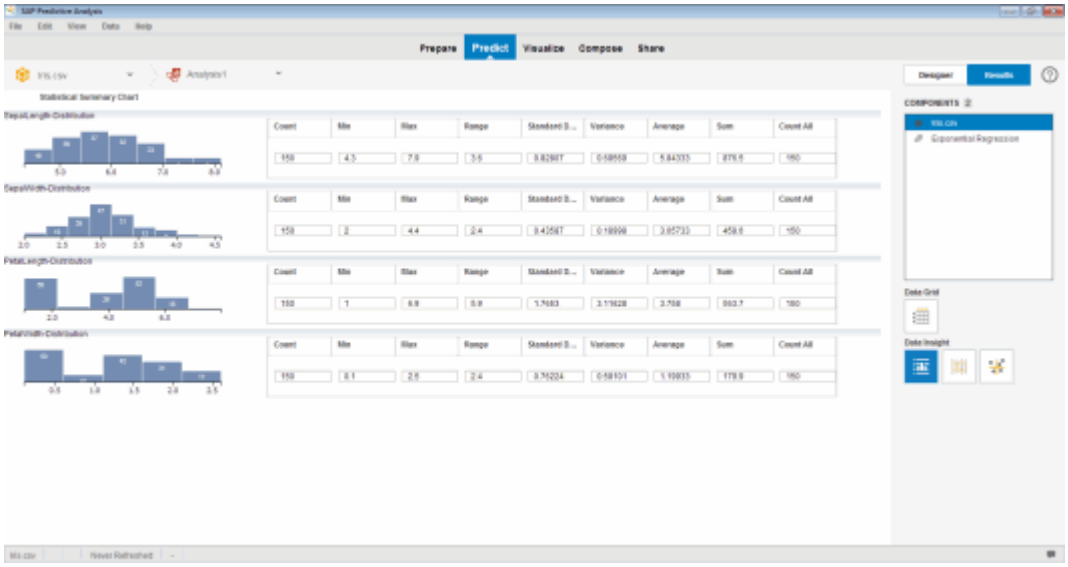
Scatter matrix charts are matrices of charts (n*n charts, where n is the number of selected attributes) used to compare data across different dimensions. By default, a maximum of three numerical attributes are selected for analysis, starting from the first attribute from the source data, and a 3*3 matrix of charts are plotted. However, you can manually select the required attributes from *Measures* in the *Data* section and refresh the visualization by choosing *Apply*.

> **Note**
>
> You can select a maximum of three numerical attributes from *Measure* in the *Data* section.



8.3 Statistical Summary Chart

Statistical Summary provides summary information for numerical attributes in the data source. The summary information includes count, minimum value, maximum value, variance, standard deviation, sum, average, range, and number of records. For HANA online data sources, the two additional parameters such as skewness and kurtosis are also included in the summary. A histogram chart is plotted for each attribute.

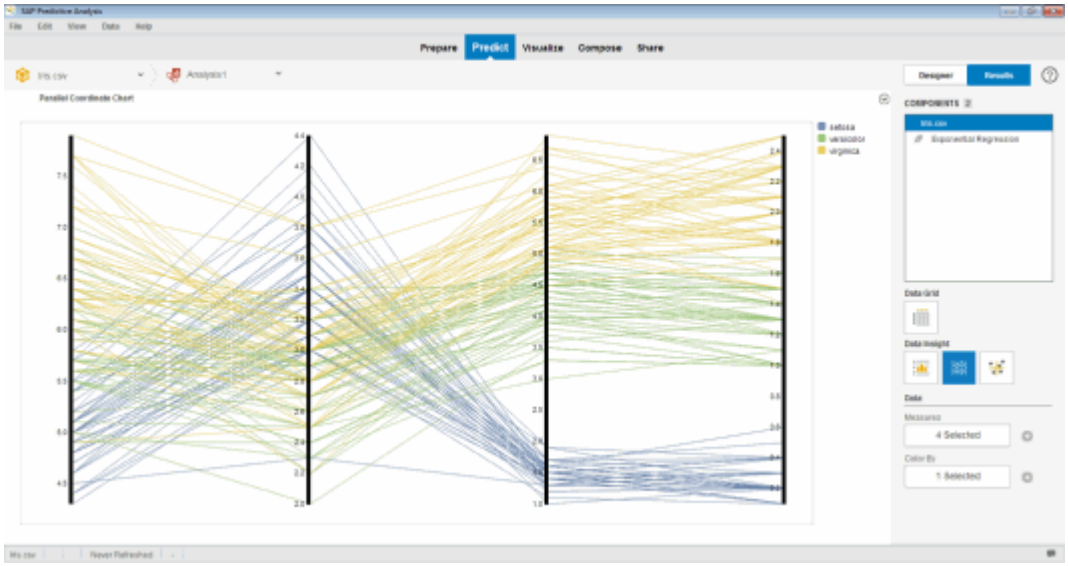


8.4 Parallel Coordinates

Parallel coordinates is a visualization technique used to visualize multi-dimensional data and multivariate patterns in the data for analysis.

In this chart, by default, the first seven attributes are represented as vertically-spaced parallel axes. You can manually select the required attributes from *Measures* and refresh the chart by choosing *Apply*. Each axis is labeled with the attribute name, and minimum and maximum values for attributes. Each observation is represented as a series of connected points along the parallel axes. You can select the color by option to filter the data based on the categorical value.

> **Note**
>
> You can select a maximum of seven numerical attributes in the *Measures* section.



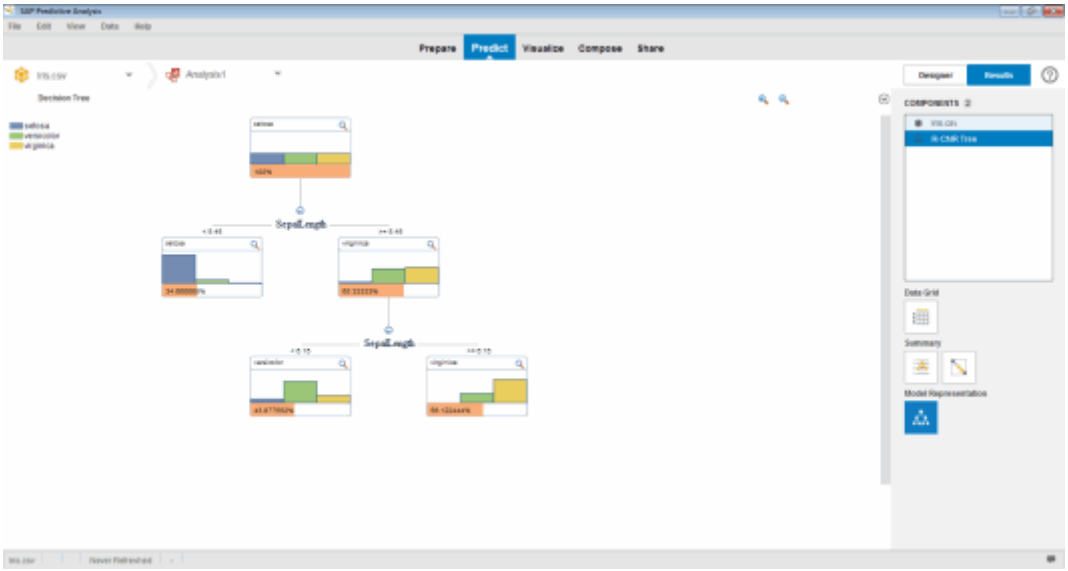
8.5 Decision Tree

A decision tree is a visualization technique that enables you to classify observations into groups and predict future events based on the set of decision rules.

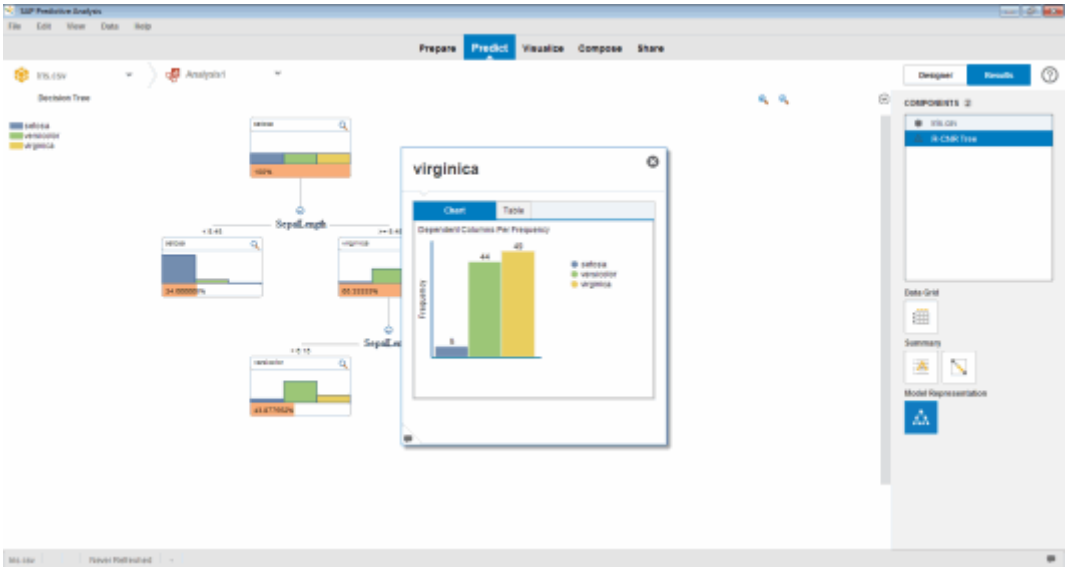
This presentation is used for decision tree analysis. In this technique, a binary decision tree is built by splitting observations into smaller sub-groups until the stopping criterion is met. The leaf node indicates classified data. You can enlarge the decision tree by choosing the zoom-in button.

> **Note**
>
> It is not possible to render a decision tree if there are more than 32 categorical values for a dependent column.

> **Note**
>
> The look and feel of the decision tree differs based on the algorithm vendor. For example, the decision tree for the R-CNR Tree algorithm is different from the decision tree for the HANA C4.5 algorithm.

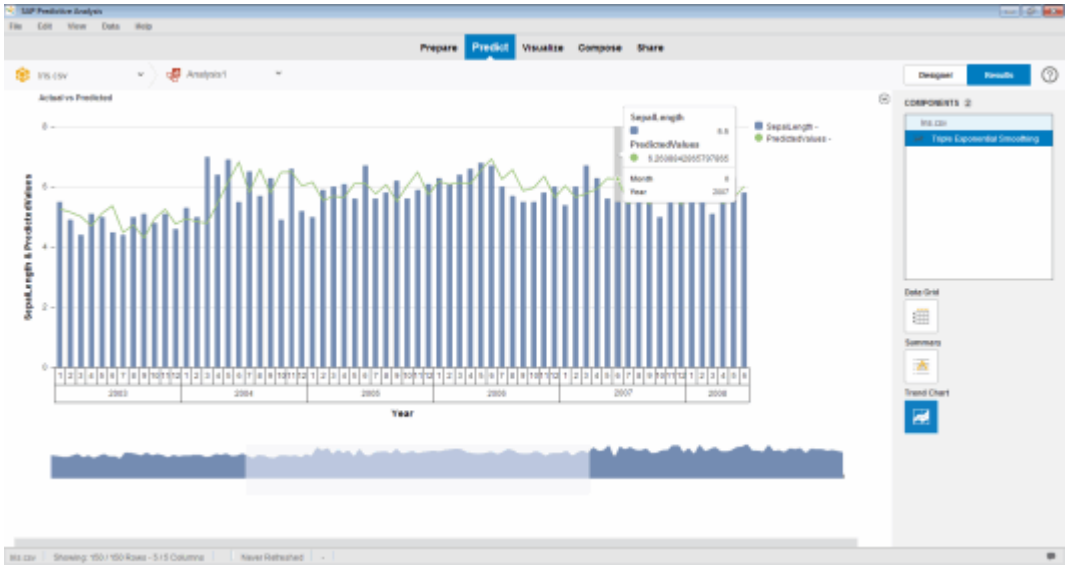


Each node in the decision tree represents the classification of data at that level. You can view node contents by choosing on each node.



8.6 Trend Chart

A trend chart is used to visualize the correlation between the dependent and independent variables. In the trend mode, you can analyze the performance of the algorithm by comparing the actual dependent variables with predicted values, where dependent variables are represented as a bar graph and predicted values are represented as a line graph. In the fill mode, the algorithm fills the missing values and displays the output as a line graph.



If the data set is very large, the graph may be unclear. For better visibility of data, use the Range selector located at the bottom of the graph to select a specific data range from the large data set. The data in the selected area is displayed in the visualization editor.

> **Note**
>
> In the Multiple Linear Regression (MLR) algorithm charts, the x-axis attribute is mentioned as *Record ID*.

8.7 Cluster Chart

A cluster graph is a visualization technique that uses different charts to represent cluster information such as cluster distribution, cluster density and distance, feature distribution, and cluster center representation.

Cluster Distribution

Cluster distribution represents the number of observations in each cluster and is represented by a horizontal bar chart. However, you can also visualize the cluster distribution in a pie chart or a vertical bar chart.

Cluster Density and Distance

The distance between clusters and density of each cluster is represented by a network chart. Each node in the network represents a cluster and its size. The color of the node represents density.

Feature Distribution

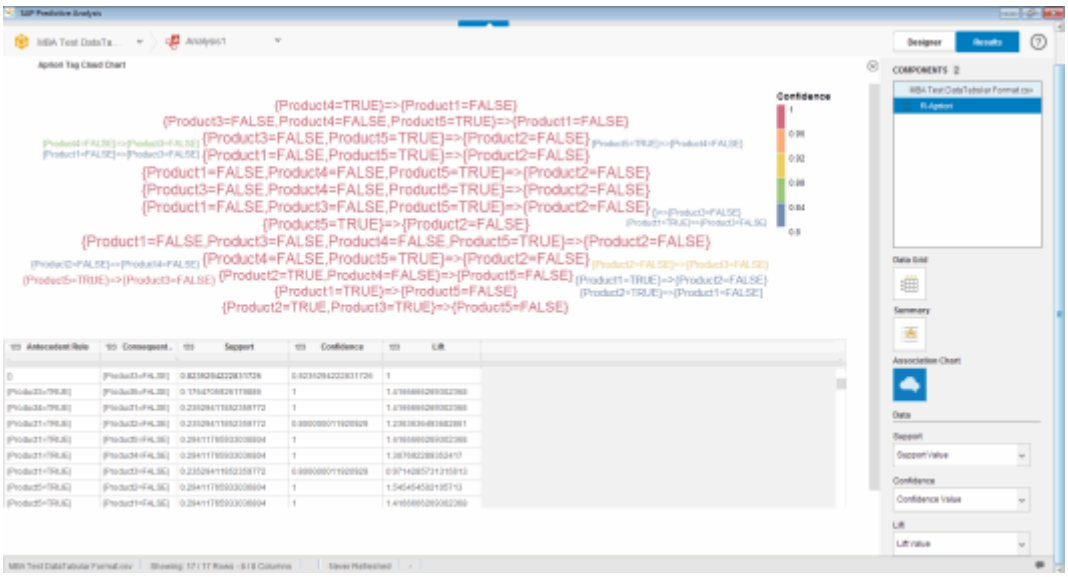
The comparison of the total distribution of all clusters against the distribution of each cluster is represented by a histogram. You can select the required measure from *Measures* under the *Data* section. You can view feature distribution for each cluster by selecting cluster number from *Clusters* under the *Data* section.

Cluster Center Representation

The R-K Means algorithm computes center points for each feature in each cluster. The comparison of each center point and cluster is represented by the radar chart. By default, the chart is displayed with normalized data. In the normalized mode, the data will be represented in the range of 0 to 1. However, you can unselect the *Normalize Result* option from *Settings*.

8.8 Apriori Tag Cloud Chart

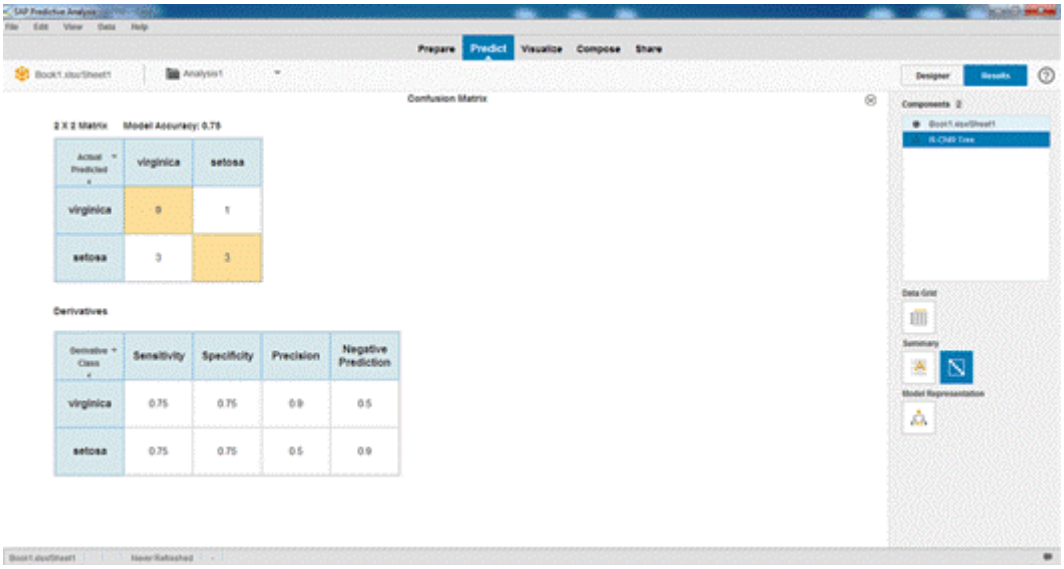
Apriori tag cloud chart enables you to visualize and find the frequent individual items, based on the association rule. In this visualization chart, the highly prominent rules are the strongest ones. The prominence of the rules varies as per the confidence and the lift value. Higher the confident value deeper is the color of rules and higher the lift value bigger is the font of rules. You can change the support, confidence, and lift values by adjusting the respective range sliders in the *Data* pane.



8.9 Confusion Matrix

Confusion matrix contains information about actual and predicted classification performed by an algorithm, which enables you to visualize the accuracy. You can view the chart by selecting the output method *Classification* and *Trend* for the CNR Tree algorithm. It is an n*n matrix (where n is the number of distinct values present in the dependent column selected for the algorithm), mapping the number of occurrences for each predicted value against the actual value. The entries on the diagonal of the matrix represents the correct prediction. The entries off the diagonal of the matrix represents the misclassification.

When you hover over a class, the true predicted value and the actual count of the data set are displayed. The derivatives table represents the efficiency (sensitivity, specificity, precision, negative prediction) of the algorithm. Using the *Settings* option, you can analyze the data in number, percentage, and both formats.



8.10 R Extensions Chart

You can display the results of a R Extensions analysis in chart format in both online and offline modes.

The chart type can vary depending on the data set and what is generated by the algorithm used in the R Extension.

> **Note**
>
> The *Show Visualization* checkbox must have been selected when creating the R Extension to display the analysis results in chart format. The R script for the component needs to contain calls to plot the data in the chart.

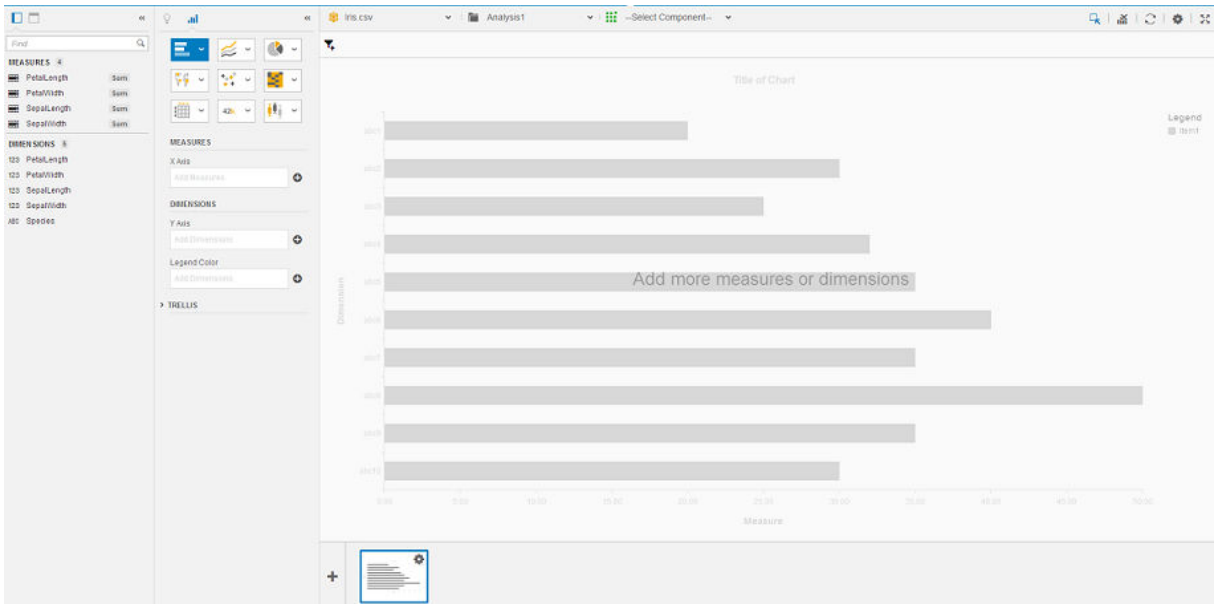
Related Information

Specifying properties with the R Extension Wizard [page 92]

9 Visualizing Data

9.1 Visualize room—creating charts

These are the main areas where you interact with the *Visualize* room:



Visualize Room (Example from English version)

Area	Description
Measures and Dimensions panel	Use this panel to view, sort, select, and filter the data in a visualization. Data is grouped into measures (for quantitative data) and dimensions (for categorical data). Measures and dimensions can be dragged directly to the Chart Canvas or to shelves in the Chart Builder. • In the *Horizontal Orientation* layout, the data associated with each dimension is displayed in a column above the Chart Canvas. You can search for specific data values within a dimension, select multiple values to include or exclude from a visualization, and view the measures associated with a dimension.

Area	Description
Chart Canvas	Use this area to create, modify, and explore a visualization. Build a chart by dragging measures and dimensions directly to the Chart Canvas or to shelves in the Chart Builder. Quickly adjust the content or appearance of a visualization with the toolbar buttons in the upper-right corner of the Chart Canvas: •  *Fit Chart to Frame*: For a bar, column, or line chart, activate this setting to display all of the datapoints on the screen at once. When it is deactivated, you can focus on a smaller set of members and use a scroll bar to navigate the data. •  *Sort*: Organizes chart data by measure. •  *Add or edit a ranking by measure*: Focuses a chart on a specified number of the highest or lowest dimension members. •  *Clear Chart*: Removes all dimensions and measures from a chart and any filters applied to the chart, not including data set filters. •  *Refresh*: Refreshes the chart data. •  *Settings*: Sets the chart properties. •  *Maximize*: Expands the Chart Canvas to full-screen mode.

Visualization Tools
 	
(The example shows the Chart Builder.)
	Use the tools at the top of this panel to switch between the *Chart Builder*  tab and the *Related Visualizations*  tab.
	
	• Use the *Chart Builder* tab to change the chart type and to customize a chart.
	• Use the *Related Visualizations* tab to choose predefined charts that were automatically generated from the measures and dimensions in the current data set.
	  ○ Add related visualizations to the current story and modify them.
	  ○ View all chart suggestions by selecting *Show All*.
	  ○ Remove measures or dimensions that were used to generate a visualization with the *Filter related visualizations*  icon. This refines the list of related visualizations that are available to you.

Area	Description
Chart Picker	Use to select the type of chart to use for a visualization.
Shelves	Use to add measures and dimensions to a visualization. When you drag a measure or dimension to a shelf, the Chart Canvas updates automatically.
Visualization Gallery	Use to create new visualizations and to select between visualizations in a story. • Create a visualization by selecting the *Create new visualization* + icon. • Remove or copy a visualization by selecting the *Settings*  icon. • Change the order of visualizations in the Visualization Gallery by dragging them to a different order.

Related Information

Creating charts [page 121]
Working with the Chart Builder [page 123]
Chart properties [page 125]

9.1.1 Creating charts

A story can contain one or more charts, and you can create charts on the Chart Canvas or by using the Chart Builder. All charts included in a story are accessible in the Visualization Gallery under the Chart Canvas.

9.1.1.1 Chart types

Some types of data are especially suited to a particular chart type.

Type of analysis	Description	Charts available
Comparison	Compares differences between values or shows a simple comparison of categorical divisions of measures. For example, use a bar chart to compare the differences in sales revenue between countries.	• Bar Chart • Column Chart • Column Chart with 2 Y-Axes • 3D Column Chart • Marimekko Chart • Radar Chart • Area Chart • Tag Cloud • Heat Map • Table • Crosstab
Percentage	Shows the percentage of parts in a whole or values as ratios to a whole. The legend shows the percentage and the total values. For example, use a pie chart to see who had the highest sales as part of a total sales value directly: Total sales = $200, Paul had 10% ($20), David had 65% ($130), and Susan had 25% ($50)	• Pie Chart • Donut Chart • Pie with Depth Chart • Stacked Column Chart • Tree • Funnel Chart
Correlation	Shows the relationship between values or compares multiple measure values. For example, you can view the correlation of two measures and understand the impact of the first measure on the second measure.	• Scatter Plot • Scatter Matrix Chart • Bubble Chart • Network Chart • Numeric Point • Tree

Type of analysis	Description	Charts available
Trend	Shows a trend in the data values (especially for dimensions that are time-based, such as Year) or the progression of your data and possible patterns. For example, you can use a line chart to view sales revenue trends of a product throughout a range of years.	• Line Chart • Line Chart with 2 Y-Axes • Combined Column Line Chart • Combined Column Line Chart with 2 Y-Axes • Waterfall Chart • Box Plot • Parallel Coordinates Chart
Geographic	Shows a map of the country object used in the analysis and can optionally show data for dimensions (sorted by the country on the map) or the geographical spread of data for a country. The data set you use must contain geographical data. Before creating a geographic chart, you must have an Esri ArcGis Online account.	• Geo Bubble Chart • Geo Choropleth Chart • Geo Pie Chart • Geo Map

9.1.1.2 Creating a chart directly on the Chart Canvas

You can quickly create a chart by dragging measures and dimensions to the Chart Canvas in the central area of the *Visualize* room.

A chart must have at least one measure. When you add a dimension to the chart, its values are calculated based on the chart's measures.

1. In the *Visualize* room, select the *Chart Builder*  icon.
2. Select a chart type from the lists in the Chart Builder.
   Bar Chart is the default chart type, but you can change the chart type.
3. Select a measure and drag it to an axis on the Chart Canvas.
   Text in the chart body guides you to the correct axis for the measure. A check mark appears when you drag the measure over an area where it can be dropped.
4. Select a dimension and drag it to the Chart Canvas.
   Text in the chart body guides you to the correct axis for the dimension. A check mark appears when you drag the dimension over an area where it can be dropped.
5. Add additional measures and dimensions as required.
   For example, if you selected *Column Chart 2 Y-Axes*, you must add a measure or dimension to the Y-Axis on the left side of the Chart Canvas and to the Y-Axis on the right side of the Chart Canvas.
6. To filter the data in the chart, select the *Add filters*  icon at the top of the Chart Canvas, and select a dimension to filter on.

7. Select + in the Visualization Gallery to add the chart to the story.

The chart is available in the Visualization Gallery and in the *Compose* room.

Each new chart that you create in the *Visualize* room is automatically saved in the current session and is available in the *Compose* room, but it is not saved in the story. To access the chart the next time you open the story, you must save the story.

Related Information

Filtering data in the Visualize room [page 133]
Saving a story [page 148]
Compose room—creating stories about visualizations [page 140]

9.1.1.3 Working with the Chart Builder

You can use the Chart Builder to change the chart type and to customize a chart.

The Chart Builder has different types of shelves (measures, dimensions, and trellis) for each chart type. Measures and dimensions can be dragged or added to shelves.

Measure shelf	Description
Area Color	The color used for each area in a map chart
Area Weight	The weighting to give to each area in a tree map
Axis	An X or Y axis of a bar and column chart, line chart, scatter chart, bubble chart, box plot, radar chart, or waterfall chart. Multiple axis shelves may be available. For example, when you select a *Bar Chart With 2 X-Axes*, the *X Axis1* and *X Axis2* shelves appear.
Bubble Width	The width of a bubble in a bubble chart
Column Width	The width of a column in a marimekko chart
Measures	The measures that are displayed in a crosstab. You can move the *Measures*  token to the *Rows* or *Columns* shelf to choose where the measures appear.

Measure shelf	Description
Pie Depth	The thickness of each section in a pie with depth chart
Pie Sectors	The sections in a pie chart
Primary Values	The primary values in a table, funnel chart, or parallel coordinates chart
Value	The primary value used in a funnel chart or the number displayed in a geographic chart or numeric point chart
Word Color	The color of text in a tag cloud
Word Weight	The weighting of text in a tag cloud

Dimension shelf	Description
Animation	Adds an animation to a scatter chart. When you select the *play* button below a chart, the chart cycles through the values of the dimension added to this shelf.
Area Name	The label used for each area in a map chart
Axis	An axis of a bar and column chart, line chart, box plot, or waterfall chart. Multiple axis shelves may be available. For example, if you select a *Bar Chart*, the *Y Axis* shelf appears.
Category	A section of data in a funnel chart or parallel coordinates chart
Color	The color of a data point in a geographic chart
Geography	A data point in a geographic chart
Legend Color	The colors in a legend for a bar and column chart, line chart, pie chart, scatter chart, box plot, or radar chart. The colors in a chart automatically update to match its legend.
Legend Shape	The shape of each entry in a legend and of each data point for a scatter chart or radar chart
Network Link	The nodes in a network chart
Overlay Data	An extra data overlay in a geographic pie chart. When multiple dimensions are added, pie charts are created on the geographic map.
Radar Branches	The quantitative variables represented on axes starting from the same point in a radar chart
Rows Axis	An axis of a row in a table
Tree Node	The nodes in a tree chart
Word	The text displayed in a tag cloud

Table or crosstab shelf	Description
Columns	The column axis of a crosstab. You can add dimensions to this shelf, and move the *Measures* token to this shelf to display measures on the columns.
Rows	The row axis of a crosstab. You can add dimensions to this shelf, and move the *Measures* token to this shelf to display measures on the rows.
Rows Subtotals	Adds a subtotal to the rows in a table or crosstab

A trellis chart is a set of small charts shown in a grid for comparison. Each small chart represents one item in a section. For example, if you create a bar chart that compares revenue by region, and then add the `<Country>` dimension to the trellis, multiple small charts appear. Each small chart displays the revenue by region for one country.

Trellis shelf	Description
Rows	The rows in a trellis chart. For example, if you place the `<Year>` dimension on the *Rows* shelf, the trellis chart will contain a row for each year in the `<Year>` dimension.
Columns	The columns in a trellis chart. For example, if you place the `<Year>` dimension on the *Columns* shelf, the trellis chart will contain a column for each year in the `<Year>` dimension.

Related Information

Creating a chart with the Chart Builder [page 126]
Adding or modifying a predefined chart [page 128]

9.1.1.3.1 Chart properties

Setting the properties for a chart can enhance its usability. For example, adding labels and legends can improve the visual analysis of data.

To set chart properties, select the *Settings* icon above the Chart Canvas.

Property	Description
Normal Stacking or *100% Stacking*	Applies to charts where each data point is divided into segments, such as stacked column charts and area charts. *Normal Stacking* allows you to compare the absolute values of data points and their segments. With *100% Stacking*, percentage values are displayed on the measures axis, allowing you to compare the proportional value of each segment across different data points.
Horizontal or *Vertical*	Switches the orientation of the chart between horizontal and vertical.
Show Title	Adds a title to the chart. You can edit the title at any time.
Show Legend	Adds a legend that shows a different color for each measure in a chart. To add dimensions to the legend in different colors, select *Legend Color* in the Chart Builder.
Choose Legend Item Colors...	Sets the colors that appear in the chart.
Show Data Labels	Displays measure values for each dimension in a chart.
Use Measures As a Dimension	Plots two or more measures as a dimension in a chart to show how data is spread over multiple measures on a single axis. You must add at least two measures to a chart before selecting this option. The measures appear as a new dimension in the Chart Builder.
Set Axis Scale	Defines the limits for values displayed on the Y-Axis, either as a range or automatically to the highest measure value. This option applies only to charts with measures on the Y-Axis.
Show Gridlines	Displays gridlines on the chart.

9.1.1.3.2 Creating a chart with the Chart Builder

Use the Chart Builder when you need more control over chart creation. (You can use the Chart Canvas for simpler charts.)

Action	Icon	Description
Move		Select the icon and drag a section to move the section.
Explore		Select the icon to explore a section.

Action	Icon	Description
Maximize		Select the icon to expand a section to fit the screen width, which gives you more space when designing a chart.
Close a section		Select the icon to close an expanded section.

1. In the *Visualize* room, select the *Chart Builder*  icon.
2. In the Chart Builder, select the chart type to create.

   Bar Chart is the default chart type, but you can change the chart type at any time.
3. Select an empty shelf in the Chart Builder, and select measures and dimensions in the list that appears. Or, drag a measure or dimension to an empty shelf.

   Each chart must have at least one measure. When you add a dimension to a chart, the dimension values are calculated based on the chart's measures.
4. Add additional measures and dimensions as required.
   For example, if you selected *Column Chart 2 Y-Axes*, you must add a measure or dimension to the Y-Axis on the left side of the Chart Canvas and to the Y-Axis that appears on the right side of the Chart Canvas.
5. To filter the data in the chart, select the *Add filters*  icon at the top of the Chart Canvas, and select the dimension to filter on.
6. Select + in the Visualization Gallery to add the chart to the story.
   The chart is available in the Visualization Gallery and the *Compose* room. Each new chart that you create in the *Visualize* room is automatically saved in the current session and is available in the *Compose* room. However, it is not automatically saved in the story.
7. Save the story.
   Saving the story ensures that the chart is available the next time you open the story.

Related Information

Filtering data in the Visualize room [page 133]
Saving a story [page 148]
Compose room—creating stories about visualizations [page 140]
Working with the Chart Builder [page 123]

9.1.1.3.3 Adding or modifying a predefined chart

The *Related Visualizations* tab contains predefined chart suggestions that are based on the measures and dimensions in a data set.

A chart must have at least one measure. When you add a dimension to a chart, the dimension values are calculated from the chart's measures.

You can select any chart on the *Related Visualizations* tab to immediately start visualizing data and then modify the chart for your information requirements.

1. In the *Visualize* room, open the *Related Visualizations* tab, select a predefined chart, and select the + icon in the Visualization Gallery to add the chart to the current story.

   This ensures that the chart is not replaced by a predefined chart later.

2. Select the *Related Visualizations*  icon.
3. In the list of chart suggestions, select *Show All* to display all chart suggestions.
4. Select the chart to add.
   The chart appears on the Chart Canvas and its measures and dimensions are loaded in the Chart Builder.
5. Use the Chart Builder to add or modify dimensions and measures:
   - To add measures or dimensions, select an empty shelf in the Chart Builder, and select measures and dimensions for your chart in the list that appears.
   - To add a measure or dimension to the chart, drag it to an empty shelf.
   - To remove a measure or dimension, position the pointer over it and select the *Remove*  icon. Or, drag a measure or dimension off a shelf.
6. To filter the data in the chart, select the *Add filters*  icon at the top of the Chart Canvas, and select a dimension to filter on.
7. Select the + icon in the Visualization Gallery to add the chart to the story.
   The chart is available in the Visualization Gallery and the *Compose* room. Each new chart that you create in the *Visualize* room is automatically saved in the current session and is available in the *Compose* room. However, it is not automatically saved in the story.
8. Save the story.
   Saving the story ensures that the chart is available the next time you open the story.

Related Information

Filtering data in the Visualize room [page 133]
Saving a story [page 148]
Compose room—creating stories about visualizations [page 140]
Working with the Chart Builder [page 123]

9.1.1.3.4 Adding a calculation

You can add one or more calculations to a visualization.

The following calculations are available:

- *Running Sum*
- *Running Minimum*
- *Running Maximum*
- *Running Count*
- *Running Count (Empty Values Excluded)*
- *Running Average*
- *Running Average (Empty Values Excluded)*
- *Moving Average*
- *Percentage*

1. In the Chart Builder, select the measure in the visualization to add a calculation to.
2. Select the *Options* icon, and select *Add Calculation*.
3. Select a calculation in the list.

The visualization is updated to include the calculation, and a measure containing the calculation appears in the Chart Builder.

9.1.1.3.5 Removing a calculation

1. In the Chart Builder, select the measure that contains the calculation to remove.
2. Select the *Remove* icon.

9.1.1.3.6 Renaming a chart

The title displayed above a chart is generated automatically from the measures and dimensions added to the chart.

Select the *Options* icon next to a chart title, select *Rename*, and enter a new title.

> **Tip**
>
> You can double-click a chart title to quickly change it.

9.1.1.3.6.1 Restoring a chart's default title

After a chart's title has been changed, if necessary, you can restore the original title that was generated automatically from the measures and dimensions in the chart.

Select the *Options*  icon beside a chart title, and select *Restore Default Title*.

9.1.1.4 Analyzing data with tables and crosstabs

Crosstabs and tables show data points only as values, rather than providing a visual representation of those values. As a result, they are useful when your analysis depends on viewing exact values, or examining data from multiple measures with different scales or units of measurement.

In addition to regular sorting and ranking functionality, you can also use conditional formatting in tables and crosstabs to help identify noteworthy data points.

Tables

With a table, you can add multiple measures, which are displayed on the columns, and multiple dimensions, which appear on the rows. For example, a table could be an effective way of examining several measures related to the sales performance for a list of products. You might add a Product Category dimension so that you can display the totals for each category on the rows.

Crosstabs

For more flexible data analysis, you can use a crosstab. You add multiple measures to the *Measures* shelf, and switch the display of the measures between the columns and rows by moving the *Measures*  token. Dimensions can be added to the rows, columns, or both, allowing complex multidimensional analysis.

For example, adding a Year dimension to the rows of your sales analysis in a table might make it difficult to compare data across both time and product type. Instead, you could create a crosstab with the measures and Year dimension on the columns and the Category and Product dimensions on the rows, making it easier to spot relationships between the dimensions.

> **Note**
>
> You can sort a crosstab by a measure, however, the sort is removed if a dimension is added to the same axis as the measures.

9.1.1.4.1 Conditional formatting

Conditional formatting can highlight important data points in a table or crosstab and be used to distinguish values that meet a condition (such as being greater than a certain number or within a specific range).

You can define multiple conditional formatting rules and manage them in the *Rules Manager* dialog.

Multiple conditional formatting rules

- When you create multiple conditional formatting rules based on the same measure, cells may meet the condition for multiple rules. When this happens, all rules that apply to a cell (that is, active rules) are considered a set. Formatting for the set (of all active rules) will be applied or no formatting will be applied, depending on the rule priorities.
- For each cell in a table, the formatting set for the highest-priority active rule is applied first. Formatting for lower priority rules can also be applied. However, if two formatting sets for active rules that modify the same attribute conflict, none of the formatting defined for the lower priority rule is applied to the cell.
- For each cell in a table, bold and italic formatting can be applied only by the highest priority active rule.

Example

In a table with a measure that shows inventory shrinkage at your company's retail outlets, you could use conditional formatting to identify stores with high rates of shrinkage. A conditional formatting rule could change the cell background color in a shrinkage column to red for each store with shrinkage higher than an amount you specify.

Example

A cell meets the conditions for three conditional formatting rules. The highest-priority active rule sets the font to Times New Roman. The rule with the second highest priority sets the background color to red. A final rule would set the background color to black and the font color to white, but that rule is ignored because it conflicts with the second rule.

9.1.1.4.1.1 Creating a conditional formatting rule

By default, new conditional formatting rules have higher priority than older rules.

Before you can define a conditional formatting rule, a table must have a measure added to it.

1. Select the *Create new conditional formatting rule*  icon.
2. In the *Rule Editor* dialog, enter a name for the rule.
   If you do not enter a name, the rule will be named automatically, based on the condition that you set.
3. In the *Based On* list, select a measure.
   This measure determines the values that are used in the rule and the column where formatting appears.

4. Select an operator, and enter one or more values for the condition.
5. Select *Format*, choose the appearance of cells that meet the condition, and select *OK*.
6. In the *Rule Editor* dialog, select *OK*.

The conditional formatting rule is applied to the table.

If needed, you can use the *Rules Manager* dialog to change the priority of rules.

9.1.1.4.1.2 Managing conditional formatting rules

Use the *Rules Manager* dialog to edit, add or remove, turn on or off, and set the priority order of rules.

Before you can manage conditional formatting rules, a table must have a measure added to it.

1. Select the arrow beside the *Create new conditional formatting rule*  icon, and select *Manage Rules*.
2. In the *Rules Manager* dialog, perform any of these actions:

Option	Description
To create a rule	Select the + icon.
To delete a rule	Select the - icon.
To modify a rule	Select a rule and select *Edit Rule*.
To disable a rule	Clear the check box in the *Applied* column next to the rule name. Disabled rules are not applied to the table, but you can turn them on again if necessary.
To change the priority of a rule	Select a rule and use the *Change Rule Order* icons to move it higher or lower in the list.

3. Select *OK*.

9.1.2 Data sorting in charts

You can sort measures and dimensions in charts in ascending or descending order.

9.1.2.1 Sorting by measure

Before you can sort by measure, if chart data is filtered by rank, the rank must be removed.

1. Select a measure on the Chart Builder.
2. Select the *Settings*  icon, and select *Sort Ascending* or *Sort Descending*.

> Tip
>
> Select the *Sort*  icon on the Chart Canvas toolbar to quickly change the sort order.

The chart data is sorted.

9.1.2.2 Sorting dimensions

When the *Measures and Dimensions* panel is displayed in a horizontal orientation, you can sort dimensions that are visible in the panel. Sorting dimensions does not affect the data displayed in a visualization.

1. Select the *Horizontal Orientation*  icon on the *Measures and Dimensions* panel.
2. Select the dimension to sort, and select the *Options*  icon.
3. Choose a sort order:
   - For a numeric dimension, select *Sort Lowest to Highest* or *Sort Highest to Lowest*.
   - For an alphanumeric dimension, select *Sort A to Z* or *Sort Z to A*.
   - For a date or time dimension, select *Show Earliest to Latest* or *Show Latest to Earliest*.

The data in the dimension column is sorted.

9.1.2.3 Sorting dimensions by occurrence on the Measures and Dimensions panel

You can sort dimensions visible in the *Measures and Dimensions* panel by the number of times each dimension value occurs in a data set.

Sorting dimensions does not affect the data displayed in a visualization.

1. Select the *Horizontal Orientation*  icon on the *Measures and Dimensions* panel.
2. Display the number of occurrences:
   a. Position the pointer over the dimension to filter.
   b. Select the *Options*  icon, and select *Show Measure* and *Occurrences*.

   The number of occurrences appears beside each dimension value in the column.
3. Sort by occurrence:
   a. Position the pointer over the dimension that you selected in step 2.
   b. Select the *Options*  icon, and select *Sort by Measure Lowest to Highest* or *Sort by Measure Highest to Lowest*.

Data in the dimension column is sorted by occurrence.

9.1.3 Filtering data in the Visualize room

You can filter data in the *Visualize* room in the following ways:

- By selecting the *Add filters*  icon
- By selecting data points in a chart to filter or exclude them
- By selecting the data to display on the *Measures and Dimensions* panel

As well, you can use the ranking by measure feature to filter data by measure.

Related Information

Filtering data by rank [page 135]

9.1.3.1 Using the filter dialog in the Visualize room

1. On the filter dialog, choose an operator from the list.
2. Select or type the values to filter:
   - For filters that use the *Between* operator, type a beginning value and an end value.
   - For filters that use the *In List* or *Not In List* operator, select values from the list in the filter dialog.

   > **Note**
   >
   > You can hold SHIFT while clicking values to select a range of values.
   >
   > You can also select the *Options*  icon to change the filter dialog settings, including displaying the number of times that each record occurs in the data set, and sorting the data by value or by number of occurrences.
   >
   > When filtering an alphanumeric dimension, you can select the *Find*  icon to search for a member by name.
3. Select *Apply*.

The data is filtered and a token representing the filter is added above the Chart Canvas.

9.1.3.2 Filtering or excluding data points in a chart

You can exclude non-relevant data points or filter data points to focus a chart on a specific set of data.

1. On the Chart Canvas, select the data points to exclude or filter.

   > **Tip**
   >
   > You can drag a box around a group of data points to select the group.

2. In the tooltip that appears, select *Filter* or *Exclude*.

The data in the chart is filtered and a token representing the filter is added above the Chart Canvas.

9.1.3.3 Filtering data with the Measures and Dimensions panel

1. Select the *Horizontal Orientation*  icon to display the *Measures and Dimensions* panel in a horizontal layout.
2. On the *Measures and Dimensions* panel, select one or more data points in the dimension to filter.
3. Select the *Options*  icon.
4. Depending on the kind of filter to apply, select one of the following options:

Option	Description
Clear Selections	Clears all values selected in the dimension
Include	Includes selected values in the chart. A filter token with the selected values appears on the filter bar.
Exclude	Excludes selected values from the chart. A filter token with the selected values in a strike-through font appears on the filter bar.

The data in the chart is filtered and a token representing the filter is added above the Chart Canvas.

9.1.3.4 Filtering data by rank

Filtering data by rank focuses a visualization on a specified number of data points with the highest or lowest values.

1. On the Chart Canvas toolbar, select the *Add or edit a ranking by measure*  icon.
2. In the *Ranking* dialog, select the measure to rank.
3. Select *Top* or *Bottom* as the focus of the ranking.
4. Choose the number of results to display.

   The default number is three.
5. Select *(ALL)* to rank data based on all dimensions, or select the dimension to rank data on.

   For example, if a chart shows Sales Revenue by Country and Product Line, ranking the top five data points by Country shows data for each product line in the five countries with the highest sales revenue.
6. Select *OK*.

The data is filtered by rank and a token representing the filter is added above the Chart Canvas. Only one ranking can be applied to a visualization at a time.

9.1.4 Hierarchical data

The *Dimension Hierarchy* icon indicates that a hierarchy is associated with a dimension. There are multiple ways you can find and interact with hierarchical data.

9.1.4.1 Finding dimensions in a hierarchy

Hierarchical relationships between dimensions are visible on the *Measures and Dimensions* panel.

Only the dimension containing the highest level of a hierarchy appears on the *Measures and Dimensions* panel, but you can expand the dimension to see additional levels.

You can add a dimension at any level of the hierarchy to a chart.

Perform one of the following actions:

- If the *Measures and Dimensions* panel is in the vertical orientation, select the + icon beside a dimension to display all dimensions in the hierarchy.
- If the *Measures and Dimensions* is in the horizontal orientation, look for dimensions displayed beside each other in the hierarchy.

9.1.4.2 Choosing the level of hierarchy displayed in the Chart Builder

If a dimension containing a hierarchy is included in a chart, the level displayed in the chart can be changed in the Chart Builder.

1. Select a dimension that contains a hierarchy.
2. Select the *Settings* icon and choose a level in the hierarchy.

The chart displays data from the selected level.

9.1.4.3 Drilling through hierarchical data

If hierarchical dimensions are included in a chart, you can drill up or down through dimensions on the Chart Canvas to explore the data at different levels. If the chart contains more than one hierarchical dimension, you can select which dimension to drill into. You can use the *drill back* icon to undo the drill operation and restore the chart to its original state.

The drill operation comprises:

- applying a filter
- redrawing the visualization at the new level in the hierarchy

When you drill, a filter token may appear above the chart, or the filter may be added to an existing filter token.

1. Select an area in the chart or a label on the axis.

   For example, you can select one or more bars in a bar chart, or an axis label in a trellis.

   The selected area in the chart is highlighted.

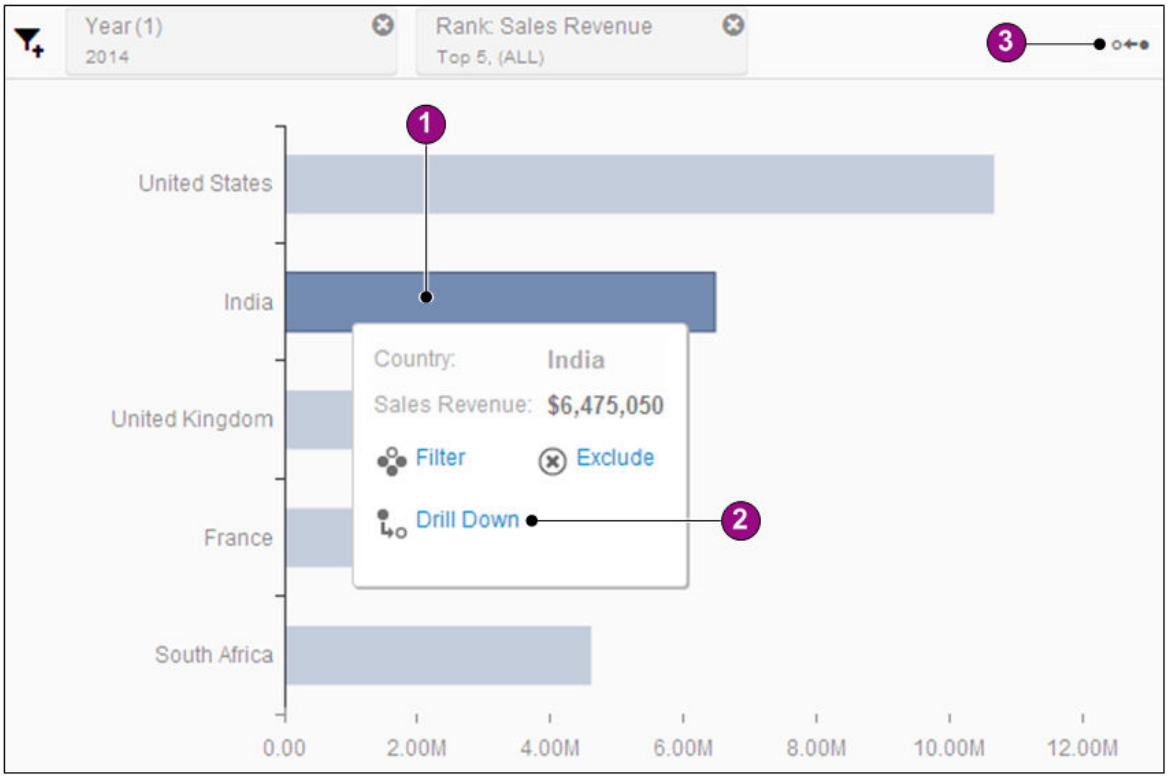
2. In the tooltip that appears, select the *drill down* or *drill up* icon.

   If the area you selected contains more than one hierarchical dimension, you can choose which dimension to drill into.

   A filter is applied to the data and the chart is re-drawn at the new level in the hierarchy.

3. To step back through the drill operation, select the *drill back* icon.

   The filter created by the drill operation is removed and the visualization is re-drawn at the previous level. Any filters applied by hand are maintained. Note that the *drill back* history is reset when you switch to the *Visualize* room.



Related Information

Drilling through hierarchical data in a story [page 149]

9.1.5 Finding measures, dimensions, and data values

You can search text and integer dimension values for the name of a measure or dimension.

The *find* icon is located on the *Measures and Dimensions* panel

- When the panel is in a vertical orientation, you can use the *find* icon to search for measures and dimensions by name.
- When the panel is in a horizontal orientation, the *find* icon becomes available when the pointer is positioned inside a column, and you can use it to search each dimension for specific values.

Operator	Description
*	Matches any character zero or more times. For example, entering **a*a** matches any word containing the letter "a" followed by any combination of letters, followed by another "a."
?	Matches any character one time. For example, entering **a?a** matches any word containing the letter "a" followed by any single letter, followed by another "a."

If a dimension contains mapped labels, select the *Options* icon, and select *Find by Key* or *Find by Label*.

Restriction

Date, time, time stamp, and non-integer numeric dimensions cannot be searched.

Restriction

Literal * and ? characters cannot be used in search text or values.

9.1.6 Measures associated with dimensions

You must display the *Measures and Dimensions* panel in the horizontal orientation to view the measure values associated with a dimension. You can also view the number of times each dimension value occurs in a data set.

Example

Suppose a data set contains a measure called “Number of Games Won” (calculated as a sum) and a dimension called “Name of Team”. You can display the total number of games that each team won beside each team name on the *Measures and Dimensions* panel.

9.1.6.1 Viewing a measure associated with a dimension

1. Select the *Horizontal Orientation*  icon to display the *Measures and Dimensions* panel in a horizontal layout.
2. Position the pointer over a dimension, and select the *Options*  icon next to the dimension.
3. Select *Show Measure*, and select the measure to view.

A measure value appears beside each value in the dimension column.

9.1.6.2 Viewing the number of occurrences of dimension values

You can view the number of times each dimension appears in your data set.

1. Select the *Horizontal Orientation*  icon to display the *Measures and Dimensions* panel in a horizontal layout.
2. Position the pointer over the dimension, and select the *Options*  icon next to the dimension name.
3. Select *Show Measure* > *Occurrences*.

The number of occurrences appears beside each dimension value in the column.

Related Information

Sorting dimensions by occurrence on the Measures and Dimensions panel [page 133]

9.1.7 Aggregation types supported

SAP Lumira supports the sum, count, minimum, and maximum aggregation types. You cannot change the aggregation type of a measure. However, you can add measures representing more complex calculations, currency values, and other units of measure.

10 Creating Stories

10.1 Compose room—creating stories about visualizations

A story is a presentation-style document that uses visualizations, text blocks, pictures, graphics, and input controls to describe data.

A story can include multiple pages, and each page can have its own layout—a board, infographic, or report. Step one is choosing a layout for the first page of the story.

10.1.1 Page Settings panel

After choosing the page layout for a story, you can format its pages in the *Compose* room.

Use the *Compose* room to create and edit presentation-style documents known as stories. Stories use visualizations, text blocks, pictures and graphics, and input controls to describe your data. They can include multiple pages, and each page can be a board, infographic, or report.

Page layout	Page elements available
Infographic	*PAGE SETTINGS* • *Size*: Select *Standard (4:3)*, *Widescreen (16:9)*, or *Continuous Scrolling*. • *Background Color*: Select a background color for the infographic page. • *Grid Properties*: Select the *Show* check box to display grid lines on the infographic page. • *Refresh page*: Select to refresh visualizations on the infographic page. • *Refresh page on open*: Select *ON* to refresh the infographic page or *OFF* to avoid refreshing the page when it opens.
Board	*PAGE SETTINGS* • *Board Title*: Select the *Show Title* check box to display the board page title. • *Background Color*: Select a background color for the board page. • *Background Image*: Add a background image to the board page.
Report	*BACKGROUND COLOR*: Select a background color for the report page.

Formatting	Description
Background color	Select a background color for this section of the page.

Action	Description
Order Items	Choose where the item is positioned in relation to other items: *Send Backward*, *Send to Back*, *Bring Forward*, or *Bring to Front*.
Alignment	Choose how the item is aligned: *Align Left*, *Align Center*, *Align Right*, *Align Top*, *Align Middle*, or *Align Bottom*.
Other Actions	Select *Duplicate* to copy the item or *Expand* to expand the item to the window size.
Size and Position	Select the width, height, X-axis position, Y-axis position, and angle of the item.

Page element	Formatting element

Visualizations	*VISUALIZATION PROPERTIES*
	• *General*:
	  ○ *Show Chart Title*: Select to display the chart title and format it.
	  ○ *Show Legend*: Select to display the chart legend and format it.
	  ○ *Show Data Labels*: Select to display the values for each dimension in a chart.
	  ○ *Format Data Labels*: Select to format the values for each dimension in a chart.
	• *X Axis* and *Y Axis*:
	  ○ *Show Axis*: Select to display the axis.
	  ○ *Show Axis Title*: Select to display the axis title and to format it.
	  ○ *Show Axis Labels*: Select to display axis labels and to format them.
	• *Bar*: Select a bar shape or pictogram to display as the bars in a bar chart, and choose the color of the bars.
	• *Column*: Select a column shape or pictogram to display as the columns in a column chart, and choose the color of the columns.
	• Line chart elements:
	  ○ *Chart Area*: Select the background color of the chart area.
	  ○ *Chart Title*: Display the chart title and format it.
	  ○ *Plot Area*: Select the background color of the plot area.
	  ○ *Legend*: Display a chart legend and to display a legend title and format it.
	  ○ *Data Label*: Display data labels or data-label pictograms.
	  ○ *Horizontal Axis*: Display the axis line and ticker, display axis labels and format them, and display axis pictograms.
	  ○ *Horizontal Axis Title*: Display the axis title and format it.
	  ○ *Vertical Axis*: Display the axis line and ticker, display axis labels and format them, and adjust the axis value scale.
	  ○ *Vertical Axis Title*: Display the axis title and format it.
	  ○ *Marker*: Select and format a pictogram to represent data points.
	  ○ *Line*: Set the line color, thickness, and style.
	  ○ *Plot Area*: Show or hide grid lines.
	• Donut chart elements:
	  ○ *Chart Area*: Change the size of the inner circle in the donut.
	  ○ *Chart Title*: Display the chart title and format it.
	  ○ *Plot Area*: Change the background color of the plot area.
	  ○ *Legend*: Display a chart legend and to display a legend title and format it.
	  ○ *Slice*: Change the color of a slice of the donut (to draw attention to that data point).
	  ○ *Data Label*: Select the *Show Data Labels* check box to display the data labels and format them.
	• Crosstab chart elements:
	  ○ *Expand Crosstab to See All rows*: Expand the crosstab vertically to see all the rows it contains. The page size is changed to continuous scrolling when this option is selected.

Page element	Formatting element
Text	*TEXT PROPERTIES* • *Font*: Select the font style, size, and color for text. • *Background Color*: Select the background color for text. • *Alignment*: Select the paragraph justification for text. • *Lists*: Add bulleted or numbered lists to text. • *Hyperlink*: Add, edit, or remove a hyperlink. • *Dynamic Text*: Add, edit, or remove dynamic text.

Pictures	*IMAGE PROPERTIES*
	• *Display Mode*: Select how to handle image scaling.
	  ○ *Contain*: The entire image is contained in the frame, maintaining the image's aspect ratio.
	  ○ *Cover*: The image is scaled to cover or fill the entire frame, maintaining the image's aspect ratio. Some parts of the image may be cropped.
	  ○ *Stretched*: The entire image is stretched to fit in the entire frame.
	  ○ *Pan*: The image is scaled to fill the horizontal dimension of the frame. The bottom of the image may be cropped.
	• *Background Color*: Select a background color for the picture.
	• *Image Actions*: Add, edit, or remove a hyperlink.

Input Controls	*SELECTION MODE:* For a dimension in a visualization, select *Single* to show one value or *Multi* to show multiple values.
Pictograms	*PICTOGRAM PROPERTIES* • *Fill Color*: Select the fill color for the pictogram. • *Line Color*: Select the line color for the pictogram. • *Pictogram Actions*: Add, edit, or remove a hyperlink. **Note** *Fill Color* and *Line Color* properties are not available for custom pictograms that you added to the application.
Shapes	*SHAPE PROPERTIES* • *Fill Color*: Select the fill color for the shape. • *Line Color*: Select the line color for the shape. • *Line Width*: Select the width of lines (in pixels) for a shape. • *Shape Actions*: Add, edit, or remove a hyperlink. **Note** *Fill Color*, *Line Color*, and *Line Width* properties are not available for custom shapes that you added to the application.

10.1.2 Creating a story

You can use board, infographic, and report pages to create stories about data. Stories can contain multiple pages, and each page is divided into sections that you can resize, reposition, or delete.

When you add a chart to a board or report page in the *Compose* room and then modify the chart data in the *Visualize* room, the chart updates automatically. However, charts added to infographic pages are not affected by changes made later in the *Visualize* room.

1. Drag elements from the *Content Panel* to the page and arrange the elements on the page.
    - To reposition an element, select the *Move*  icon in the upper-right corner of the element, and drag the element.
    - To resize an element, select the element, and drag the bounding box around the element.
      When a page includes a table, you can use the page scroll-bar to see all of the elements in the table.
2. To filter data on board or report pages, drag a dimension from the *Content Panel* to the page, and select the values to filter on.
   Charts are updated with the values applied by the filter.
3. To create additional pages, select *Add Page*, and repeat steps 2 to 5.
4. Save the story.

   If you don't save the story and close the browser, changes made to the story will be lost.

10.1.2.1 Formatting a story

A story includes one or more pages, and each page can include one or more sections.

You can format the general appearance of each page and of each element used on the page with color, text formatting, paragraph alignment, chart titles, axis labels, legends, and shape and line formatting.

1. In the *Compose* room, open the page of the story to format.
   The *Page Settings* panel displays the options available for this page layout.
2. Select page formatting options as needed.
3. Select an element on the page.
   The *Page Settings* panel displays the options available for this element.
4. Select element formatting options as needed.
5. Save the story.

10.1.2.2 Pictograms and shapes

Shapes and pictograms can add visual flair to your story.

You can insert them in two ways:

- As a separate element
  In the *Compose* room, drag a pictogram or shape from the *Content Panel* to a report or infographic page.
  The graphic can then be formatted using the *Board Settings* panel.

- As part of a visualization
  On infographic pages, you can use pictograms to represent certain chart elements, such as columns, bars, data point markers, data labels, and axes. These display options are available on the *VISUALIZATION PROPERTIES* panel when you select the element or group of elements. You can use the same pictogram for each member or select individual members to customize the appearance of each one.

10.1.2.2.1 Uploading custom pictograms and shapes

Before you can add your own vector graphics to stories, you must upload the graphics to the application.

1. In the *Compose* room, select *Pictograms* or *Shapes* on the *Content Panel*.
2. Select the + icon and select *Add from Local*.
3. Choose the vector graphics file to add, and select *Open*.

   The file must be an SVG file with valid XML encoding.

The graphic appears in the *Personal* section of the *Content Panel* for *Pictograms* or *Shapes*. You can add the graphic to infographic or report pages. You can add custom pictograms as part of a visualization on an infographic page, as well.

> **Note**
>
> - Changing the line color, fill color, or line width of custom shapes and pictograms is not supported.

10.1.2.3 Adding text to a visualization

In all page layouts, you can annotate visualizations with simple text, titles, and notes.

When an infographic has multiple elements (visualizations, pictures, pictograms, shapes, and filtered data), adding annotations to page elements can reinforce the intended data message.

1. In the *Compose* room, select the visualization to add text to.
2. Select *Text* on the *Content Panel*, and drag the *Simple Text*, *Title*, or *Note* box from the panel to the visualization.

   A blue bounding box shows the position of the text box in the visualization.
3. Enter the text, title, or note in the box.
4. (Optional) To move the text box, drag the bounding box to a new location.
5. (Optional) To resize the text box, select an anchor on the bounding box, and drag it to the desired size.
6. (Optional) To format the text, use the options under *TEXT PROPERTIES* on the *Board Settings* panel.

10.1.2.4 Adding dynamic text to a story

In all page layouts, you can add dynamic text that is based on the measures in the data set. Dynamic text is updated when the data set is refreshed.

1. In the *Compose* room, select the page to add dynamic text to.
2. Select the text element and position the pointer where you want to insert dynamic text. You can also highlight existing text you want to change to dynamic text.
3. On the *TEXT PROPERTIES* panel, select the *Add or Edit Dynamic Text*  icon.
4. In the *New Formula* dialog, enter a name for the formula.
5. Double-click the measures and functions you want to add to the *Formula* syntax box.

   You cannot create dynamic text objects based on dimensions.
6. Enter the parameters for the function and associated information, based on the function task.

   You must enter the names of columns used in the formula. After you enter the first letter, if the application can match an existing name to the letter, it displays the name.
7. If you are inputting calendar information, select the *Select a Date* button at the bottom of the functions list to use the date picker.
8. Select *OK* to apply the formula.

   You cannot add both dynamic text and a hyperlink to the same text.

The dynamic text element is added to the text object and will be updated each time the data set is refreshed.

10.1.2.4.1 Modifying dynamic text in a story

You can modify dynamic text in a story.

1. In the *Compose* room, select the page to edit.
2. Select the dynamic text to edit.
3. In the *TEXT PROPERTIES* panel, select the *Add or Edit Dynamic Text*  icon.
   The *Edit Formula* dialog appears.
4. Modify the text in the *Formula* box or change other options as needed, and select *OK*.

10.1.2.4.2 Removing dynamic text from a story

You can remove dynamic text from a story.

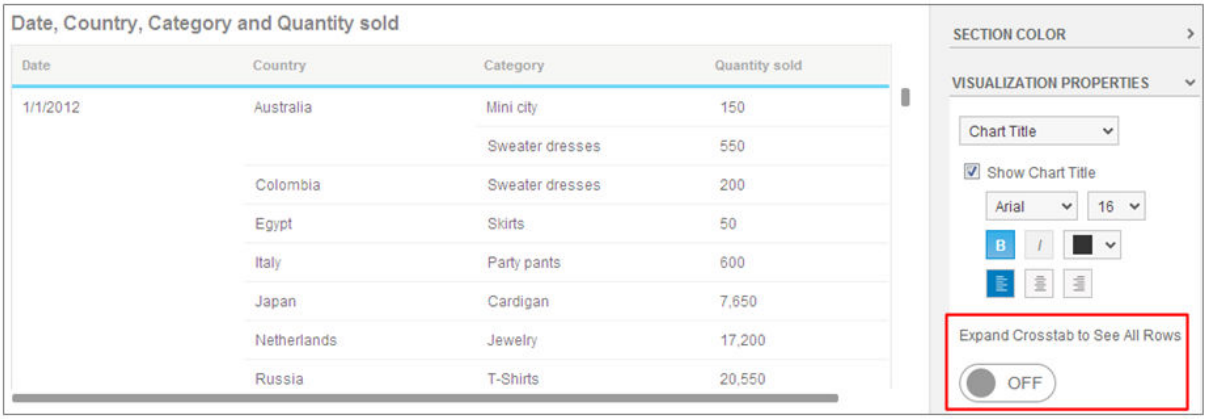
1. In the *Compose* room, select the page to remove dynamic text from.
2. Select the dynamic text to remove.
3. On the *TEXT PROPERTIES* panel, select the *Remove Dynamic Text*  icon.

10.1.2.5 Working with crosstabs in stories

Crosstabs are used in stories to display all the original data from a data set. By default, a crosstab is sized to fit into its container, but you can use *Expand Crosstab to See All Rows* to show all of the data on a single page. This setting changes the page to continuous scrolling mode.

When you select *Expand Crosstab to See All Rows*, the page's horizontal dimensions remain the same, and if the width of the crosstab is greater than the width of the page, you can use the scroll bars to view all columns.

This feature is available for Infographic pages.



Using pictograms and shapes with crosstabs

You can place pictograms and shapes on your visualization. You can place these items on a crosstab if *Expand Crosstab to See All Rows* is not selected. If *Expand Crosstab to See All Rows* is selected, these items cannot be placed on the crosstab.

Limits to the amount of data available in a crosstab

The amount of data you can retrieve from a data source is customized by your administrator. This may result in the crosstab displaying fewer rows than are available in original data source.

Exporting to PDF

Stories that contain crosstabs can be exported to PDF. If *Expand Crosstab to See All Rows* is selected, only the first 100 rows are exported.

10.1.3 Modifying a story

Select *Edit* in the upper-right corner of the window.

The story opens in the *Compose* room, where you can modify it.

Related Information

Compose room—creating stories about visualizations [page 140]

10.1.4 Saving a story

You can save a story that you own or have been granted permission to edit.

To save or make a copy of an existing story, use the *Save As* option.

10.1.5 Refreshing data on an infographic page

Data on infographic pages is not automatically refreshed, but you can optionally refresh it once or each time the page is opened.

Refreshing is helpful for real-time data, when you need the most current information in an infographic. However, refreshing data can change the narrative message of an infographic because it changes the data that the infographic is built on.

1. In the *Compose* room, open the infographic page to refresh data for.
2. On the *PAGE SETTINGS* panel, perform one of the following actions:
   - To refresh data on the page now, select the *Refresh visualizations on page*  icon.
   - To automatically refresh data each time you open the page under *Refresh page on open*, select the *ON* button.

   A dialog appears, indicating that visualizations will be updated to use the most recent data, which may change existing customizations.

10.1.6 Exploring a visualization in a story

You can explore visualizations on board pages. For example, you can drill down and up, filter values, and add rankings.

You can explore a visualization in many ways, such as:

- Drill down on a value and drill up
- Filter one or more values
- Add, change, or remove a ranking
- Zoom and pan on geographic charts
- Change the title

1. In the *Visualize* room, select the *Explore*  icon in the upper-right corner of a visualization.
   The visualization opens in a new window.
2. Explore the visualization and make changes as needed.

   You can select the *Fit Chart to Frame*  icon to expand the visualization to the size of the window.
3. Select *Update* to save your changes.

10.1.7 Drilling through hierarchical data in a story

In stories with a board layout, drilling through hierarchical data has the same capabilities as drilling in the *Visualization* room. If the story contains more than one visualization with the same hierarchical dimension, all visualizations in the story are updated. Input controls and filters applied to the story are maintained during the drill and drill back operations.

The drill operation consists of:

- applying a filter
- redrawing the visualization at the new level of the hierarchy

When you drill through one visualization in a story, the filter is applied to all visualizations that contain the same hierarchical dimension. The selected visualization, and any other instances of that visualization in the story, are redrawn at the new level in the hierarchy. Other visualizations remain drawn at the previous level.

The updates from drilling are applied to the visualization in the *Compose* room.

> **Note**
>
> Drilling is only available in stories with a board layout.

1. Select an area in a visualization to drill through.
2. In the tooltip that appears, select the *drill down*  or *drill up*  icon.

   The filter is applied to every visualization in the story, and the selected visualization is re-drawn at the new level in the hierarchy.
3. To step back through the drill operation, select the *drill back*  icon.

   The filter created by the drill action is removed from all visualizations in the story. The selected visualization is re-drawn at the previous level. Any filters or input controls applied by hand are maintained. Note that the *drill back* history is reset when you switch to the *Compose* room.

Related Information

Drilling through hierarchical data [page 136]

11 Optimizing Data

11.1 Optimizing Data

As an expert user, you can create a HANA Optimization Function. The function can be thought of as a powerful calculator that enables you to solve complex optimization functions. Using the functionality, you create an objective function with linear constraints to calculate how best to optimize an aspect of your business. An optimization example is to maximize profits on a product or store location. You can also save the optimization function for future use. Currently, the function is unavailable for use in predictive modeling chains.

To configure the HANA Optimization Function, ensure first that you are connected to an SAP HANA data source.

1. In the *Predict* room, you open a new optimization function in the list of components in the right-hand panel by selecting  *Optimization Function*.

   > **Note**
   >
   > To edit an existing function, go to *Predict - Optimizations - + (plus sign) - Functions - [Function Name].*

2. On the *Objective Function* tabbed page of the HANA Optimization Function, enter the following information:
   a. Enter a unique *Function Name*.
   b. Optionally, enter a *Function Description*.
   c. Select a *Function Optimization Type*, which can be *Maximize* or *Minimize*. For example, your function might be used to maximize profits or minimize costs.
   d. In the *Function* text box, enter an objective, linear optimization function in the format, *5x + 8y + (-4)z*. Note that you must include plus (+) signs as separators between monomes (e.g. *5x*). Monomes consist of coefficients and variables.

      > **Note**
      >
      > To maximize the screen, click the Expand icon .

   e. Click *Validate Function*. The validation ensures that your function is a mathematical expression that is supported by the underlying Optimization Function Library (OFL). If the function passes validation, you are allowed to configure its variables. If not reenter the function following the formatting rules outlined in the previous step.
   f. Configure the optimization function variables. Choose a variable type of *Nominal* (integers), *Continuous* (the resulting values for the variable do not need to be integers) or *Binary* (one or zero); Select a positive or negative range (+/-), or both; optionally, enter an alias that more accurately identifies your variable, such as Product or Store Location.

      > **Note**
      >
      > Optionally, you can click *Save* to store the function in the Optimization section of the component list in the Predict room.

g. Click *Next* to move to the next screen.

3. On the *Function Constraints* tabbed page, enter the following information:
   a. In the Function Constraints text box, enter any amount of linear constraints. The constraints are enforced to maximise (or minimize) your objective functions.

> **Note**
>
> Enter linear constraints in the format, *5x + 8y >= 400*; *16y +(-4)z >= 200*. Note that you must include semicolon (;) signs as separators between constraints, except for the final constraint. Each constraints must include monomes (e.g. *5x+8y*), a constraint type (e.g. >=) and a constraint value (e.g. *400*). Otherwise, the application returns an error.

   b. Click *Validate Constraints*. If your constraint(s) throw an error, consult the rules outlined in the previous step and re-enter your constraints.
   c. Click *Solve*.

> **Note**
>
> Optionally, you can click *Save* to store the function in the Optimization section of the component list in the Predict room. You can edit the function to try when you reopen.

4. Click *Results* to open the Results page.

In the Results page, you view the optimization result. For example, consider a result of *x:130.0* and *y:20.0*. In this case, to maximize or minimize your function, the output for *x* (or its alias) should be 130.0. And for *y* (or its alias), the output should be 20.0. The results also display an overview of the function and constraints that you entered.

Optionally, you can do the following:

- Select *Split Lines* checkbox to view each function on a separate line for easier readability.
- Select the *Use Alias* checkbox to list your function variables by their chosen aliases.
- Copy and paste the results using the standard short cut keys, Ctrl A and Ctrl C, into a text editor or report tool.
- Click *Save* to store the function in the Optimization section of the component list in the Predict room.

You can now configure the HANA Optimization Function for use as a preprocessing step in a complex analysis.

12 Working with Models

12.1 Creating a Model

A model is a reusable component created by training an algorithm using historical data and saving the instance.

To create a model, you need to save the state of the algorithm. Typically, you create models for the following reasons:

- To share computed business rules that can be applied to similar data
- To predict unseen data using the trained instance of the algorithm

1. Acquire data from the required data source.
   The data source component is added to a new analysis in the *Predict* room.
2. In the *Predict* room, double-click the required algorithm component.
3. From the context menu for the component, choose *Configure Settings* and configure the component settings.
4. Choose  (Run Analysis).
5. From the context menu for the algorithm, choose *Save as Model*.
6. Enter a name and description for the model.
7. If a model with the same name already exists, select the *Overwrite, if exists* option to overwrite the existing model.
8. Choose *Save*.
9. Choose *OK*.

The model is created and appears in the *Models* section under the list of components on the right. You can use this model just like any other component for creating an analysis.

> **Note**
>
> Independent column names used while scoring the model should be the same as the independent column names used while creating the model.

12.2 Sharing Models via PMML

You can export the model information into a local file in industry-standard Predictive Modeling Markup Language (PMML) format and share the model with other PMML compliant applications to perform analysis on similar data set.

To export a model in the PMML format, perform the following steps.

1. Create a model.
2. In the *Predict* room, from the *Models* section, double-click the required model.
3. From the contextual menu of the model, choose *Export Model*.
4. Select *Use this option to export data models into the Predictive Model Markup Language (*.pmml) file*.
5. Choose *Export*.
6. Enter a name for the file.
7. Select the file type, either PMML or XML, as required.
8. Choose *Save*.

12.3 Sharing Models Using Spar Files

You can share models using .spar files.

To share a model, proceed as follows:

1. Create a model.
2. Select the model you want to export.
3. In the component actions, select *Export Model*.

> **Note**
>
> If the model is in the analysis editor, select *Export Model* from the contextual menu.

4. Select *Use this option to export data model to the Expert Analytics Archive (.spar) file*.
5. Click *Export*.
6. Enter a name for the .spar file.
7. Click *Save*.
8. Click *OK*.

To export multiple models into a single .spar file, click *File* > *Export All Models*. Select the models you want to export and click *Export*.

12.4 Sharing Custom Components Using .spar Files

1. Create a custom component.
2. Select the model you want to export.
3. In the component actions, select *Export Model*.

> **Note**
>
> If the model is in the analysis editor, select *Export Model* from the contextual menu.

4. Select *Use this option to export data model to the Expert Analytics Archive (.spar) file*.
5. Click *Export*.
6. Enter a name for the .spar file.
7. Click *Save*.
8. Click *OK*.

To export multiple custom components into a single .spar file, click *File* > *Export All Models*. Select the models you want to export and click *Export*.

12.5 Importing a Model

You can import a model that someone has shared with you using a .spar file.

To import models , proceed as follows:

1. In the *Predict* room, under the list of components on the right, click  > *Import Model*.
2. Select a valid .spar file.
3. Click *Open*.
4. Select the model that you want to import.
5. Click *Finish*.
   The models are imported and displayed under the list of components.

12.6 Deleting a Model

We recommend that you use this option with caution, since deleting a model might make the analysis that contains the model's reference unusable.

To delete a model, perform the following steps:

1. In the *Predict* room, under the list of components, choose *Models*.
2. Select the required model and from the component actions, choose *Delete*.

13 Comparing Models

13.1 Overview

Use the Model Compare component to compare models and learn the best algorithm for your predictive problem. Use in all scenarios, SAP HANA and agnostic.

Why Compare Models?

Comparing models in Expert Analytics enables you to try different algorithms and discover the best one to solve your predictive problem. When comparing the performance of two or more algorithms, you first use the Model Statistics component to calculate performance statistics for either Classification or Rregression algorithms. After which, the Model Compare component compares the calculated performance statistics to pick the best algorithm of those run at execution. Finally, the Model Compare component merges the results to provide a detailed summary on the best performing component.

Configuring Partitions

You can configure partition types in the Model Compare component for more control over your analysis chain. In the Properties Panel of the component, you can select either a Validate or Test partition to compare the performance of the models. The component slices a data set into three subsets called Train, Validate and Test.

The component calculates performance results on every partition, but only on the partition that you select does it identify a winner. The result is the best component of those compared only.

Configuring KPIs

You can choose the type and comparison order of the Key Performance Indicators (KPIs) in your analysis chain.

The following tables define the KPIs specific to the Classification and Regression algorithms.

KPI	Definition
Ki	Predictive power. A quality indicator that corresponds to the proportion of information contained in the target variable that the explanatory variables are able to explain.

KPI	Definition
Kr	Model reliability, or the ability to produce similar on new data. A robustness indicator of the models generated. It indicates the capacity of the model to achieve the same performance when it is applied to a new data set exhibiting the same characteristics as the training data set.
Ki & Kr	Predictive power and model reliability. Gives equal importance to the robustness and generalizing capabilities of the model. For more information, see the definitions above.
AUC	Area Under The Curve. Rank-based measure of the model performance or the predictive power calculated as the area under the Receiver Operating Characteristic curve (ROC).
S(KS)	The distance between the distribution functions of the two classes in binary classification (for example, Class 1 and Class 0). The score that generates the greatest separability between the functions is considered the threshold value for accepting or rejecting the target. The measure of seperability defines how well the model is able to distinguish between the records of two classes. If there are minor deviations in the input data, the model should still be able to identify these patterns and diiferentiate between the two. In this way, seperability is a metric of how good the model is; the greater the seperability, the greater the model. Note that the predictive model producing the greatest amount of separability between the two distributions is considered the superior model.
Gain % (Profit %)	The gain or profit that is realized by the model based on a percentage of the target population selection.
Lift %	The amount of lift that the trained model gives in comparison to a random model. It enables you to examine of the difference between a perfect model, a random model and the model created.

KPI	Definition
Ki	Predictive power. A quality indicator that corresponds to the proportion of information contained in the target variable that the explanatory variables are able to explain.

KPI	Definition
Kr	Model reliability, or the ability to produce similar on new data. A robustness indicator of the models generated. It indicates the capacity of the model to achieve the same performance when it is applied to a new data set exhibiting the same characteristics as the training data set.
Ki & Kr	Predictive power and model reliability. Gives equal importance to the robustness and generalizing capabilities of the model. For more information, see the definitions above.
R2	The determination coefficient R2 is the proportion of variability in a data set that is accounted for by a statistical model; the ratio between the variability (sum of squares) of the prediction and the variability (sum of squares) of the data.
L1	The mean absolute error L1 is the mean of the absolute values of the differences between predictions and actual results (for example, city block distance or Manhattan distance)
L2	The mean square error L2 is the square root of the mean of the quadratics errors (that is, Euclidian Distance or root mean squared error – RMSE).
Linf	The maximum error Linf is the maximum absolute difference between the predicted and actual values (upper bound); also know as the Chebyshev Distance.
ErrorMean	The mean of the difference between predictions and actual values.
ErrorStdDev	The dispersion of errors around the actual result.

Control over the order is important because if the top KPI cannot identify a winning algorithm, the component can perform calculations with the second KPI in the list, and so on. In addition, a precise percentage can be configured for the Gain and Lift parameters. The result is an even more accurate calculation when comparing two or more components.

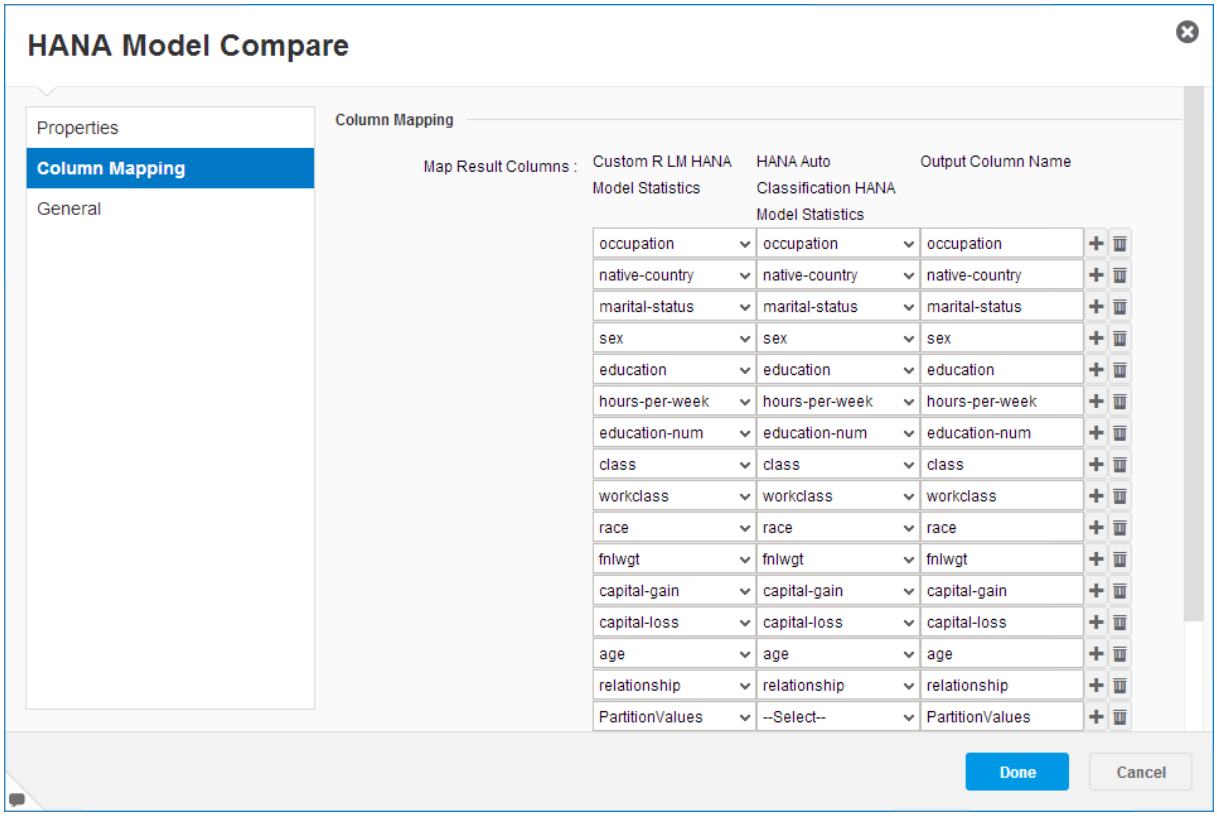
Column Mapping

Column mapping in the Model Compare component enables you to map the output from two compared algorithms. The Column mapping section lists side-by-side the matching column types from both algorithms. A third column is the output column for the Model Compare component. This offers a one-to-one mapping between columns and serves as the result data schema for the Model Compare component. This will feed winning outputs into any following algorithms or components that you can add to the chain, such as a report or a decision tree. The data in the mapped columns comes from the winning component.

Columns are mapped only if the column types match. At first a default mapping is completed that is based on exact names, data and statistical types. After which it checks if the columns are of the same type.

Optionally, you can add or remove columns to include in the Model Compare result set.

The below image shows the Column Mapping panel of the Model Compare component in which you can configure the Partition and the KPIs (using the English language version as an example):



Comparing Two Components

You can perform a model comparison on multiple algorithms in one analysis. However, the Model Compare is designed to behave differently depending on the number of algorithms that you add to the comparison chain. On a model comparison chain that has two parent components, you can create a child node. The child node receives the output of the model comparison and displays it in a configurable mapping screen. This means that you can map the columns from two parent components into one for consumption by a child node. This enables you to perform further analysis on your chain. The Model Compare component displays the following icon when in two-component compare mode:



Comparing Three or More Components

You can perform a model comparison on multiple algorithms in one analysis. When Model Compare has three or more parents, the component becomes a terminal (or leaf) component. Therefore you cannot add a child component to perform further analysis after the original comparison. If you try to compare a third component, you receive an error message. The Model Compare component displays the following icon when comparing three or more components:



Results and Summary

The Results tab shows the Summary of the comparison results, and highlights the best component.

The feedback includes a star icon that indicates the best performing component. This is based on the comparison of performance statistics for the algorithms, which can be either classification or regression types. The Summary sorts the model algorithms in order of performance. It compares the results based on the partition selected, which can be either Test or Validate.

Titles display in the order set in the Model Compare component, with the bolded titles indicating those chosen for comparison. In the case of a classification algorithm, the Gain or Lift settings will default to 10% if you have not specified a percentage.

13.2 Comparing Two Models

Use the Model Compare component to identify the best performer from two algorithms to solve a complex problem in all scenarios (SAP HANA and agnostic). Add a child component to perform further analysis.

Prerequisites: It is mandatory to use the *Model Statistics* and *Partition* components with the *Model Compare* component to create your model comparison chain.

Take the following steps to perform a two-component compare:

1. In Expert Analytics, connect to a *Data Source* and navigate to the *Predict* room.
2. From the Component List choose the *Data Preparation* section.
3. Drag-and-drop a *Partition* component to the analysis editor. Alternatively, double-click the *Partition* component. Click *OK*.
4. In the Algorithms section, drag-and-drop selected algorithms to the analysis editor. For example, if solving a classification problem, you might choose three classification algorithms, *Auto Classification*, *R-CNR Tree*, and *Naïve Bayes*.
5. From the Data Preparation section, add *Model Statistics* components for each chosen algorithm. This enables Expert Analytics to calculate performance statistics on the data set that the algorithms generate.
6. Double-click the *Model Statistics* components to display the configuration options. Alternatively, click the context menu icon on the component and select *Configure Settings*. The result is a configured chain that can perform the model comparison.

7. Set the *Target* and *Predicted* columns to compute performance statistics in both *Model Statistics* components.
8. From the Data Preparation section, add the *Model Compare* component to the analysis editor.
9. Drag-and-drop the *Model Compare* component that you have added to the analysis editor over both of the *Model Statistics* components that you want to compare. After which, the *Model Compare* component is linked to all the components that you wish to compare.

> **Note**
>
> For a two-component compare, the *Model Compare* component enables you to add a child node, which the component indicates by displaying the following icon: 

10. To start configuring the comparison, double-click the *Compare* component to view its configuration settings. Alternatively, on the component click the *Settings*  icon and from the context menu, select *Configure Settings*.
11. In the *Model Compare* dialog box, select a *Validate* or *Test* partition to compare the performance of various components connected to it.

> **Note**
>
> The *Model Compare* component uses the *Validate* setting by default to compare models.

12. In the Performance KPI (Key Performance Indicator) section, take any of the following actions:
    a. Choose the KPIs for use and sort the order in which they should be compared. Control over the order is important because if the top KPI cannot identify a winning algorithm, the component can perform calculations with the second KPI in the list, and so on.
    b. Click the arrows to move KPIs up or down in the comparison order. The input components must be of the same type in the *Model Statistics* component. If not, an error message displays.
    c. Specify the percentage for the Gain comparison. The percentage of the target population must be between 1% and 100%, to one decimal point (for example, 15.3%).

> **Note**
>
> Classification has 7 KPIs = KI, KR, KI + KR, AUC, S(KS), Gain % and Lift %, whereas Regression has 9 KPIs = KI, KR, KI + KR, R2, L1, L2, LInf, ErrorMean and ErrorStdDev.

13. When you have completed the configuration, click *Done*.
14. The analysis chain is now fully configured and ready to be executed. The summary of the *Model Statistics* component shows the KPIs calculated for all partitions. Titles are in the order set in the *Model Compare* component and the Test partitions shows only when the *Model Compare* component exists and Test was selected for comparison.

> **Note**
>
> If all partitions are not available in the algorithm or the *Model Statistics* component, the component considers it as a chain without partition.

15. Click the *Run Analysis*  icon.

> **Note**
>
> The component calculates performance results on every partition, but only on the partition that you select does it identify a winner. The result is the best component of those compared only. It is advised that you ensure that the KI values are greater than 95% before deploying the component in production.

16. The Results tab shows the Summary of the comparison results, and highlights the best component. The feedback includes the following information:
    a. A star icon  indicates the best performing component. This is based on the comparison of performance statistics for the algorithms, which can be either classification or regression types. The Summary sorts the model algorithms in order of performance. It compares the results based on the partition selected, which can be either Test or Validate.
    b. Titles are in the order set in the *Model Compare* component, with the bolded titles indicating those chosen for comparison.
    c. In the case of a classification algorithm, the Profit or the Lift settings will default to 10% if you have not specified a percentage.
17. Optionally, when you are using two parent components, you can extend the analysis by adding a child node with a mapping screen to the *Model Compare* component. To do so, right-click *Model Compare* and select *Configure Settings*. Alternatively, double-click the *Model Compare* component or press F5. After which, a default mapping occurs that is based on column name and type.
18. Optionally, name the columns that result from the mapping for the child component to use. You can add or remove other columns of the same type. To map all other columns, manually add the additional rows.

> **Note**
>
> The data in the mapped columns comes from the winning component. None of the columns in the configuration window can be empty.

19. Optionally, you can export the best model as a stored procedure for consumption. To do so, in the *Model Compare* component click the *Settings*  icon and from the resulting context menu, select *Export as Stored Procedure*.
20. Optionally, you can save and export the best chain directly from the *Model Compare* component. To do so, in *Model Compare* click the *Settings*  icon and from the resulting context menu, select *Save as Model*.

You can now use the Partition, Model Statistics and Model Comparison components in unison to compare multiple algorithms to find the best one to use in a complex analysis.

13.3 Comparing Three or More Models

You can compare three or more algorithms to discover the best one to solve a predictive problem in all scenarios (SAP HANA and agnostic). However, when your analysis has three or more algorithms, you cannot add a child node to perform further fine-grained analysis.

Prerequisites: It is mandatory to use the *Model Statistics* and *Partition* components with the *Model Compare* component to create your model comparison chain.

Take the following steps to compare three or more algorithms:

1. In Expert Analytics, connect to a *Data Source* and navigate to the *Predict* room.
2. Drag-and-drop a *Partition* component to the analysis editor.
3. In the Algorithms section, drag-and-drop selected three or more algorithms to the analysis editor.
4. Add to the analysis chain the appropriate *Model Statistics* and *Model Compare* components.
5. Drag-and-drop the *Model Compare* component that you added to the analysis editor over all of the *Model Statistics* components that you want to compare.

   > **Note**
   >
   > After dragging-and-dropping the *Model Compare* component over the third *Model Statistics* component, the *Model Compare* will become a terminal (or leaf) component in a chain because the component has three parents. Be aware that you cannot perform further analysis on a terminal component. The component displays the following icon when comparing three or more components: 

6. Name the columns that result from the mapping for the child component to use. Optionally, you can add or remove other columns of the same type. To map all other columns, manually add the additional rows.

   > **Note**
   >
   > The data in the mapped columns comes from the winning component. None of the columns in the configuration window can be empty.

You can now use the Partition, Model Statistics and Model Comparison components to compare three or more algorithms to solve a complex analysis.

14 Exporting Models and Analyses

14.1 Exporting Models and Extensions

Expert Analytics enables you to export trained models as stored procedures and views to SAP HANA and to a local folder as a zip file. You can export a model as a single component or as part of an entire chain. When exporting an entire chain, ensure that you include only supported components when constructing chains.

Supported Components in Export of Entire Chain

To export an entire chain to SAP HANA, construct the chain with the following components only:

CATEGORY	TYPE	COMPONENTS
ANY	R Extensions	Any component
CLASSIFICATION	PAL	HANA Naive Bayes
CLASSIFICATION	PAL	HANA Support Vector Machine
CLASSIFICATION	R	HANA R-Bagging Classification
CLASSIFICATION	R	HANA R-Boosting Classification
CLASSIFICATION	R	HANA R-Random Forest Classification
CLASSIFICATION	APL	HANA Auto Classification
CLUSTERING	APL	HANA Auto Clustering
DECISION TREE	PAL	HANA C 4.5
DECISION TREE	PAL	HANA CHAID
DECISION TREE	PAL	HANA R-CNR Tree
PREPROCESSING	PAL	HANA Filter
REGRESSION	APL	HANA Auto Regression
REGRESSION	PAL	HANA Exponential Regression
REGRESSION	PAL	HANA Geometric Regression

CATEGORY	TYPE	COMPONENTS
REGRESSION	PAL	HANA Logistic Regression
REGRESSION	PAL	HANA Logarithmic Regression
REGRESSION	PAL	HANA Multiple Linear Regression
REGRESSION	PAL	HANA Polynomial Regression
REGRESSION	R	HANA R-Multiple Linear Regression
REGRESSION	R	HANA R-Random Forest Regression

Related Information

Exporting a Single SAP HANA Model as a Stored Procedure [page 165]
Exporting a Single R Extension as a Stored Procedure from Expert Analytics [page 166]
Exporting an Analysis as a Stored Procedure [page 168]
Exporting a Chain of Trained Models [page 169]
Removing an Exported Stored Procedure from SAP HANA [page 170]

14.2 Exporting a Single SAP HANA Model as a Stored Procedure

You can export an SAP HANA model as a stored procedure to SAP HANA database. Any SAP HANA user can consume those models for analysis.

- You must have a model created and saved in the list of components under *Models*.
- Before exporting an SAP HANA model as a stored procedure, ensure that your account is defined in SAP HANA.

> **Note**
>
> You can export models that contain SAP Automated Predictive Library (APL) or custom Predictive Analysis Library (PAL) components.

1. In the *Predict* room, from the list of components on the right, choose *Models*.
2. Select the required model and from the *Component Actions* section, choose *Export Model*.
3. Select *Use this option to export an SAP HANA Model as a stored procedure*.
4. Click *Export*.
5. Select the required schema under which you want the procedure to appear in SAP HANA.

6. Specify a procedure name.

> **Note**
>
> If you want to overwrite an existing procedure with the same name in the selected schema, select *Overwrite, if exists*.

7. Choose *Export*.

This functionality exports the single model that you have selected as a stored procedure, as opposed to the entire analysis chain. The exported stored procedure and associated objects (tables, types) appear under the selected schema in the SAP HANA database. You can consume a stored procedure for use outside of Expert Analytics.

> **Tip**
>
> The following is an example call on an exported stored procedure in SAP HANA:

```
CREATE TABLE InputData like PAS00AMYWGCT0Y_ZE4LISJ2MWSCOREPROCEDURE_INPUT_TYPE;
--- Insert the data that you would like to score on into the InputData table:
CREATE TABLE ResultTable like
PAS00AMYWGCT0Y_ZE4LISJ2MWSCOREPROCEDURE_OUTPUT_TYPE;
call "ANALYTICS"."ScoreProcedure"(InputData,ResultTable) WITH OVERVIEW;
select * from ResultTable;
```

Where the following entities should be replaced with your own:

- PAS00AMYWGCT0Y_ZE4LISJ2MWSCOREPROCEDURE_INPUT_TYPE: The table type that defines the input columns of the exported stored procedure. It contains all feature columns that you used to train your model.
- PAS00AMYWGCT0Y_ZE4LISJ2MWSCOREPROCEDURE_OUTPUT_TYPE: The table type that defines the output columns of the exported stored procedure. It contains all feature columns that you used to train your model, plus the columns that your stored procedure generates.
- ANALYTICS: The schema name to which you exported your stored procedure.

Related Information

Creating a Model [page 153]

14.3 Exporting a Single R Extension as a Stored Procedure from Expert Analytics

You can export a single R Extension as a stored procedure from Expert Analytics to SAP HANA. After which, any SAP HANA user can consume those models for analysis.

- You must have a single R Extension created and saved in Expert Analytics.
- Before exporting a model as a stored procedure, ensure that your account is defined in SAP HANA.

To export an R Extension as a stored procedure, take the following steps.

1. Go to *Predict* > *Components List* > *Models* > *R Extensions* and select an existing R Extension for export.
2. Click *Export Model*.
3. Select *Use this option to export a SAP HANA Model as a stored procedure*.
4. Click *Export*.
5. Select the required schema under which you want the procedure to appear in SAP HANA.
6. Specify a procedure name.

   > **Note**
   >
   > If you want to overwrite an existing procedure with the same name in the selected schema, select *Overwrite, if exists*.

7. Select *Export*.

You have exported a single R Extension to SAP HANA. On calling the procedure, the result grid consists of the features that you selected to train your model on, plus all predicted columns.

This functionality exports the single model that you have selected as a stored procedure, as opposed to the entire analysis chain. The exported stored procedure and associated objects (tables, types) appear under the selected schema in the SAP HANA database.

> **Tip**
>
> The following is an example call on an exported stored procedure in SAP HANA:

```
CREATE TABLE InputData like PAS00AMYWGCT0Y_ZE4LISJ2MWSCOREPROCEDURE_INPUT_TYPE;

--- Insert the data that you would like to score on into the InputData table:

CREATE TABLE ResultTable like
PAS00AMYWGCT0Y_ZE4LISJ2MWSCOREPROCEDURE_OUTPUT_TYPE;

call "ANALYTICS"."ScoreProcedure"(InputData,ResultTable) WITH OVERVIEW;

select * from ResultTable;
```

Where the following entities should be replaced with your own:

- PAS00AMYWGCT0Y_ZE4LISJ2MWSCOREPROCEDURE_INPUT_TYPE: The table type that defines the input columns of the exported stored procedure. It contains all feature columns that you used to train your model.
- PAS00AMYWGCT0Y_ZE4LISJ2MWSCOREPROCEDURE_OUTPUT_TYPE: The table type that defines the output columns of the exported stored procedure. It contains all feature columns that you used to train your model, plus the columns that your stored procedure generates.
- ANALYTICS: The schema name to which you exported your stored procedure.

14.4 Exporting an Analysis as a Stored Procedure

You can export an SAP HANA analysis -- including a Demand Forecast analysis -- into the SAP HANA database. Any SAP HANA user can consume that analysis in SAP HANA Studio.

Before exporting an analysis as a stored procedure in SAP HANA database, ensure that your account is defined in SAP HANA. Afterwards, take the following steps:

1. Create an analysis in *Predict* room.
2. Run the analysis.
3. Select the last algorithm component in the analysis and from the context menu -- or from Component Actions under the Component List -- select *Export as Stored Procedure*.
4. In the Export Options panel, select to export to SAP HANA. The*Export to SAP HANA* option provides the further option to overwrite an existing procedure with the same name. It will replace any current version in Expert Analytics and SAP HANA.
5. Select the schema name.
6. Enter a procedure name.
7. Optional: Enter a name for the view.
8. Select *Export*.

The entire analysis chain is exported as a stored procedure. The exported procedure and the associated objects appear under the selected schema in the SAP HANA database.

The following objects generate in SAP HANA database under the selected schema:

- Procedure with name specified.
- If you decided to create a view, then a column view with name specified.
- One or more procedures for each component in the analysis chain.
- One column view for each component in the chain.
- Table types needed by SAP HANA algorithms.
- Tables required by these algorithms that may contain input parameters.

Example

Consuming the Stored Procedure

Tip

The following SQL snippet shows how you can consume the exported stored procedure:

```
CREATE TABLE  ResultTable like "TestProc_OUT_TYPE";
                    call "TEST80"."TestProc"(ResultTable) WITH OVERVIEW;
                    select * from ResultTable ;
```

TestProc_OUT_TYPE is the same as the table type that TestProc expects as out parameter.

Consuming the Column View

Tip

There is an additional feature available where the user can create column views on top of the stored procedure:

```
Select * from VIEW_TESTPROC
```

Related Information

Exporting a Single SAP HANA Model as a Stored Procedure [page 165]

14.5 Exporting a Chain of Trained Models

You can export an entire chain of trained models from Expert Analytics to SAP HANA, where the chain can be consumed directly as a procedure or view. You can also export a model chain in a zip file to a local folder.

System Requirements:

SAP HANA system with Predictive Analytics Library (PAL), Automated Predictive Library (APL) and R configured. To get the latest version of SAP HANA and associated libraries, go to the SAP Product Availability Matrix.

1. Create and execute a chain of models in *Predict* room. In the chain you can use different types of components, including PAL, APL, and R Extensions.
2. To export the chain to SAP HANA, click a *Model Statistics* or *Model Compare* component on the chain. Next, from its context menu -- or from Component Actions under the Component List -- select *Export as Model Chain*.
3. In the General panel, enter a name for the model chain and, optionally, a description.
4. In the Export Options panel, you have the option to select *Export to SAP HANA* to save the model chain to SAP HANA. When exporting to SAP HANA, you have the further option to select the *Overwrite, if exists* checkbox. This checkbox will replace any existing version of the model chain with the same name in Expert Analytics and SAP HANA.
5. When exporting to SAP HANA, enter the details required to generate metadata for the exported chain. Add Schema, Procedure, View, and Model Chain names. Optionally, add a model chain description.
6. Alternatively, you can choose to export the model chain locally in a zip file. Select the *Export to file* checkbox and browse to select a destination folder. Note that when this export option is selected, you cannot enter metadata in the dialog box.
7. Click *Export*.

The entire chain of trained models is exported to SAP HANA. Here you can work with the model chain in a single stored procedure or view. Alternatively, the model chain has been exported in a zip file to the location of your choice.

Example

Work with Model Chain as an SAP HANA Stored Procedure

To run the model chain against a specified data set, call the stored procedure with the data set specified. After which, the results display in the SAP HANA User Interface (UI).

The following are sample calls:

- SAP HANA UI sample call:

```
call "myschema"."mymodel"("someSchema"."someTable", [?])
```

- Specified table sample call:

```
call "myschema"."mymodel"("someSchema"."someTable",
"someSchema"."someOutputTable")
```

Work with Model Chain in the SAP HANA Result View

To run the model chain against a data set trained in Expert Analytics, select from the Result View. The following is a sample SELECT statement:

```
SELECT * FROM "myschema"."my_model_result_view"
```

14.6 Removing an Exported Stored Procedure from SAP HANA

You can delete an exported stored procedure from SAP HANA using SAP HANA Studio. Ensure that your account is defined in SAP HANA.

To remove the exported stored procedure from SAP HANA, perform the following steps:

1. In SAP HANA Studio, navigate to the procedure that you exported.

   ### Note

   You can find the exported procedure under the Procedure folder of the schema.

2. Right-click the procedure and choose *Open Definition*.
   The *Definition tab* appears.
3. Under *Definition tab*, choose *Create Statement* tab.
4. On the *Create Statement* tab, copy the SQL comments (that is, commands preceded with double hyphen '--').
5. On the Navigator tab, right-click the procedure and select *SQL Console*.
   The *SQL Console* tab appears.
6. On the *SQL Console* tab, paste the SQL comments and choose *Execute*, or press F8.

> **Note**
>
> Ensure that before executing the comments, you delete the double hyphen (--) that precedes the SQL comments.

15 Component Properties

15.1 Algorithms

Use algorithm components to perform data mining and statistical analysis on your data; for example, to determine trends and patterns in data.

Expert Analytics provides built-in algorithms such as regressions, time series and outliers. In addition, it supports decision trees, k-means, neural network, time series and regression algorithms from the open-source R library. You can also perform in-database analysis using Predictive Analysis Library (PAL) or SAP Automated Predictive Library (APL) algorithms from SAP HANA.

15.1.1 R Algorithms and Dependent Packages

Expert Analytics supports many R algorithms.

The following tables detail the supported R algorithms and dependent packages for both SAP HANA and agnostic systems.

R Algorithm	Dependent Packages
HANA R-Apriori	arules
HANA R-CNR Tree	rpart
HANA R-Multiple Linear Regression	stats
HANA R-Triple Exponential Smoothing	stats
HANA R-Bagging Classification	adabag, rpart
HANA R-Boosting Classification	adabag, rpart
HANA R-Random Forest Classification	randomForest
HANA R-Random Forest Regression	randomForest

R Algorithm	Dependent Packages
R-CNR Tree	rpart
R-Apriori	arules
R-K-Means	stats

R Algorithm	Dependent Packages
R-Linear Regression	stats
R-Multiple Linear Regression	stats
R-Exponential Regression	stats
R-Geometric Regression	stats
R-MONMLP Neural Network	monmlp
R-NNet Neural Network	nnet
R-Single Exponential Smoothing	stats
R-Double Exponential Smoothing	stats
R-Triple Exponential Smoothing	stats
R-Bagging Classification	adabag, rpart
R-Boosting Classification	adabag, rpart
R-Random Forest Classification	randomForest
R-Random Forest Regression	randomForest

15.1.2 Association

Association algorithms that are available in Expert Analytics.

15.1.2.1 HANA Apriori

Properties that can be configured for the HANA Apriori algorithm.

Syntax

Use this algorithm to find frequent itemsets patterns in large transactional data sets for generating association rules. This algorithm is used to understand what products and services customers tend to purchase at the same time. By analyzing the purchasing trends of customers with association analysis, you can predict their future behavior.

For example, the information that a customer who buys shoes is more likely to buy socks at the same time can be represented in an association rule (with a given minimum support and minimum confidence) as:

```
Shoes=> Socks [support = 0.5, confidence= 0.1]
```

> **Note**
>
> Creating models using the HANA Apriori algorithm is not supported.

HANA Apriori Properties

Property	Description
Apriori Type	Choose *Apriori*.
Item Column	Select the columns containing the items to which you want to apply the algorithm.
TransactionID Column	Select the column containing the transaction IDs to which you want to apply the algorithm.
Missing Values	Select the method for handling missing values. Possible values: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains missing values for processing.
Support	Enter a value for the minimum support of an item. The default value is 0.1.
Confidence	Enter a value for the minimum confidence of rules/association. The default value is 0.8.
Maximum Item Count	Enter the length of leading items and dependent items in the output. The default value is 5.
Number of Threads	Enter the number of threads using which the algorithm should execute. The default value is 1.

15.1.2.2 HANA AprioriLite

Properties that can be configured for the HANA AprioriLite algorithm.

> **Syntax**
>
> Use this algorithm to find frequent itemset patterns in large transactional data sets to generate association rules. Apriori Lite also supports sampling within the algorithm.

> **Note**
>
> - You can use HANA AprioriLite from within HANA Apriori algorithm properties by selecting *AprioriLite* as the *Apriori Type*.

- Creating models using the HANA AprioriLite algorithm is not supported.
- It only calculates two large itemsets.

HANA AprioriLite Properties

Property	Description
Apriori Type	Click *AprioriLite*.
Item Column	Select the columns containing the items to which you want to apply the algorithm.
TransactionID Column	Select the column containing the transaction IDs to which you want to apply the algorithm.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains missing values for processing.
Support	Enter a value for the minimum support of an item. The default value is 0.1.
Confidence	Enter a value for the minimum confidence of rules/association. The default value is 0.8.
Sampling Required	Select this option if you want to sample the data.
Sampling Percentage	Enter the sampling percentage.
Recalculation Required	Select this option if you want to recalculate the support and confidence in each iteration.
Number of Threads	Enter the number of threads to be used for execution.

15.1.2.3 HANA R-Apriori

Properties that can be configured for the HANA R-Apriori algorithm.

Syntax

Use this algorithm to find frequent itemsets patterns in large transactional data sets for generating association rules using the "arules" R package. This algorithm is used to understand what products and services customers tend to purchase at the same time. By analyzing the purchasing trends of customers with association analysis, prediction of their future behavior can be made.

For example, the information that a customer who buys shoes is more likely to buy socks at the same time can be represented in an association rule (with a given minimum support and minimum confidence) as:
`Shoes=> Socks [support = 0.5, confidence= 0.1]`

HANA R-Apriori Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm.
Input Format	Select the format of the input data.
Item Column(s)	Select the columns containing the items to which you want to apply the algorithm.
TransactionID Column	Select the column containing the transaction IDs to which you want to apply the algorithm.
Support	Enter a value for the minimum support of an item.
Confidence	Enter a value for the minimum confidence of rules/association.
Rules	Enter a name for the new column that contains the apriori rules for the given data set.
Support Values	Enter a name for the new column that contains the support for the corresponding rules.
Confidence Values	Enter a name for the new column that contains the confidence values for the corresponding rules.
Lift values	Enter a name for the new column that contains the lift values for the corresponding rules.
Transaction ID	Enter a name for the new column that contains transaction ID.
Items	Enter a name for the new column that contains the names of the items.
Matching Rules	Enter a name for the new column that contains the matching rules.
Lhs Item(s)	Enter comma-separated labels for the items which should appear on the left hand side of rules or itemsets.
Rhs Item(s)	Enter comma-separated labels for the items which should appear on the right hand side of rules or itemsets.
Both Item(s)	Enter comma-separated labels for the items which should appear on both sides of rules or itemsets.
None Item(s)	Enter a comma-separated labels of the items which need not appear in the rules or itemsets.
Default Appearance	Enter default appearance of items that are not explicitly mentioned.

Property	Description
Sort Type	Select the sort option to sort items with respect to their frequency.
Filter Criteria	Enter a numerical value that indicates how to filter unused items from transactions. The default value is 0.1.
Use Tree Structure	To organize transactions as a prefix tree, select True.
Use HeapSort	To use heapsort instead of quick sort for sorting transactions, select True.
Optimize Memory	To minimize memory usage instead of maximizing speed, select True.
Load Transactions into Memory	To load transactions into memory, select True.

15.1.2.4 R-Apriori

Properties that can be configured for the R-Apriori algorithm.

Syntax

Use this algorithm to find frequent itemsets patterns in large transactional data sets for generating association rules using the "arules" R package. This algorithm is used to understand what products and services customers tend to purchase at the same time. By analyzing the purchasing trends of customers with association analysis, prediction of their future behavior can be made.

For example, the information that a customer who buys shoes is more likely to buy socks at the same time can be represented in an association rule (with a given minimum support and minimum confidence) as:

```
Shoes=> Socks [support = 0.5, confidence= 0.1]
```

R-Apriori Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm.
Input Format	Select the format of the input data.
Item Column(s)	Select the columns containing the items to which you want to apply the algorithm.
TransactionID Column	Select the column containing the transaction IDs to which you want to apply the algorithm.
Support	Enter a value for the minimum support of an item. The default value is 0.1.

Property	Description
Confidence	Enter a value for the minimum confidence of rules/association. The default value is 0.8.
Rules	Enter a name for the new column that contains the apriori rules for the given data set.
Support Values	Enter a name for the new column that contains the support for the corresponding rules.
Confidence Values	Enter a name for the new column that contains the confidence values for the corresponding rules.
Lift values	Enter a name for the new column that contains the lift values for the corresponding rules.
Transaction ID	Enter a name for the new column that contains transaction ID.
Items	Enter a name for the new column that contains the names of the items.
Matching Rules	Enter a name for the new column that contains the matching rules.
Lhs Item(s)	Enter comma-separated labels for the items which should appear on the left hand side of rules or itemsets.
Rhs Item(s)	Enter comma-separated labels for the items which should appear on the right hand side of rules or itemsets.
Both Item(s)	Enter comma-separated labels for the items which should appear on both sides of rules or itemsets.
None Item(s)	Enter a comma-separated labels of the items which need not appear in the rules or itemsets.
Default Appearance	Enter default appearance of items that are not explicitly mentioned.
Sort Type	Select the sort option to sort items by their frequency.
Filter Criteria	Enter a numerical value that indicates how to filter unused items from transactions. The default value is 0.1.
Use Tree Structure	To organize transactions as a prefix tree, select True.
Use HeapSort	To use heapsort instead of quick sort for sorting the transactions, select True.
Optimize Memory	To minimize memory usage instead of maximizing speed, select True.
Load Transaction into Memory	To load transactions into memory, select True.

15.1.3 Classification

Classification algorithms that are available in Expert Analytics.

15.1.3.1 HANA ABC Analysis

Properties that can be configured for the HANA ABC Analysis algorithm.

Syntax

Use this algorithm to classify objects (such as customers, employees, or products) based on a particular measure (such as revenue or profit). It suggests that inventories of an organization are not of equal value. Thus, the inventories can be grouped into three categories (A, B, and C) by their estimated importance. "A" items are very important for an organization. "B" items are of medium importance, that is to say, less important than "A" items and more important than "C" items. "C" items are of the least importance.

An example of ABC classification is as follows:

- "A" items – 20% of the items accounts for 70% of the annual consumption value of all items.
- "B" items – 30% of the items accounts for 25% of the annual consumption value of all items.
- "C" items – 50% of the items accounts for 5% of the annual consumption value of all items.

HANA ABC Analysis Properties

Property	Description
Features	Select the input columns with which you want to perform the analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in features or target variables. • Keep: The algorithm retains the record containing missing values during calculation.
Percentage Breakdown of A	Enter the percentage of items that you want to classify under group A. The default value is 40. The possible range is 0-100%. Ensure that the sum of the percentages of items in groups A, B, and C is equal to 100%.
Percentage Breakdown of B	Enter the percentage of items that you want to classify under group B. The default value is 30. The possible range is 0-100%. Ensure that the sum of the percentages of items in groups A, B, and C is equal to 100%.
Percentage Breakdown of C	Enter the percentage of items that you want to classify under group C. The default value is 30. The possible range is 0-100%. Ensure that the sum of the percentages of items in groups A, B, and C is equal to 100%.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 30.

Property	Description
Predicted Column Name	Enter a name for the newly-added column that contains the predicted values.

15.1.3.2 HANA Auto Classification

Properties that can be configured for the HANA Automated (Auto) Classification algorithm.

Syntax

The HANA Automated Classification algorithm is used for binary/categorical classification. This algorithm detects the model type and algorithm used for best fit based on the target variable you select. It also decides whether the input should be continuous or categorical and determines the most appropriate binning for variables. As a result, you can reduce the data preparation and model testing activities that you perform when building a predictive model. In addition, it also creates Train and Validate data sets for model evaluation.

The HANA Auto Classification algorithm is only available in online mode (connected to SAP HANA). There is a similar Auto Classification algorithm available in offline mode.

For more information about the functions used in online Automated algorithms, see the *SAP Automated Predictive Library Reference Guide* (APL) at http://help.sap.com/pa

HANA Automated Classification Properties

Property	Description
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select the target column on which you want to perform the analysis.
Predicted Column Name	Enter a name for a new column that contains the predicted values.

15.1.3.3 HANA KNN

Properties that can be configured for the HANA KNN algorithm.

Syntax

Use this component to classify objects based on the trained sample data. In KNN, objects are classified by the majority votes of its neighbors.

> **Note**
>
> Creating models using the HANA KNN algorithm is not supported.

HANA KNN Properties

Property	Description
Features	Select input columns with which you want to perform the analysis.
Neighborhood Count	Enter the number of neighbors to consider for finding distances. The default value is 5.
Voting Type	Select the voting type for calculating neighborhood count.
Missing Values	Select the method for handling missing values. • Ignore: The algorithm skips the records containing missing values in features or target variables. • Keep: The algorithm retains the missing values.
Schema Name	Enter the schema name that contains the trained data.
Table Name	Enter the table name that contains the trained data.
Independent Columns	Enter input columns, which you want to consider for training data.
Dependent Column	Enter the output column that you want to consider for training data.
Predicted Column Name	Enter a name for the new column that contains the classification values.
Number of Threads	Enter the number of threads using which you want the algorithm to execute. The default value is 1.

15.1.3.4 HANA Naive Bayes

Properties that can be configured for the HANA Naive Bayes algorithm.

> **Syntax**
>
> Naive Bayes is a classification algorithm based on Bayes theorem. It estimates the class-conditional probability by assuming that the attributes are conditionally independent of one another. Despite its simplicity, Naive Bayes works quite well in areas like document classification and spam filtering, and it only requires a small amount of training data to estimate the parameters necessary for classification.

HANA Naive Bayes Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm.
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select the target column for which you want to perform the analysis.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Laplace Smoothing	Enter the smoothing constant for smoothing observations. Smoothing constant must be a double value greater than 0. Enter 0 to disable Laplace smoothing.
Missing Values	Select the method for handling missing values. • Ignore: The algorithm skips the records containing missing values in features or target variables. • Keep: The algorithm retains the records containing missing values during calculation.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 1.

15.1.3.5 HANA R-Bagging Classification

Properties that can be configured for the HANA R-Bagging Classification algorithm.

Overview:

The Bagging algorithm, also known as "Bootstrap aggregating", is a popular ensemble method that can be applied for classification tasks. The algorithm creates random subsets of the original dataset and performs classification on each subset. The predicted values from the classifier are aggregated to form the final prediction. This ensemble method is designed to improve the accuracy and robustness of single classification algorithm on business datasets.

The R packages that implement the algorithm are *adabag* and *rpart*.

> **Note**
>
> In the component, the decision tree method is selected as the classification algorithm.

> **Note**
>
> When the column names contain the hyphen symbol (-), use the Data Type Transformation component to re-define the column name.

HANA R-Bagging Classification Properties

Property	Description
Maximum Depth	Enter the maximum node level in the final tree with the root node counted as level 0. This parameter can be set between 1 and 20 inclusive.
Minimum Split	Enter the minimum number of observations required for splitting a node. The default value is 0. The parameter can be set between 0 and 500 inclusive.
Complexity Parameter	Enter the complexity parameter, which saves computing time by preventing any split that does not improve the fit. The value for the parameter must be between [-1, 1), which is equal to or more than -1 and less than 1.
Number of Trees to Use	Number of trees used in the forest of a decision tree algorithm. The decision tree algorithm is used for bagging. The parameter can be set between 5 and 500 inclusive.
Features	Select the input columns with which you want to perform the analysis.
Target Columns	Select the target column on which you want to perform the analysis.

15.1.3.6 HANA R-Boosting Classification

Properties that can be configured for the HANA R-Boosting Classification algorithm.

Overview:

The Boosting algorithm is a popular ensemble method that can be applied for classification. The *Adaboost.M1* and *Adaboost-SAMME* algorithms are supported in the component. The ensemble method is designed to improve the accuracy and robustness of weak classifiers on business datasets.

The R packages that implement the algorithm are *adabag* and *rpart*.

> **Note**
>
> In this component, the decision tree method is selected as the classification algorithm.

> **Note**
>
> When the column names contain the hyphen symbol (-), use the Data Type component to re-define the column name.

HANA R-Boosting Classification Properties

Property	Description
Maximum Depth	Enter the maximum node level in the final tree with the root node counted as level 0. This parameter can be set between 1 and 20 inclusive.
Minimum Split	Enter the minimum number of observations required for splitting a node. The default value is 0. The parameter can be set between 0 and 500 inclusive.
Complexity Parameter	Enter the complexity parameter, which saves computing time by preventing any split that does not improve the fit. The value for the parameter must be between [-1, 1), which is equal to or more than -1 and less than 1.
Number of Iterations	Number of iterations for which boosting is running. This parameter can be set between 5 and 500 inclusive.
Sample Weights	If TRUE, a bootstrap sample of the training set is drawn by using the weights for each observation on that iteration. If FALSE, every observation is used with its weights.
Weight Updating Coefficient	Three ways to calculate the weight updating coefficient, which is α in AdaBoost.M1 algorithm are as follows: A) 'Breiman': α=1/[2 ln((1-err)/err)], and B) 'Freund': α=ln((1-err)/err), and C) 'Zhu': α=ln((1-err)/err)+ln(N_classes-1).
Features	Select the input columns with which you want to perform the analysis.
Target Columns	Select the target column on which you want to perform the analysis.

15.1.3.7 HANA R-Random Forest Classification

Properties that can be configured for the HANA R-Random Forest Classification algorithm.

Overview:

Random Forest is a popular ensemble method that is used for classification and regression algorithms. The algorithm is performed by constructing a set of decision trees at training time. For a classification task, the output class is based on the majority vote from an individual decision tree in the forest. Compared to other classification algorithms, this ensemble method leads to better accuracy and generalization on business datasets.

The R package that implements the algorithm is *randomForest*.

> **Note**
>
> The maximum level supported on each dataset feature is 53.

HANA R-Random Forest Classification Properties

Property	Description
Features	Select the input columns with which you want to perform the analysis.
Target Columns	Select the target column on which you want to perform the analysis.
Number of Trees to Grow	The amount of trees that are required to grow in the Random Forest. This parameter can be set between 5 and 1000 inclusive.
Minimum terminal nodes	Minimum number of terminal nodes in the decision tree. This parameter can be set between 10 and 500 inclusive.

15.1.3.8 HANA Support Vector Machine

Properties that can be configured for the HANA Support Vector Machine algorithm.

Syntax

Support Vector Machines (SVMs) refer to a family of supervised learning models using the concept of support vector. Compared with many other supervised learning models, SVMs have the advantages in that the models produced by SVMs can be either linear or non-linear, where the latter is realized by a technique called Kernel Trick.

Like most supervised models, SVMs have training and testing phases. In the training phase, a function `f(x)->y` where `f(·)` is a function (can be non-linear) mapping a sample onto a TARGET, is learnt. The training set consists of pairs denoted by $\{x_i, y_i\}$, where `x` denotes a sample represented by several attributes, and `y` denotes a TARGET (supervised information). In the testing phase, the learnt `f(·)` is further used to map a sample with unknown TARGET onto its predicted TARGET.

In the current implementation in PAL, SVMs can be used for the following three tasks:

- **Support Vector Classification (SVC)**
  Classification is one of the most frequent tasks in many fields including machine learning, data mining, computer vision, and business data analysis. Compared with linear classifiers like logistic regression, SVC is able to produce non-linear decision boundary, which leads to better accuracy on some real world dataset. In classification scenario, `f(·)` refers to decision function, and a TARGET refers to a "label" represented by a real number.
- **Support Vector Regression (SVR)**
  SVR is another method for regression analysis. Compared with classical linear regression methods like least square regression, the regression function in SVR can be non-linear. In regression scenario, `f(·)` refers to regression function, and TARGET refers to "response" represented by a real number.
- **Support Vector Ranking**
  This implements a pairwise "learning to rank" algorithm which learns a ranking function from several sets (distinguished by Query ID) of ranked samples. In the scenario of ranking, `f(·)` refers to ranking function, and TARGET refers to score, according to which the final ranking is made. For pairwise ranking, `f(·)` is learnt so that the pairwise relationship expressing the rank of the samples within each set is considered.

Because non-linearity is realized by Kernel Trick, besides the datasets, the kernel type and parameters should be specified as well.

HANA Support Vector Machine Properties

Property	Description
Algorithm Type	Select the type of analysis the algorithm should perform. • Classification • Regression • Ranking
Output Mode	Select the mode in which you want to use the output of this algorithm.
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select the target column on which you want to perform the analysis.
Query ID	Select a Query ID column for Ranking.
Missing Values	Select the method for handling missing values. Possible values: • Ignore: Algorithm skips the records containing missing values in the independent or dependent columns. • Keep: Algorithm retains the records containing missing values during calculation.
Kernel Type	Select the kernel type.
Gamma	Enter the gamma coefficient for the RBF kernel.
Maximum Margin	Enter a trade-off value that you want to consider between the training error and margin.
Degree	Enter a degree for polynomial kernel. The default value is 3.
Linear Coefficient	Enter a value for linear coefficient.
Coefficient Constant	Enter a value for coefficient constant.
Cross Validation	Select this option to use cross validation for calculation.
Normalization Type	Select the type of normalization.
Number of Threads	Enter the number of threads the algorithm should use for execution. The default value is 1.
Predicted Column Name	Enter a name for the newly-created column that contains predicted values.

15.1.3.9 HANA Weighted Score Analysis

Properties that can be configured for the HANA Weighted Score Analysis algorithm.

Syntax

A weighted score table is a method for evaluating alternatives when the importance of each criterion differs. In a weighted score table, each alternative is given a score for each criterion. These scores are then weighted by the importance of each criterion. All of an alternative's weighted scores are then added together to calculate its total weighted score. The alternative with the highest total score should be the best alternative.

You can use weighted score tables to make predictions about future customer behavior. You first create a model based on historical data in the data mining application, and then apply the model to new data to make the prediction. The prediction, that is, the output of the model, is called a score. You can create a single score for your customers by taking into account different dimensions.

A function defined by weighted score tables is a linear combination of functions of a variable.

$f(x_1,...,x_n) = w_1 \times f_1(x_1) + ... + w_n \times f_n(x_n)$

HANA Weighted Score Analysis

Property	Description
Column Name	Select the input column with which you want to perform the analysis.
Type	Select the type as "Discrete" if the selected column has categorical data or select the type as "Continuous" if the selected column has numerical data.
Weights	Enter the weigths for the selected column. The default value is 0.0.
Keys and Scores	Enter the values for keys and scores.
Missing Values	Select the method for handling missing values. • Ignore: The algorithm skips the records containing missing values in features or target variables. • Keep: The algorithm retains missing values.
Number of Threads	Enter the number of threads using which the algorithm should execute. The default value is 1.
Predicted Column Name	Enter a name for the new column that contains the predicted values.

15.1.3.10 Auto Classification

Properties that can be configured for the Automated (Auto) Classification algorithm.

Syntax

The Automated Classification algorithm is used for binary/categorical classification. This algorithm detects the model type and algorithm used for best fit based on the target variable you select. It also decides whether the input should be continuous or categorical and determines the most appropriate binning for variables. As a result, you can reduce the data preparation and model testing activities that you perform when building a predictive model. In addition, it also creates Train and Validate datasets for model evaluation.

Automated Classification Properties

Property	Description
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select the target column on which you want to perform the analysis.
Predicted Column Name	Enter a name for a new column that contains the predicted values.

15.1.3.11 R-Bagging Classification

Properties that can be configured for the R-Bagging Classification algorithm.

Note

To activate the algorithm, apply SAP BusinessObjects Predictive Analytics 2.3 Patch 2 from the SAP Software Download Centre.

Syntax

The Bagging algorithm, also known as "Bootstrap aggregating", is a popular ensemble method that can be applied for classification tasks. The algorithm creates random subsets of the original dataset and performs classification on each subset. The predicted values from the classifier are aggregated to form the final prediction. This ensemble method is designed to improve the accuracy and robustness of single classification algorithm on business datasets.

The R packages that implement the algorithm are *adabag* and *rpart*.

Note

In the R-Bagging component, the decision tree method is selected as the classification algorithm.

> **Note**
>
> When the column names contain the hyphen symbol (-), use the Data Type Transformation component to re-define the column name.

R-Bagging Classification Properties

Property	Description
Maximum Depth	Enter the maximum node level in the final tree with the root node counted as level 0. This parameter can be set between 1 and 20 inclusive.
Minimum Split	Enter the minimum number of observations required for splitting a node. The default value is 0. The parameter can be set between 0 and 500 inclusive.
Complexity Parameter	Enter the complexity parameter, which saves computing time by preventing any split that does not improve the fit. The value for the parameter must be between [-1, 1), which is equal to or more than -1 and less than 1.
Number of Trees to Use	Number of trees used in the forest of a decision tree algorithm. The decision tree algorithm is used for bagging. The parameter can be set between 5 and 500 inclusive.
Features	Select the input columns with which you want to perform the analysis.
Target Columns	Select the target column on which you want to perform the analysis.

15.1.3.12 R-Boosting Classification

Properties that can be configured for the R-Boosting Classification algorithm.

Overview:

The Boosting algorithm is a popular ensemble method that can be applied for classification. The *Adaboost.M1* and *Adaboost-SAMME* algorithms are supported in the component. The ensemble method is designed to improve the accuracy and robustness of weak classifiers on business datasets.

The R packages that implement the algorithm are *adabag* and *rpart*.

> **Note**
>
> In this component, the decision tree method is selected as the classification algorithm.

> **Note**
>
> When the column names contain the hyphen symbol (-), use the Data Type component to re-define the column name.

R-Boosting Classification Properties

Property	Description
Maximum Depth	Enter the maximum node level in the final tree with the root node counted as level 0. This parameter can be set between 1 and 20 inclusive.
Minimum Split	Enter the minimum number of observations required for splitting a node. The default value is 0. The parameter can be set between 0 and 500 inclusive.
Complexity Parameter	Enter the complexity parameter, which saves computing time by preventing any split that does not improve the fit. The value for the parameter must be between [-1, 1), which is equal to or more than -1 and less than 1.
Number of Iterations	Number of iterations for which boosting is running. This parameter can be set between 5 and 500 inclusive.
Sample Weights	If TRUE, a bootstrap sample of the training set is drawn by using the weights for each observation on that iteration. If FALSE, every observation is used with its weights.
Weight Updating Coefficient	Three ways to calculate the weight updating coefficient, which is α in AdaBoost.M1 algorithm are as follows: A) 'Breiman': α=1/[2 ln((1-err)/err)], and B) 'Freund': α=ln((1-err)/err), and C) 'Zhu': α=ln((1-err)/err)+ln(N_classes-1).
Features	Select the input columns with which you want to perform the analysis.
Target Columns	Select the target column on which you want to perform the analysis.

15.1.3.13 R-Random Forest Classification

Properties that can be configured for the R-Random Forest Classification algorithm.

Overview:

Random Forest is a popular ensemble method that is used for classification and regression algorithms. The algorithm is performed by constructing a set of decision trees at training time. For a classification task, the output class is based on the majority vote from an individual decision tree in the forest. Compared to other classification algorithms, this ensemble method leads to better accuracy and generalization on business datasets.

The R package that implements the algorithm is *randomForest*.

> **Note**
>
> The maximum level supported on each dataset feature is 53.

R-Random Forest Classification Properties

Property	Description
Features	Select the input columns with which you want to perform the analysis.
Target Columns	Select the target column on which you want to perform the analysis.
Number of Trees to Grow	The amount of trees that are required to grow in the Random Forest. This parameter can be set between 5 and 1000 inclusive.
Minimum terminal nodes	Minimum number of terminal nodes in the decision tree. This parameter can be set between 10 and 500 inclusive.

15.1.4 Clustering

Clustering algorithms that are available in Expert Analytics.

15.1.4.1 HANA K-Means

Properties that can be configured for the HANA K-Means algorithm.

> **Syntax**
>
> Use this algorithm to cluster observations into groups of related observations without any prior knowledge of those relationships. The algorithm clusters observations into k groups, where k is provided as an input parameter. The algorithm then assigns each observation to clusters based on the proximity of the observation to the mean of the cluster. The process continues until the clusters converge.
>
> **Note**
>
> - You might obtain a different cluster number for each cluster each time you execute the HANA K-Means algorithm. However, the observations in each cluster remain the same.
> - Creating models using the HANA K-Means algorithm is not supported.

HANA K-Means Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm.
Features	Select the input columns with which you want to perform the analysis.
Category Columns	Select the input columns, which you want to consider as category columns.
Categorical Weights	Enter the categorical weights.
Calculate Silhouette	Select this option to calculate silhouette values. Silhouette signifies the quality of clustering. The silhouette value 1 signifies that the clustering is good and 0 signifies that the clustering is bad.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: Algorithm skips the records containing missing values in the independent or dependent columns. • Keep: Algorithm retains the record containing missing values during calculation.
Number of Clusters	Enter the number of groups for clustering. The default value is 5.
Cluster Name	Enter a name for the newly created column that contains the cluster name.
Distance	Enter a name for the newly created column that contains the distance of the clusters from their centroids' name.
Maximum Iterations	Enter the number of iterations allowed for finding clusters. The default value is 100.
Center Calculation Method	Select the method to be used for calculating initial cluster centers.
Distance Measure	Enter the method for calculating the distance between the item and cluster centre.
Normalization Type	Select the type of normalization.
Number of Threads	Enter the number of threads that can be used for execution. The default value is 1.
Exit Threshold	Enter the threshold value for exiting from the iterations. The default value is 0.000000001.

15.1.4.2 HANA R-K-Means

Properties that can be configured for the HANA R-K-Means algorithm.

Syntax

Use this algorithm to cluster observations into groups of related observations without any prior knowledge of those relationships. The algorithm clusters observations into k groups, where k is provided as an input parameter. The algorithm then assigns each observation to clusters based on the proximity of the observation to the mean of the cluster. The process continues until the clusters converge.

Note

- You might obtain a different cluster number for each cluster each time you execute the HANA R-K-Means algorithm. However, the observations in each cluster remain the same.
- Creating models using the HANA R-K-Means algorithm is not supported.

HANA R-K-Means Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm
Features	Select input columns with which you want to perform the analysis.
Number of Clusters	Enter the number of groups for clustering. The default value is 5.
Cluster Name	Enter a name for the newly created column that contains cluster numbers.
Maximum Iterations	Enter the number of iterations allowed for finding clusters. The default value is 100.
Number of Initial Cluster Center Sets	Enter the number of random initial cluster center sets for clustering (n start). The default value is 1.
Initial Cluster Center Seed	Enter a value to randomly select initial cluster centers from acquired data.
Algorithm Type	Select the type of algorithm that you want to use for performing HANA R-K-Means clustering.

15.1.4.3 HANA Self-Organizing Maps

Properties that can be configured for the HANA Self-Organizing Maps algorithm.

Syntax

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. Self-organizing maps are different from other artificial neural networks in that they use a neighborhood function to preserve the topological properties of the input space.

This makes SOMs useful for visualizing low-dimensional views of high-dimensional data, akin to multi-dimensional scaling. The model was first described as an artificial neural network by the Finnish professor Teuvo Kohonen, and is sometimes called a Kohonen map. Like most artificial neural networks, SOMs operate in two modes: training and mapping. Training builds the map using input examples. It is a competitive process, also called vector quantization. Mapping automatically classifies a new input vector.

The SOM approach has many applications, such as virtualization, web document clustering, and recognition of speech.

HANA Self-Organizing Maps Properties

Property	Description
Map Height	Enter the map height. The default value is 5.
Map Width	Enter the map width. The default value is 5.
Alpha	Enter a value for the learning rate. The default value is 0.5.
Map Shape	Select the map shape.
Features	Select input columns with which you want to perform the analysis.
Calculate Silhouette	Select this option to calculate silhouette values. Silhouette signifies the quality of clustering. The silhouette value 1 signifies that the clustering is good and 0 signifies that the clustering is bad.
Cluster Name	Enter a name for the new column that contains the cluster numbers for the given dataset.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the record containing missing values during calculation.

Property	Description
Normalization Type	Select the type of normalization. Possible types: • Normalization not required • New range normalization • Zero score normalization
Random Seed	Enter a random number that you want to use to perform the calculation. If you enter -1, the algorithm selects a random number by itself for calculation. The default value is -1.
Maximum Iterations	Enter the number of iterations you want the algorithm to use for finding clusters. The default value is 100.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 2.

15.1.4.4 HANA DB Scan

Properties that can be configured for the HANA DB Scan algorithm.

Syntax

HANA DB Scan (Density-Based Spatial Clustering of Applications with Noise) is a density-based data clustering algorithm. It finds a number of clusters starting from the estimated density distribution of corresponding nodes.

DB Scan requires two parameters: scan radius (eps) and the minimum number of points required to form a cluster (minPts). The algorithm starts with an arbitrary starting point that has not been visited. This point's eps-neighborhood is retrieved, and if the number of points it contains is equal to or greater than minPts, a cluster is started. Otherwise, the point is labeled as noise. These two parameters are very important and are usually determined by user.

PAL provides a method to automatically determine these two parameters. You can choose to specify the parameters by yourself or let the system determine them for you.

HANA DB Scan Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm.
Define Parameters Automatically	To enable the algorithm to determine the minimum points and the radius parameters automatically, select True; otherwise, False.

Property	Description
Features	Select input columns with which you want to perform the analysis.
Calculate Silhouette	Select this option to calculate silhouette values. Silhouette signifies the quality of clustering. The silhouette value 1 signifies that the clustering is good and 0 signifies that the clustering is bad.
Cluster Name	Enter a name for the new column that contains the cluster numbers for the given dataset (cluster).
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: Algorithm skips the records containing missing values in the independent or dependent columns. • Keep: Algorithm retains the record containing missing values during calculation.
Distance Measure	Select the option for computing the distance between items and cluster center.
Number of Threads	Enter the number of threads the algorithm should use for execution. The default value is 1.

15.1.4.4.1 Auto Clustering

Properties that can be configured for the Automated (Auto) Clustering algorithm in HANA and non-HANA scenarios.

What is Auto Clustering ?

The Auto Clustering algorithm discovers segments in the data with reference to a target variable. This is done by automatically selecting a clustering algorithm and key input variables to generate the best model.

However, you can train Auto Clustering without a target variable. If one is provided, it is used internally to verify the performance of clustering and fine tune the model automatically.

> **Note**
>
> You can see the results of an analysis that uses the Auto Clustering algorithm displayed in chart format. You can also display the summary view of the analysis results.

> **Syntax**
>
> Automated Clustering is a semi-supervised or targeted clustering algorithm designed and optimized to reveal segments that are related to a specific business question. It discovers natural segments or common behaviors in a dataset and provides the description for each of the segments.

Note

When using the Automated Clustering algorithm, we recommend that you trim the values before acquiring the dataset. You can find the *Trim Values* option in the *Advanced Options* section of the "New Dataset" dialog.

For more information about the functions used in online Automated algorithms, see the *SAP Automated Predictive Library Reference Guide* (APL) at http://help.sap.com/pa

HANA Automated Clustering Properties

Property	Description
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select an optional target column for which you want to perform the analysis.
Minimum Number of Clusters	Enter the minimum number of clusters that you want to use for clustering.
Maximum Number of Clusters	Enter the maximum number of clusters that you want to use for clustering.
Predicted Column Name	Enter a name for the newly-created column that contains predicted values.

15.1.4.4.2 R-K-Means

Properties that can be configured for the R-K-Means algorithm.

Syntax

Use this algorithm to cluster observations into groups of related observations without any prior knowledge of those relationships. The algorithm clusters observations into k groups, where k is provided as an input parameter. The algorithm then assigns each observation to clusters based on the proximity of the observation to the mean of the cluster. The process continues until the clusters converge.

Note

- You might obtain a different cluster number for each cluster each time you execute the R-K-Means algorithm. However, the observations in each cluster remain the same.
- Creating models using the R-K-Means algorithm is not supported.

R-K-Means Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm.
Features	Select the input columns with which you want to perform the analysis.
Number of Clusters	Enter the number of groups for clustering.
Cluster Name	Enter a name for the newly created column that contains the cluster name.
Maximum Iterations	Enter the number of iterations allowed for finding clusters. The default value is 100.
No. of Initial Cluster Center Sets	Enter the number of random initial sets of cluster centers for clustering (n start). The default value is 1.
Initial Cluster Center Seed	Enter a value to randomly select initial cluster centers from acquired data.
Algorithm	Select the type of algorithm to be used for performing R-K-Means clustering.

15.1.5 Decision Trees

Decision tree algorithms that are available in Expert Analytics.

15.1.5.1 HANA C 4.5

Properties that can be configured for the HANA C 4.5 algorithm.

Syntax

Use this algorithm to classify observations into groups and predict one or more discrete variables based on other variables.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

HANA C 4.5 Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values. • Fill: Fills missing values in the target column.
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select the target column for which you want to perform the analysis. **Note** It only accepts column with integer data type.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation.
Percentage of Input Data	Enter the percentage of data that you want to consider for analysis.
Minimum Split	Enter the number of records, beyond which the splitting of leaf node is not allowed. The default value is 0.
Columns	Select the independent columns containing numerical values.
Bin Ranges	Enter bin ranges.
Predicted Column name	Enter a name for the new column that contains the predicted value.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 1.

15.1.5.2 HANA R-CNR Tree

Properties that can be configured for the HANA R-CNR Tree algorithm.

Syntax

Use this algorithm to classify observations into groups and predict one or more discrete variables based on other variables. However, you can also use this algorithm to find trends in data.

Note

- The "rpart" package which is part of R 2.15 cannot handle column names with spaces or special characters. The "rpart" package supports only the input column name format that is supported by R dataframe.
- Independent column names used while scoring the model should be same as independent column names used while creating the model.
- Column names containing spaces or any other special character other than period (.) are not supported.

HANA R-CNR Tree Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values. • Fill: Fills missing values in the target column.
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select the target column for which you want to perform the analysis.
Missing Values	Select the method for handling missing values. Possible values: • Ignore: The algorithm skips the records containing missing values in the independent column or the dependent column. • Keep: The algorithm retains the records containing missing values during calculation.

Property	Description
Algorithm Type	Select the type of analysis you want the algorithm to perform. Possible values: • Classification: Use this method - if the dependent variable has categorical values. • Regression: Use this method - if the dependent variable has numerical values.
Minimum Split	Enter the minimum number of observations required for splitting a node. The default value is 10.
Split Criteria	Select the splitting criteria of the node. Possible values: • Gini: Gini impurity. • Information: Information gain.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.
Complexity Parameter	Enter the complexity parameter that saves computing time by preventing any split that does not improve the fit. The default value is 0.005.
Maximum Depth	Enter the maximum node level in the final tree with the root node counted as level 0. **Note** If the maximum depth is greater than 30, the algorithm does not produce results as expected (on 32-bit machines).
Cross Validation	Enter the number of cross validations. A higher cross validation value increases the computational time and produces more accurate results.
Prior Probability	Enter the vector of prior probabilities.
Use Surrogate	Select the surrogate to use in the splitting process. Possible values: • Display Only - an observation with a missing value for the primary split rule is not sent further down the tree. • Use Surrogate - use this option to split subjects missing the primary variable; if all surrogates are missing, the observation is not split. • Stop if missing - if all surrogates are missing, sends the observation in the majority direction.

Property	Description
Surrogate Style	Enter the style that controls the selection of the best surrogate. Possible values: • Use total correct classification - algorithm uses total number of correct classifications to find a potential surrogate variable. • Use percent non missing cases - algorithm uses the percentage of non missing cases classified to find a potential surrogate.
Maximum Surrogate	Enter the maximum number of surrogates to be retained at each node in a tree.
Show Probability	Select the *Show Probability* check box to get the probability of predicted values during scoring of a classification model.

15.1.5.3 HANA CHAID

Properties that can be configured for the HANA CHAID algorithm.

Syntax

CHAID stands for CHi-squared Automatic Interaction Detection. CHAID is a classification method for building decision trees by using chi-square statistics to identify optimal splits.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

HANA CHAID Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm Possible values: • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values. • Fill: Fills missing values in the target column.

Property	Description
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select the target column for which you want to perform the analysis. **Note** It only accepts column with integer data type.
Missing Values	Select the method for handling missing values. Possible values: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation.
Percentage of Input Data	Enter the percentage of data to be considered for analysis.
Minimum split	Enter the minimum number of records for a node, beyond which the splitting of that particular node is not allowed. The default value is 0.
Maximum Depth	Enter the maximum depth of the tree.
Column Name	Select the name of the independent column containing numerical values.
Enter Bin Ranges	Enter bin ranges.
Predicted Column name	Enter a name for the new column that contains the predicted values.
Number of Threads	Enter the number of threads that the algorithm should use during execution.

15.1.5.4 R-CNR Tree

Properties that can be configured for the R-CNR Tree algorithm.

Syntax

Use this algorithm to classify observations into groups and predict one or more discrete variables based on other variables. However, you can also use this algorithm to find trends in data.

Note

- The "rpart" package which is part of R 2.15 cannot handle column names with spaces or special characters. The "rpart" package supports only the input column name format that is supported by R dataframe.

- Independent column names used while scoring the model should be same as independent column names used while creating the model.
- Column names containing spaces or any other special character other than period (.) are not supported.

R-CNR Tree Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values. • Fill: Fills missing values in the target column.
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select the target column for which you want to perform the analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Rpart: The algorithm deletes all observations for which the dependent column is missing. However, it retains those observations for which one or more independent columns are missing. • Ignore: The algorithm skips the records containing missing values in the independent column or the dependent column. • Keep: The algorithm retains the records containing missing values during calculation. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.
Algorithm Type	Select the type of analysis you want the algorithm to perform. Possible values: • Classification: Use this type - if the dependent variable has categorical values. • Regression: Use this type - if the dependent variable has numerical values.

Property	Description
Minimum Split	Enter the minimum number of observations required for splitting a node. The default value is 10.
Split Criteria	Select the splitting criteria of the node. Possible values: • Gini: Gini impurity. • Information: Information gain.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.
Complexity Parameter	Enter the complexity parameter that saves computing time by preventing any split that does not improve the fit. The default value is 0.005.
Maximum Depth	Enter the maximum node level in the final tree with the root node counted as level 0. **Note** If the maximum depth is greater than 30, the algorithm does not produce results as expected (on 32-bit machines).
Cross Validation	Enter the number of cross validations. A higher cross validation value increases the computation time and produces more accurate results.
Prior Probability	Enter the vector of prior probabilities.
Use Surrogate	Select the surrogate to use in the splitting process. Possible values: • Display Only - an observation with a missing value for the primary split rule is not sent further down the tree. • Use Surrogate - use this option to split subjects missing the primary variable; if all surrogates are missing, the observation is not split. • Stop if missing - if all surrogates are missing, the algorithm sends the observation in the majority direction.
Surrogate Style	Enter the style that controls the selection of the best surrogate. Possible values: • Use total correct classification - algorithm uses total number of correct classifications to find a potential surrogate variable. • Use percent non missing cases - algorithm uses the percentage of non missing cases classified to find a potential surrogate.

Property	Description
Maximum Surrogate	Enter the maximum number of surrogates to be retained at each node in a tree.
Show Probability	Select the *Show Probability* check box to get the probability of predicted values during scoring of a classification model.

15.1.6 Outliers

Outlier algorithms that are available in Expert Analytics.

15.1.6.1 HANA Anomaly Detection

Properties that can be configured for the HANA Anomaly Detection algorithm.

Syntax

Use this algorithm to find patterns in data that do not conform to expected behavior.

Note

Creating models using the HANA Anomaly Detection algorithm is not supported.

HANA Anomaly Detection Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm.
Independent Columns	Select the input source columns.
Missing Values	Select the method for handling missing values. Possible values: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation.
Percentage of Anomalies	Enter the percentage value that indicates the proportion of anomalies in the source data. The default value is 10.

Property	Description
Anomaly Detection Method	Select the anomaly detection method. • By distance from the center • By sum of distances from all centers
Maximum Iterations	Enter the number of iterations allowed for finding clusters. The default value is 100.
Center Calculation Method	Select the method to use for calculating the initial cluster centers.
Normalization Type	Select the type of normalization.
Number of Clusters	Enter the number of groups for clustering.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 1.
Exit Threshold	Enter the threshold value for exiting from the iterations. The default value is 0.0001.
Distance Measure	Enter the measure for calculating the distance between the records and cluster centers.
Predicted Column Name	Enter a name for the new column that contains the predicted values.

15.1.6.2 HANA Inter Quartile Range Test

Properties that can be configured for the HANA Inter Quartile Range algorithm.

Syntax

Use this algorithm to find outlying values based on the statistical distribution between the first and third quartiles.

Note

- The input data for the IQR (Inter Quartile Range) Test algorithm must be at least 4 rows.
- Creating models using the HANA Inter Quartile Range Test algorithm is not supported.

HANA Inter Quartile Range Test Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Show Outliers: Adds a Boolean column to the input data specifying if the corresponding value is an outlier. • Remove Outliers: Removes outlying values from the input data.
Independent Column	Select an input source column.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation.
Fence Coefficient	Enter the deviation allowed for values from the inter quartile range. The default value is 1.5.
Predicted Column Name	Enter a name for the new column that contains the predicted values.

15.1.6.3 HANA Variance Test

Properties that can be configured for the HANA Variance Test algorithm.

Syntax

HANA Variance test identifies the outliers in a set of numerical data. The lower boundary and upper boundary for the data are calculated based on the mean and the standard deviation of data and the multiplier value provided by you.

The multiplier is a double type coefficient, which helps you to test whether all the values of a numerical vector are in the range.

If a value is outside the range, this suggests that it does not pass the variance test and the value is therefore marked as an outlier.

Note

Creating models using the HANA Anomaly Detection algorithm is not supported.

HANA Variance Test Properties

Property	Description
Output mode	Select the mode in which you want to use the output of this algorithm. • Show Outliers: Adds a Boolean column to the input data specifying if the corresponding value is an outlier. • Remove Outliers: Removes outlying values from the input data.
Independent Columns	Select the input source columns.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation.
Multiplier	Enter the multiplier value to decide the range of lower and upper boundaries, which helps in identifying the outliers. The default value is 3.0. **Note** Input must be a positive integer value.
Number of Threads	Enter the number of threads that the algorithm should use during execution.
Predicted Column Name	Enter a name for the new column that contains the predicted values.

15.1.6.4 Inter Quartile Range

Properties that can be configured for the Inter Quartile Range algorithm.

Syntax

Use this algorithm to find outlying values based on the statistical distribution between the first and third quartiles.

Note

- The input data for the IQR (Inter Quartile Range) algorithm must be at least 4 rows.
- Creating models using the IQR (Inter Quartile Range) algorithm is not supported.

Inter Quartile Range Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Show Outliers: Adds a Boolean column to the input data specifying if the corresponding value is an outlier. • Remove Outliers: Removes outlying values from the input data.
Feature	Select the input column with which you want to perform the analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.
Fence Coefficient	Enter the deviation allowed for values from the inter quartile range. The default value is 1.5.
Predicted Column Name	Enter a name for the new column that contains the predicted values.

15.1.6.5 Nearest Neighbor Outlier

Properties that can be configured for the Nearest Neighbor Outlier algorithm.

Syntax

Use this algorithm to find outlying values based on the number of neighbors (N) and the average distance of values compared to their nearest N neighbors.

Note

Creating models using the Nearest Neighbor Outlier is not supported.

Nearest Neighbour Outlier Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Show Outliers: Adds a Boolean column to the input data specifying if the corresponding value is an outlier. • Remove Outliers: Removes outlying values from the input data.
Feature	Select the input column with which you want to perform the analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.
Neighborhood Count	Enter the number of neighbors for finding distances. The default value is 5.
Number of Outliers	Enter the number of outliers, which you want to remove.
Predicted Column Name	Enter a name for the new column that contains the predicted values.

15.1.7 Neural Network

Neural network algorithms that are available in Expert Analytics.

15.1.7.1 R-MONMLP Neural Network

Properties that can be configured for the R-MONMLP Neural Network algorithm.

Syntax

Use this algorithm for forecasting, classification, and statistical pattern recognition using R library functions.

Note

R does not support PMML storage for MONMLP Neural Network.

R-MONMLP Neural Network Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values. • Fill: Fills missing values in the target column.
Features	Select the input columns with which you want to perform the analysis.
Target Variable	Select the target column for which you want to perform the analysis.
Hidden Layer1 Neurons	Enter the number of nodes/neurons in the first hidden layer (hidden1). The default value is 5.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Hidden Layer Transfer Function	Select the activation function to be used for the hidden layer (Th).
Output Layer Transfer Function	Select the activation function to be used for the output layer (To).
Derivative of Hidden Layer Transfer Function	Select the derivative of the hidden layer activation function (Th.prime).
Derivative of Output Layer Transfer Function	Select the derivative of the output layer activation function (To.prime).
Hidden Layer2 Neurons	Enter the number of nodes/neurons in the second hidden layer (hidden2). The default value is 0.
Maximum Iterations	Enter the maximum number of iterations for the optimization algorithm (iter.max). The default value is 5000.
Monotone Columns	Enter column indexes to which you want to apply the monotonicity constraint (monotone).
Training Iterations	Enter the number of training iterations after which the cost function calculation stops (iter.stopped).
Initial Weights	Enter an initial weight vector (init.weights).
Maximum Exceptions	Enter the maximum number of exceptions for the optimization routine (max.exceptions).
Scale Dependent Column	To scale dependent columns to zero mean and unit variance prior to fitting, select True (scale.y).
Bagging Required	To use bootstrap aggregation, select True (bag).
Trials to Avoid Local Minima	Enter the number of repeated trials to avoid local minima (n.trials).

Property	Description
No. Ensemble Members	Enter the number of ensemble members to fit (n.ensemble).

15.1.7.2 R-NNet Neural Network

Properties that can be configured for the R-NNet Neural Network algorithm.

Syntax

Use this algorithm for forecasting, classification, and statistical pattern recognition using R library functions.

R-NNet Neural Network Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values. • Fill: Fills missing values in the target column.
Features	Select input columns with which you want to perform the analysis.
Target Variable	Select the target column for which you want to perform the analysis.
Missing Values	Select the method for handling missing values. Possible values: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains missing values. • Stop: The algorithm stops if a value is missing in the independent column or the dependent column.
Hidden Layer Neurons	Enter the number of nodes/neurons in the hidden layer. The default value is 5.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.

Property	Description
Algorithm Type	Select the type of analysis you want the algorithm to perform.
Skip Hidden Layer	To add skip-layer connections from input to output, select True.
Linear Output	To obtain the linear output, select True. If you select the algorithm type as Classification, then this value must be true.
Use Softmax	Select True to use "log-linear model" and "maximum conditional likelihood" fittings. Linout, entropy, softmax, and censored are mutually exclusive.
Use Entropy	To use "Maximum Conditional Likelihood" fitting, select True. By default, the algorithm uses the least-squares method. Possible values: • True: Use the "Maximum Conditional Likelihood" fitting • False: Use the least-squares method
Use Censored	For softmax, a row of (0,1,1) indicates one example each of classes 2 and 3, but for censored it indicates one example each of classes 2 or 3.
Range	Enter initial random weights [-rang, rang]. Set this value to 0.5 unless the input is large. If the input is large, choose the rang using the formula: rang * max(|x|) <= 1.
Weight Decay	Enter a value used for calculating new weights (weight decay).
Maximum Iterations	Enter the maximum number of iterations allowed.
Hessian Matrix Required	To return the Hessian measure at the best set of weights, select True.
Maximum Weights	Enter the maximum number of weights allowed in the calculation. There is no intrinsic limit in the code, but increasing the maximum number of weights may allow fits that are very slow and time-consuming.
Abstol	Enter the value that indicates the perfect fit (abstol).
Reltol	Algorithm terminates if the optimizer is unable to reduce the fit criterion by a factor: 1 - reltol.
Contrasts	Enter the list of contrasts to be used for factors appearing as variables in the model.

15.1.8 Regression

Regression algorithms that are available in Expert Analytics.

15.1.8.1 HANA Auto Regression

Properties that can be configured for the HANA Automated (Auto) Regression algorithm.

Syntax

The HANA Automated Regression algorithm uses a technique called Structural Risk Minimization and builds a polynomial model. This algorithm can handle a very high number of input attributes in an automated fashion to find trends in data. It provides indicators and graphs to ensure that the quality and robustness of trained models can be easily assessed.

The HANA Auto Regression algorithm is only available in online mode (connected to SAP HANA). There is a similar Auto Regression algorithm available in offline mode.

For more information about the functions used in online Automated algorithms, see the *SAP Automated Predictive Library Reference Guide* (APL) at http://help.sap.com/pa

HANA Automated Regression Properties

Property	Description
Features	Select input columns with which you want to perform the regression analysis.
Target Variable	Select the target column for which you want to perform the regression analysis.
Predicted Column Name	Enter a name for the newly-created column that contains predicted values.

15.1.8.2 HANA Exponential Regression

Properties that can be configured for the HANA Exponential Regression algorithm.

Syntax

Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable using an exponential function.

> **Note**
>
> The data type of columns used during model scoring should be same as the data type of columns used while building the model.

HANA Exponential Regression properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Columns	Select the input columns with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • *Ignore*: The algorithm skips the records containing missing values in the independent or dependent columns. • *Keep*: The algorithm retains the records containing missing values during calculation.
Predicted Column Name	Enter a name for the newly-added column that contains the predicted values.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 1.

15.1.8.3 HANA Geometric Regression

Properties that can be configured for the HANA Geometric Regression algorithm.

> **Syntax**
>
> Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable using a geometric function.

> **Note**
>
> The data type of columns used during model scoring should be same as the data type of columns used while building the model.

HANA Geometric Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Columns	Select the input columns with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation.
Predicted Column Name	Enter a name for the newly-added column that contains the predicted values.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 1.

15.1.8.4 HANA Logarithmic Regression

Properties that can be configured for the HANA Logarithmic Regression algorithm.

> **Syntax**
>
> Use this algorithm to find trends in data. This algorithm performs bi-variate logarithmic regression analysis. It determines how an individual variable influences another variable using a Predictive Analysis Library (PAL) logarithmic function.

> **Note**
>
> The data type of columns used during model scoring should be same as the data type of columns used while building the model.

HANA Logarithmic Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Column	Select the input columns with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 1.

15.1.8.5 HANA Logistic Regression

Properties that can be configured for the HANA Logistic Regression algorithm.

> **Syntax**
>
> Use this algorithm when the independent variables are categorical, or a mix of continuous and categorical values. Logistic Regression is a prediction approach similar to Ordinary Least Square (OLS) regression.

> **Note**
>
> The data type of columns used during model scoring should be same as the data type of columns used while building the model.

HANA Logistic Regression properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values. • Fill: Fills missing values in the target column.
Independent Columns	Select the input columns with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Iteration Method	Select the iteration method.
Missing Values	Select the method for handling missing values. Possible methods: • *Ignore*: The algorithm skips the records containing missing values in the independent or dependent columns. • *Keep*: The algorithm retains the records containing missing values during calculation.
Show Fitted Values	Select this option to view the fitted values in a new column.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.
Maximum iteration	Enter the maximum number of iterations allowed to calculate the algorithm coefficient. The default value is 100.
Exit Threshold	Enter the threshold value for exiting from the iterations. The default value is 0.00001.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 4.
Mapping Value for 0	Enter a value for a variable, which is mapped to 0.
Mapping Value for 1	Enter a value for a variable, which is mapped to 1.

15.1.8.6 HANA Multiple Linear Regression

Properties that can be configured for the HANA Multiple Linear Regression algorithm.

Syntax

Use this algorithm to find the linear relationship between a dependent variable and one or more independent variables.

HANA Multiple Linear Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Columns	Select the input columns with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 1.

15.1.8.7 HANA Polynomial Regression

Properties that can be configured for the HANA Polynomial Regression algorithm.

Syntax

Use this algorithm to find the relationship between the independent variable and the dependent variable in a curvilinear fitted line.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

HANA Polynomial Regression properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Columns	Select the input columns with which you want to perform the regression analysis.
Degree of the Polynomial	Enter the greatest exponent value of a polynomial expression.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • *Ignore*: The algorithm skips the records containing missing values in the independent or dependent columns. • *Keep*: The algorithm retains the records containing missing values during calculation.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.
Number of Threads	Enter the number of threads that the algorithm should use during execution. The default value is 1.

15.1.8.8 HANA R-Multiple Linear Regression

Properties that can be configured for the HANA R-Multiple Linear Regression algorithm.

Syntax

Use this algorithm to find the linear relationship between a dependent variable and one or more independent variables.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

HANA R-Multiple Linear Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Columns	Select the input columns with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm ignores the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.
Confidence Level	Enter the confidence level of the algorithm (the accuracy of predictions). The default value is 0.95.

Property	Description
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.

15.1.8.9 HANA R-Random Forest Regression

Properties that can be configured for the HANA R-Random Forest Regression algorithm.

Random Forest is a popular ensemble method that is used for classification and regression algorithms. The algorithm is performed by constructing a set of decision trees at training time. For a regression task, the mean prediction of individual trees is calculated as the output. Compared to other regression algorithms, this ensemble method leads to better accuracy and generalization on business datasets.

The R package that implements the algorithm is *randomForest*.

> **Note**
>
> The maximum level supported on each dataset feature is 53. The level refers to the category, variety or type of values that can be taken by a variable; for example, the column "Gender" has two levels, Male" and "Female". In this case the variable cannot have more than 53 types of values.

HANA R-Random Forest Regression Properties

Property	Description
Features	Select the input columns with which you want to perform the analysis.
Target Columns	Select the target column on which you want to perform the analysis.
Number of Trees to Grow	The amount of trees that are required to grow in the Random Forest. This parameter can be set between 5 and 1000 inclusive.
Minimum terminal nodes	Minimum number of terminal nodes in the decision tree. This parameter can be set between 10 and 500 inclusive.

15.1.8.10 Auto Regression

Properties that can be configured for the Automated (Auto) Regression algorithm.

> **Syntax**
>
> The Automated Regression algorithm uses a technique called Structural Risk Minimization and builds a polynomial model. This algorithm can handle a very high number of input attributes in an automated fashion

to find trends in data. It provides indicators and graphs to ensure that the quality and robustness of trained models can be easily assessed.

Automated Regression Properties

Property	Description
Features	Select input columns with which you want to perform the regression analysis.
Target Variable	Select the target column for which you want to perform the regression analysis.
Predicted Column Name	Enter a name for the newly-created column that contains predicted values.

15.1.8.11 Exponential Regression

Properties that can be configured for the Exponential Regression algorithm.

Syntax

Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable using an exponential function with the least square methodology.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

Exponential Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible modes: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output that contains the predicted values.

Property	Description
Independent Column	Select the input column with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent column. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.

15.1.8.12 Geometric Regression

Properties that can be configured for the Geometric Regression algorithm.

Syntax

Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable using a geometric function with the least square methodology.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

Geometric Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Column	Select the input column with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column
Predicted Column Name	Enter a name for the newly-created column that contains predicted values.

15.1.8.13 Linear Regression

Properties that can be configured for the Linear Regression algorithm.

Syntax

Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable with the least square methodology.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

Linear Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Column	Select the input column with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible values: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.

15.1.8.14 Logarithmic Regression

Properties that can be configured for the Logarithmic Regression algorithm.

Syntax

Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable using a logarithmic function with the least square methodology.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

Logarithmic Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Column	Select the input column with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.

15.1.8.15 R-Exponential Regression

Properties that can be configured for the R-Exponential Regression algorithm.

Syntax

Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable using an exponential function from the R open-source library.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

R-Exponential Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Column	Select the input column with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.
Allow Singular Fit	A Boolean value- if set to true, the aliased coefficients are ignored in the coefficient covariance matrix. If set to false, a model with aliased coefficients produces an error. A model with aliased coefficients signifies that the square matrix x*x is singular.
Contrasts	Select the list of contrasts, which you want to use for factors appearing as variables in the model.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.

15.1.8.16 R-Geometric Regression

Properties that can be configured for the R-Geometric Regression algorithm.

Syntax

Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable using a geometric function from the R open-source library.

> **Note**
>
> The data type of columns used during model scoring should be same as the data type of columns used while building the model.

R-Geometric Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Column	Select the input column with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.
Allow Singular Fit	A Boolean value - if set to true, the aliased coefficients are ignored in the coefficient covariance matrix. If set to false, a model with aliased coefficients produces an error. A model with aliased coefficients signifies that the square matrix x*x is singular.
Contrasts	Select the list of contrasts, which you want to use for factors appearing as variables in the model.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.

15.1.8.17 R-Linear Regression

Properties that can be configured for the R-Linear Regression algorithm.

Syntax

Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable by using the R open-source library.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

R-Linear Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Column	Select the input column with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation. • Stop: The algorithm stops the execution if a value is missing in the independent column or the dependent column.

Property	Description
Allow Singular Fit	A Boolean value - if set to true, the aliased coefficients are ignored in the coefficient covariance matrix. If set to false, a model with aliased coefficients produces an error. A model with aliased coefficients signifies that the square matrix x*x is singular.
Contrasts	Select the list of contrasts, which you want to use for factors appearing as variables in the model.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.

15.1.8.18 R-Logarithmic Regression

Properties that can be configured for the R-Logarithmic Regression algorithm.

Syntax

Use this algorithm to find trends in data. This algorithm performs univariate regression analysis. It determines how an individual variable influences another variable using a logarithmic function from the R open-source library.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

R-Logarithmic Regression Properties

Property	Description
Output Mode	Select the mode in which you want to display the output data. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Column	Select the input source column with which you want to perform regression.
Dependent Column	Select the target column on which you want to perform regression.

Property	Description
Missing Values	Select the method for handling missing values. Possible values: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: The algorithm retains the records containing missing values during calculation. • Stop: The algorithm stops execution - if a value is missing in the independent column or the dependent column.
Allow Singular Fit	A Boolean value - if set to true, the aliased coefficients are ignored in the coefficient covariance matrix. If set to false, a model with aliased coefficients produces an error. A model with aliased coefficients signifies that the square matrix x*x is singular.
Contrasts	Select the list of contrasts to be used for factors appearing as variables in the model.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.

15.1.8.19 R-Multiple Linear Regression

Properties that can be configured for the R-Multiple Linear Regression algorithm.

Syntax

Use this algorithm to find the linear relationship between a dependent variable and one or more independent variables.

Note

The data type of columns used during model scoring should be same as the data type of columns used while building the model.

R-Multiple Linear Regression Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. Possible values: • Fill: Fills missing values in the target column. • Trend: Predicts the values for the dependent column and adds an extra column in the output containing the predicted values.
Independent Columns	Select the input columns with which you want to perform the regression analysis.
Dependent Column	Select the target column for which you want to perform the regression analysis.
Missing Values	Select the method for handling missing values. Possible methods: • Ignore: Algorithm skips the records containing missing values in the independent or dependent columns. • Keep: Retains missing values. • Stop: Algorithm stops the execution if a value is missing in the independent column or the dependent column.
Confidence Level	Enter the confidence level of the algorithm. The default value is 0.95.
Predicted Column Name	Enter a name for the newly-created column that contains the predicted values.

15.1.8.20 R-Random Forest Regression

Properties that can be configured for the R-Random Forest Regression algorithm.

Random Forest is a popular ensemble method that is used for classification and regression algorithms. The algorithm is performed by constructing a set of decision trees at training time. For a regression task, the mean prediction of individual trees is calculated as the output. Compared to other regression algorithms, this ensemble method leads to better accuracy and generalization on business datasets.

The R package that implements the algorithm is *randomForest*.

> **Note**
>
> The maximum level supported on each dataset feature is 53. The level refers to the category, variety or type of values that can be taken by a variable; for example, the column "Gender" has two levels, Male" and "Female". In this case the variable cannot have more than 53 types of values.

R-Random Forest Regression Properties

Property	Description
Features	Select the input columns with which you want to perform the analysis.
Target Columns	Select the target column on which you want to perform the analysis.
Number of Trees to Grow	The amount of trees that are required to grow in the Random Forest. This parameter can be set between 5 and 1000 inclusive.
Minimum terminal nodes	Minimum number of terminal nodes in the decision tree. This parameter can be set between 10 and 500 inclusive.

15.1.9 Time Series

Time Series algorithms that are available in Expert Analytics.

15.1.9.1 HANA Single Exponential Smoothing

Properties that can be configured for the HANA Single Exponential Smoothing algorithm.

Syntax

Use this algorithm to smooth the source data.

Note

Creating models using the HANA Single Exponential Smoothing algorithm is not supported.

HANA Single Exponential Smoothing Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. • Trend: Displays source data along with predicted values for the given dataset. • Forecast: Displays forecasted values for the given time period.

Property	Description
Target Variable	Select the target column for which you want to perform time series analysis.
Period	Select the period for forecasting.
Periods Per Year	Select the period for forecasting. This option is only enabled if you select "Custom" for "Period".
Start Year	Enter the year from which the observations must be considered. For example, 2009, 1987, 2019.
Start Period	Enter the period from which the observations must be considered. The default value is 1.
Periods to Predict	Enter the number of periods to forecast. This value is used only if the output mode is Forecast.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Year Values	Enter a name for the newly created column that contains year values.
Quarter Values	Enter a name for the newly created column that contains quarter values.
Month Values	Enter a name for the newly created column that contains month values.
Period Values	Enter a name for the newly created column that contains period values.
Alpha	Enter a smoothing constant for smoothing observations (base parameters). Range: 0-1.

15.1.9.2 HANA Double Exponential Smoothing

Properties that can be configured for the HANA Double Exponential Smoothing algorithm.

Syntax

Use this algorithm to smooth the source data.

Note

Creating models using the HANA Double Exponential Smoothing algorithm is not supported.

HANA Double Exponential Smoothing Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. • Trend: Displays source data along with predicted values for the given dataset. • Forecast: Displays forecasted values for the given time period.
Target Variable	Select the target column for which you want to perform time series analysis.
Period	Select the period for forecasting.
Periods Per Year	Select the period for forecasting. This option is only enabled if you select "Custom" for "Period".
Start Year	Enter the year from which the observations must be considered. For example, 2009, 1987, 2019.
Start Period	Enter the period from which the observations must be considered.
Periods to Predict	Enter the number of periods to forecast. This value is used only if the output mode is Forecast.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Year Values	Enter a name for the newly created column that contains year values.
Quarter Values	Enter a name for the newly created column that contains quarter values.
Month Values	Enter a name for the newly created column that contains month values.
Period Values	Enter a name for the newly created column that contains period values.
Alpha	Enter a smoothing constant for smoothing observations (base parameters). Range: 0-1.
Beta	Enter a smoothing constant for finding trend parameters. Range: 0-1.

15.1.9.3 HANA Triple Exponential Smoothing

Properties that can be configured for the HANA Triple Exponential Smoothing algorithm.

Syntax

Use this algorithm to smooth the source data and find seasonal trends in data.

> **Note**
>
> Creating models using the HANA Triple Exponential Smoothing algorithm is not supported.

HANA Triple Exponential Smoothing Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. • Trend: Displays source data along with predicted values for the given dataset. • Forecast: Displays forecasted values for the given time period.
Target Variable	Select the target column for which you want to perform time series analysis.
Period	Select the period for forecasting.
Periods Per Year	Select the period for forecasting. This option is only enabled if you select "Custom" for "Period".
Start Year	Enter the year from which the observations must be considered. For example, 2009, 1987, 2019.
Start Period	Enter the period from which the observations must be considered.
Periods to Predict	Enter the number of periods to forecast. This value is used only if the output mode is Forecast.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Year Values	Enter a name for the newly created column that contains year values.
Quarter Values	Enter a name for the newly created column that contains quarter values.
Month Values	Enter a name for the newly created column that contains month values.
Period Values	Enter a name for the newly created column that contains period values.
Alpha	Enter a smoothing constant for smoothing observations (base parameters). Range: 0-1.
Beta	Enter a smoothing constant for finding trend parameters. Range: 0-1.
Gamma	Enter a smoothing constant for finding seasonal trend parameters. Range: 0-1.

15.1.9.4 HANA R-Triple Exponential Smoothing

Properties that can be configured for the HANA R-Triple Exponential Smoothing algorithm.

Syntax

Use this algorithm to smooth the source data and find seasonal trends in data.

HANA R-Triple Exponential Smoothing Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. • Trend: Displays source data along with predicted values for the given dataset. • Forecast: Displays forecasted values for the given time period.
Target Variable	Select the target column for which you want to perform time series analysis.
Period	Select the period for forecasting.
Periods Per Year	Select the period for forecasting. This option is only enabled if you select "Custom" for "Period".
Start Year	Enter the year from which the observations must be considered. For example, 2009, 1987, 2019.
Start Period	Enter the period from which the observations must be considered.
Periods to Predict	Enter the number of periods to forecast. This value is used only if the output mode is Forecast.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Year Values	Enter a name for the newly created column that contains year values.
Quarter Values	Enter a name for the newly created column that contains quarter values.
Month Values	Enter a name for the newly created column that contains month values.
Period Values	Enter a name for the newly created column that contains period values.
Alpha	Enter a smoothing constant for smoothing observations (base parameters). Range: 0-1.
Beta	Enter a smoothing constant for finding trend parameters. Range: 0-1.

Property	Description
Gamma	Enter a smoothing constant for finding seasonal trend parameters. Range: 0-1.
Seasonal	Select the type of HoltWinters Exponential Smoothing algorithm.
Confidence Level	Enter the confidence level of the algorithm.
No. Periodic Observations	Enter the number of periodic observations required to start the calculation.
Level	Enter the start value for level (a[0]) (l.start). For example: 0.4.
Trend	Enter the start value for finding trend parameters (b[0]) (b.start). For example: 0.4.
Season	Enter start values for finding seasonal parameters (s.start). This value is dependent on the column you select. For example, if you select quarter as period, you need to provide four double values.
Optimizer Inputs	Enter the starting values for alpha, beta, and gamma required for the optimizer. For example: 0.3, 0.1, 0.1.

15.1.9.5 R-Single Exponential Smoothing

Properties that can be configured for the R-Single Exponential Smoothing algorithm.

Syntax

Use this algorithm to smooth the source data.

Note

Creating models using the R-Single Exponential Smoothing algorithm is not supported.

R-Single Exponential Smoothing Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. • Trend: Displays source data along with predicted values for the given dataset. • Forecast: Displays forecasted values for the given time period.

Property	Description
Target Variable	Select the target column for which you want to perform time series analysis.
Period	Select the period for forecasting.
Periods Per Year	Select the period for forecasting. This option is only enabled if you select "Custom" for "Period".
Start Year	Enter the year from which the observations must be considered. For example, 2009, 1987, 2019.
Start Period	Enter the period from which the observations must be considered.
Periods to Predict	Enter the number of periods to predict.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Year Values	Enter a name for the newly created column that contains year values.
Quarter Values	Enter a name for the newly created column that contains quarter values.
Month Values	Enter a name for the newly created column that contains month values.
Period Values	Enter a name for the newly created column that contains period values.
Alpha	Enter a smoothing constant for smoothing observations (base parameters). The default value is 0.3. Range: 0-1.
Confidence Level	Enter the confidence level of the algorithm.
No. Periodic Observations	Enter the number of periodic observations required to start the calculation. The default value is 2.
Level	Enter the start value for level (a[0]) (l.start). For example: 0.4.

15.1.9.6 R-Double Exponential Smoothing

Properties that can be configured for the R-Double Exponential Smoothing algorithm.

Syntax

Use this algorithm to smooth the source data and find trends in data.

Note

Creating models using the R-Double Exponential Smoothing algorithm is not supported.

R-Double Exponential Smoothing Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. • Trend: Displays source data along with predicted values for the given dataset. • Forecast: Displays forecasted values for the given time period.
Target Variable	Select the target column for which you want to perform time series analysis.
Period	Select the period for forecasting.
Periods Per Year	Select the periods for forecasting. This option is only enabled if you select "Custom" for "Period".
Start Year	Enter the year from which the observations must be considered. For example, 2009, 1987, 2019.
Start Period	Enter the period from which the observations must be considered.
Periods to Predict	Enter the number of periods to predict.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Year Values	Enter a name for the newly created column that contains year values.
Quarter Values	Enter a name for the newly created column that contains quarter values.
Month Values	Enter a name for the newly created column that contains month values.
Period Values	Enter a name for the newly created column that contains period values.
Alpha	Enter a smoothing constant for smoothing observations (base parameters). The default value is 0.3. Range: 0-1.
Beta	Enter a smoothing constant for finding trend parameters.The default value is 0.1. Range: 0-1.
Confidence Level	Enter the confidence level of the algorithm.
No. Periodic Observations	Enter the number of periodic observations required to start the calculation. The default value is 2.
Level	Enter the start value for level (a[0]) (l.start). For example: 0.4.
Trend	Enter the start value for finding trend parameters (b[0]) (b.start). For example: 0.4.
Optimizer Inputs	Enter the starting values for alpha, beta, and gamma required for the optimizer. For example: 0.3, 0.1, 0.1.

15.1.9.7 R-Triple Exponential Smoothing

Properties that can be configured for the R-Triple Exponential Smoothing algorithm.

Syntax

Use this algorithm to smooth source data and find seasonal trends in data.

Note

Creating models using the R-Triple Exponential Smoothing algorithm is not supported.

R-Triple Exponential Smoothing Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. • Trend: Displays source data along with predicted values for the given dataset. • Forecast: Displays forecasted values for the given time period.
Target Variable	Select the target column for which you want to perform time series analysis.
Period	Select the period for forecasting.
Periods Per Year	Select the period for forecasting. This option is only enabled if you select "Custom" for "Period".
Start Year	Enter the year from which the observations must be considered. For example, 2009, 1987, 2019.
Start Period	Enter the period from which the observations must be considered.
Periods to Predict	Enter the number of periods to predict.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Year Values	Enter a name for the newly created column that contains year values.
Quarter Values	Enter a name for the newly created column that contains quarter values.
Month Values	Enter a name for the newly created column that contains month values.
Period Values	Enter a name for the newly created column that contains period values.

Property	Description
Alpha	Enter a smoothing constant for smoothing observations (base parameters). The default value is 0.3. Range: 0-1.
Beta	Enter a smoothing constant for finding trend parameters. The default value is 0.1. Range: 0-1.
Gamma	Enter a smoothing constant for finding seasonal trend parameters. The default value is 0.1.
Seasonal	Select the type of HoltWinters Exponential Smoothing algorithm.
Confidence Level	Enter the confidence level of the algorithm.
No. Periodic Observations	Enter the number of periodic observations required to start the calculation. The default value is 2.
Level	Enter the start value for level (a[0]) (l.start). For example: 0.4.
Trend	Enter the start value for finding trend parameters (b[0]) (b.start). For example: 0.4.
Season	Enter start values for finding seasonal parameters (s.start). This value is dependent on the column you select. For example, if you select quarter as period, you need to provide four double values.
Optimizer Inputs	Enter the starting values for alpha, beta, and gamma required for the optimizer. For example: 0.3, 0.1, 0.1.

15.1.9.8 Triple Exponential Smoothing

Properties that can be configured for the Triple Exponential Smoothing algorithm.

Syntax

Use this algorithm to smooth the source data and find seasonal trends in data.

Triple Exponential Smoothing Properties

Property	Description
Output Mode	Select the mode in which you want to use the output of this algorithm. • Trend: Displays source data along with predicted values for the given dataset. • Forecast: Displays forecasted values for the given time period.

Property	Description
Target Variable	Select the target column for which you want to perform time series analysis.
Consider Date Column	Select this option to specify whether to use the date column.
Date Column	Enter the name of the column that contains date values.
Period	Select the period for forecasting.
Periods Per Year	Select the periods for forecasting. This option is only enabled if you select "Custom" for "Period".
Start Year	Enter the year from which the observations must be considered. For example, 2009, 1987, 2019.
Start Period	Enter the period from which the observations must be considered.
Periods to Predict	Enter the number of periods to predict.
Predicted Column Name	Enter a name for the newly created column that contains the predicted values.
Year Values	Enter a name for the newly created column that contains year values.
Quarter Values	Enter a name for the newly created column that contains quarter values.
Month Values	Enter a name for the newly created column that contains month values.
Period Values	Enter a name for the newly created column that contains period values.
Alpha	Enter a smoothing constant for smoothing observations (base parameters). The default value is 0.3. Range: 0-1.
Beta	Enter a smoothing constant for finding trend parameters. The default value is 0.1. Range: 0-1.
Gamma	Enter a smoothing constant for finding seasonal trend parameters. The default value is 0.1. Range: 0-1.

15.2 Data Preparation Components

Data preparation components prepare your data for analysis. These are optional components.

Use the links below to access more information about the components.

Related Information

Data Type Definition [page 246]

Filter [page 246]
Formula [page 251]
Model Compare [page 256]
Model Statistics [page 258]
Normalization [page 263]
Partition [page 266]
Sample [page 267]
HANA Binning [page 261]
HANA Data Type Definition [page 269]

15.2.1 Data Type Definition

Properties can be configured for the Data Type Transformation component.

Syntax

Use this component to change the name, data type, and date format of the source column.

Defining the data type helps you to prepare data to make it suitable for further analysis.

For example:

- If the name of the column in the data source is "des", it may not be clear during analysis. You can change the name of the column to "Designation" in the analysis, so that the end users can easily understand it.
- If the date is stored in the mmddyy (120201, without any date separator) format, it may be considered as an integer value by the system. Using the Data Type Definition component, you can change the date format to any valid format such as mm/dd/yyyy, or dd/mm/yyyy, and so on.

To change the name, data type, and the date format of the source column, perform the following steps:

1. Add the data type definition component into the analysis.
2. From the component's contextual menu, choose *Configure Properties*.
3. To change the column name, enter an alias name for the required source column.
4. To change the data type of the column, select the required data type for the source column.
5. Choose *Done*.

15.2.2 Filter

Properties that can be configured for the Filter Preparation Component.

Syntax

Use this component to filter rows and columns based on a specified condition.

Note

The In-DB Filter component does not support functions and advanced expressions.

Note

If you change the data source after configuring the filter component, the filter component still retains the previously defined row filters.

Filter Properties

Property	Description
Selected Columns	Select columns for analysis.
Filter Condition	Enter the filter condition.

Example

Filter "Store" column from the source data and apply "Profit >2000" condition.

Store	Revenue	Profit
Land Mark	10000	1000
Spencer	20000	4500
Soch	25000	8000

1. Uncheck the "Store" column from the Selected Columns.
2. In the *Row Filter* pane, choose the Profit column.
3. In the *Select from Range* option, enter 2000 in the *From* text box. The *To* text box should be empty.
4. Choose *OK*.
5. Choose *Save and Close*.
6. Execute the analysis.

Output table:

Revenue	Profit
20000	4500
25000	8000

Syntax

Note

The Filter component only supports expressions that return Boolean result.

For example, in the Employee table below:

Emp ID	Emp Name	DOB	Age	Date of Joining	Date of Confirmation
1	Laura	11/11/1986	25	12/9/2005	27/11/2005
2	Desy	12/5/1981	30	24/6/2000	10/7/2000
3	Alex	30/5/1978	33	10/10/1998	24/10/1998
4	John	6/6/1979	32	2/12/1999	20/12/1999

- The expression DAYSBETWEEN([Date of Joining],[Date of Confirmation]) is not a valid filter expression since it returns a numerical value. The correct usage of the DAYSBETWEEN expression in filter is DAYSBETWEEN([Date of Joining],[Date of Confirmation]) == 14. This expression selects those rows where number of days between "Date of Joining" and "Date of Confirmation" is 14. For the employee table above, the third row is selected.
- DAYNAME([Date of Joining]) == 'Saturday' selects the second and third rows in the employee table.

Note

When entering a string literal that contains single quotation marks, each single quotation mark inside the string literal must be escaped with a backslash character. For example, enter 'Customer's' as 'Customer \'s'.

Note

When entering a column name that contains square brackets, each square bracket inside the column name must be escaped with a backslash character. For example, enter [Customer[Age]] as [Customer\[Age\]].

Supported Functions

Note

The Filter component does not support data manipulation functions.

Category	Function (Function when applied on the Employee table)	Description
Date	DAYSBETWEEN	Returns the number of days between two dates.
	CURRENTDATE	Returns the current system date.
	MONTHSBETWEEN	Returns the number of months between two dates. For example, the new column contains 2,0,2,0 when MONTHSBETWEEN([Date of Joining],

Category	Function (Function when applied on the Employee table)	Description
		[Date of Confirmation]) is applied to the Employee table.
	DAYNAME	Returns the day name in the string format. For example, the new column contains Monday, Saturday, Saturday, Thursday when DAYNAME([Date of Joining]) is applied on the Employee table.
	DAYNUMBEROFMONTH	Returns the day number of the particular month. For example, 12/11/1980 returns 12.
	DAYNUMBEROFWEEK	Returns the day number in a week. For example, Sunday =1, Monday=2.
	DAYNUMBEROFYEAR	Returns the day number in a year. For example, 1st Jan =1, 1st Feb=32, 3rd Feb=34.
	LASTDATEOFWEEK	Returns the date of the last day in a week. For example, 12/9/2005 returns 17/9/2005
	LASTDATEOFMONTH	Returns the date of the last day in a month. For example, 12/9/2005 returns 30/9/2005
	MONTHNUMBEROFYEAR	Returns the month number in a date. For example, Jan=1, Feb=2, Mar=3
	WEEKNUMBEROFYEAR	Returns the week number in a year. For example, 12/9/2005 returns 38.
	QUARTERNUMBEROFDATE	Returns the quarter number in a date. For example, 12/9/2005 returns 3.
String	CONCAT	Concatenates two strings. For example, CONCAT('USA', 'Australia') returns USAAustralia.
	INSTRING	Returns true - if the search string is found in the source string.

Category	Function (Function when applied on the Employee table)	Description
		For example, INSTRING('USA', 'US') returns true.
	SUBSTRING	Returns a substring from the source string. For example, SUBSTRING('USA', 1,2) returns US.
Math	MAX	Returns the maximum value in a column.
	MIN	Returns the minimum value in a column.
	COUNT	Returns the number of values in a column.
	SUM	Returns the sum of the values in a column.
	AVERAGE	Returns the average of the values in a column.
Conditional Expression	IF(condition) THEN(string expression/ mathematical expression/conditional expression) ELSE(string expression/ mathematical expression/conditional expression)	Checks whether the condition is met, and returns one value if 'true' and another value if 'false'. For example, IF([Date of Joining]>12/9/2005) THEN ('Employee joined after Sept 12, 2005') ELSE ('Employee joined on or before Sept 12, 2005')

> **Note**
>
> Mathematical expressions containing functions that return a numerical value are not supported. For example, expression DAYNUMBEROFMONTH(CURRENTDATE())==2 is not supported because DAYNUMBEROFMONTH returns a numerical value.

Mathematical Operators

Use mathematical operators to create formulas containing numerical columns and/or numbers. For example, the expression `[Age] + 1` adds a new column with the values 26, 31, 34, 33.

Mathematical Operators	Description
+	Addition operator
-	Subtraction operator

Mathematical Operators	Description
*	Multiplication operator
/	Division operator
()	Round brackets or parenthesis
^	Power operator
%	Modulo operator
E	Exponential operator

Conditional Operators

Use conditional operators to create IF THEN ELSE or SELECT expressions.

Conditional Operators	Description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Logical Operators

Use logical operators to compare two conditions and return 'true' or 'false'. For example, IF([Date of Joining]>12/9/2005 && [Age] >=25) THEN ('True') ELSE ('False') adds a new column with values True, False, False, False.

Logical Operators	Description
&&	AND
||	OR

15.2.3 Formula

Properties that can be configured for the Formula Preparation Component.

Syntax

Use this component to apply predefined functions and operators on the data. All functions and expressions except data manipulation functions add a new column with the formula result.

Note

When entering a string literal that contains single quotation marks, each single quotation mark inside the string literal must be escaped with a backslash character. For example, enter 'Customer's' as 'Customer \'s'.

Note

When entering a column name that contains square brackets, each square bracket inside the column name must be escaped with a backslash character. For example, enter [Customer[Age]] as [Customer\[Age\]].

Formula Properties

Property	Description
Formula Name	Enter a name for the new column created by applying the formula.
Expression	Enter the formula you want to apply. For example, Average([Age]).

Example

Calculating average age of employees

Employee Table:

Emp ID	Emp Name	DOB	Age	Date of Joining	Date of Confirmation
1	Laura	11/11/1986	25	12/9/2005	27/11/2005
2	Desy	12/5/1981	30	24/6/2000	10/7/2000
3	Alex	30/5/1978	33	10/10/1998	24/12/1998
4	John	6/6/1979	32	2/12/1999	20/12/1999

To calculate average age of employees, perform the following steps:

1. Drag the *Formula* component onto the analysis editor.
2. In the properties view, enter a name for the formula.
   For example, Average_Age.
3. In the *Expression* field, enter the formula: AVERAGE([Age])
4. Choose *Validate* to validate the formula syntax.
5. Choose *Done*.

Output table:

Emp ID	Emp Name	DOB	Age	Date of Joining	Date of Confirmation	Average_Age
1	Laura	11/11/1986	25	12/9/2005	27/11/2005	30
2	Desy	12/5/1981	30	24/6/2000	10/7/2000	30
3	Alex	30/5/1978	33	10/10/1998	24/12/1998	30
4	John	6/6/1979	32	2/12/1999	20/12/1999	30

Supported Functions

Category	Function (Function when applied on the Employee table)	Description
Date	DAYSBETWEEN	Returns the number of days between two dates.
	CURRENTDATE	Returns the current system date.
	MONTHSBETWEEN	Returns the number of months between two dates. For example, the new column contains 2,0,2,0 when MONTHSBETWEEN([Date of Joining], [Date of Confirmation]) is applied to the Employee table.
	DAYNAME	Returns the day name in string format. For example, the new column contains Monday, Saturday, Saturday, Thursday when DAYNAME([Date of Joining]) is applied to the Employee table.
	DAYNUMBEROFMONTH	Returns the day number of the particular month. For example, 12/11/1980 returns 12.
	DAYNUMBEROFWEEK	Returns the day number in a week. For example, Sunday =1, Monday=2.
	DAYNUMBEROFYEAR	Returns the day number in a year. For example, 1st Jan =1, 1st Feb=32, 3rd Feb=34.
	LASTDATEOFWEEK	Returns the date of the last day in a week. For example, 12/9/2005 returns 17/9/2005

Category	Function (Function when applied on the Employee table)	Description
	LASTDATEOFMONTH	Returns the date of the last day in a month. For example, 12/9/2005 returns 30/9/2005
	MONTHNUMBEROFYEAR	Returns the month number in a date. For example, Jan=1, Feb=2, Mar=3
	WEEKNUMBEROFYEAR	Returns the week number in a year. For example, 12/9/2005 returns 38.
	QUARTERNUMBEROFDATE	Returns the quarter number in a date. For example, 12/9/2005 returns 3.
String	CONCAT	Concatenates two strings. For example, CONCAT('USA', 'Australia') returns USAAustralia.
	INSTRING	Returns true - if the search string is found in the source string. For example, INSTRING('USA', 'US') returns true.
	SUBSTRING	Returns a substring from the source string. For example, SUBSTRING('USA', 1,2) returns US.
	STRLEN	Returns the number of characters in the source string. For example, STRLEN('Australia') returns 9.
Math	MAX	Returns the maximum value in a column.
	MIN	Returns the minimum value in a column.
	COUNT	Returns the number of values in a column.
	SUM	Returns the sum of the values in a column.
	AVERAGE	Returns the average of the values in a column.
Data Manipulation	@REPLACE	Performs in-place replacement of a string. For example, @REPLACE([country],'USA',

Category	Function (Function when applied on the Employee table)	Description
		'AMERICA') replaces USA with AMERICA in the country column.
	@BLANK	Replaces blank values with a specified value. For example, @BLANK([country], 'USA') replaces all blank values with USA in the country column.
	@SELECT	Selects rows that satisfy the given condition. You can use any conditional operator to specify the condition. For example, @SELECT([country]=='USA') selects rows where country is equal to USA.
Conditional Expression	IF(condition) THEN(string expression/ mathematical expression/conditional expression) ELSE(string expression/ mathematical expression/conditional expression)	Checks whether the condition is met, and returns one value if 'true' and another value if 'false'. For example, IF([Date of Joining]>12/9/2005) THEN ('Employee joined after Sept 12, 2005') ELSE ('Employee joined on or before Sept 12, 2005')

> **Note**
>
> Mathematical expressions containing functions that return a numerical value are not supported. For example, expression DAYNUMBEROFMONTH(CURRENTDATE())+2 is not supported because DAYNUMBEROFMONTH returns a numerical value.

Mathematical Operators

Use mathematical operators to create formulas containing numerical columns and/or numbers. For example, the expression `[Age] + 1` adds a new column with values 26, 31, 34, 33.

Mathematical Operators	Description
+	Addition operator
-	Subtraction operator
*	Multiplication operator
/	Division operator
()	Round brackets or parenthesis

Mathematical Operators	Description
^	Power operator
%	Modulo operator
E	Exponential operator

Conditional Operators

Use conditional operators to create IF THEN ELSE or SELECT expressions.

Conditional Operators	Description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Logical Operators

Use logical operators to compare two conditions and return 'true' or 'false'. For example, IF([Date of Joining]>12/9/2005 && [Age] >=25) THEN ('True') ELSE ('False') adds a new column with values True, False, False, False.

Logical Operators	Description
&&	AND
||	OR

15.2.4 Model Compare

Use the Model Compare component with the Model Statistics component to learn the best algorithm for your predictive problem in all scenarios (HANA and non-HANA).

Comparing Models

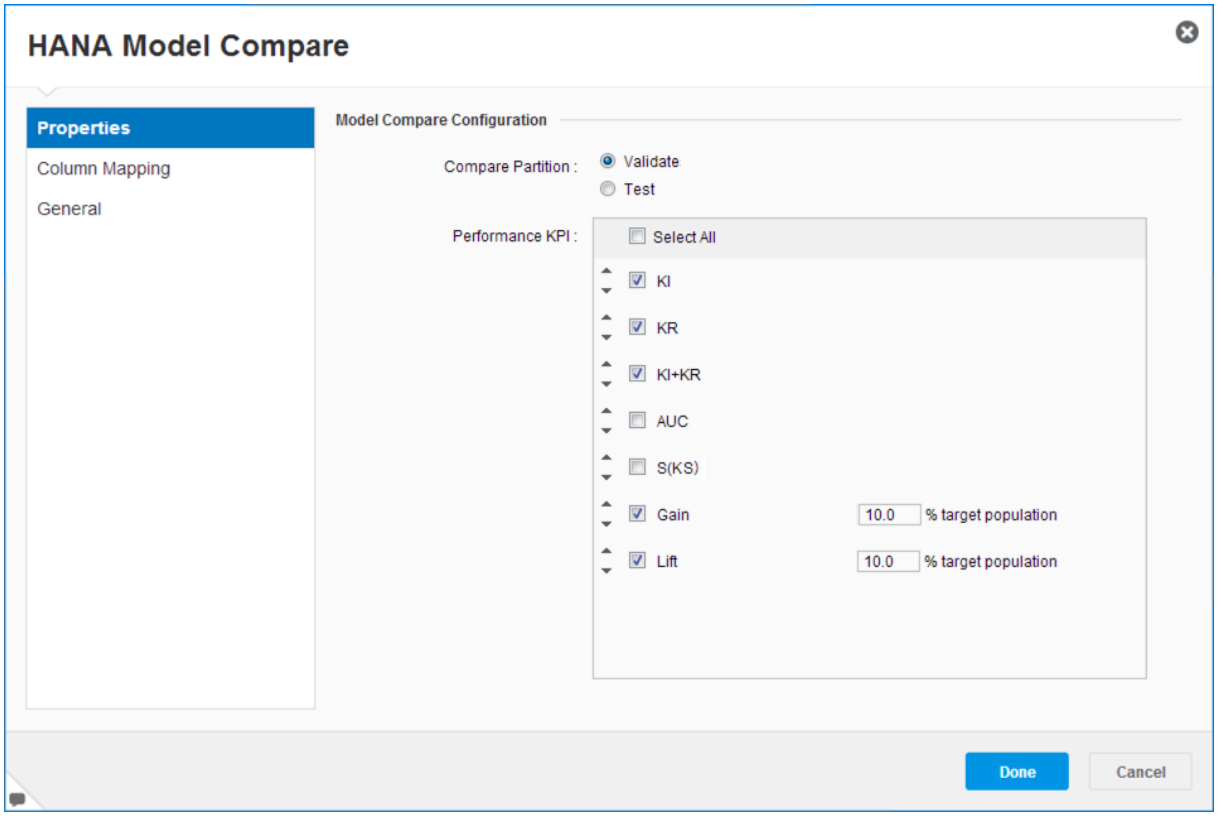
Expert Analytics can compare the performance of two or more algorithms in an analysis and indicate the best one with the Model Compare component. You use first the Model Statistics component to calculate

performance statistics for either classification or regression algorithm types. After which, the Model Compare component compares the calculated performance statistics to pick the best algorithm of those run at execution. In addition, the Model Compare component merges the results to provide a detailed summary on the best performing component.

Configuring Partitions and KPIs

You can configure partition types and KPIs in the Model Compare component for more control over your analysis chain. In the Properties Panel of the component, you can select either a Validate or Test partition to compare the performance of the connected models. Also, you can choose the required KPIs and sort the order in which they should be compared. Control over the order is important because if the top KPI cannot identify a winning algorithm, the component can perform calculations with the second KPI in the list, and so on. In addition, a precise percentage can be configured for the Gain and Lift parameters. The result is an even more accurate calculation when comparing two or more components.

The below image is of the Column Mapping panel of the Model Compare component in which you can configure the Partition and the KPIs (using the English language version as an example):



Adding Child Components

You can add child components to Model Compare. The best scenario in which to use the feature is with two parent components. With two components connected for comparison, the results mapping section becomes

enabled. From there, you can define how to manage the results from the two components. In a two-component compare, the Model Compare component displays the following icon::



When you can compare more than three or more components at a time, the Model Compare component becomes a terminal (or leaf) component. This means that you cannot add a child node to perform further analysis When comparing three or more components, the Model Compare component displays the following icon:



15.2.5 Model Statistics

Use the Model Statistics component to generate performance statistics to solve two-class problems for all scenarios (HANA and non-HANA). Visualize and share results in a range of charts. Use the component with the Model Compare component to compare two or more models and discover the best one for a predictive problem.

Calculate Performance Statistics

Model Statistics is a component that calculates performance statistics on datasets that are generated by algorithms. It can do so for two algorithm types, classification and regression. In addition, you can configure the component to generate performance statistics for Train, Validate and Test datasets and selected KPIs.

Two-Class Problems

The component works only with two-class problems. A two-class problem is a business problem with a binary outcome, which means that it classifies the elements of a given dataset into two groups by a classification rule.

One example is in churn modeling for a business with a subscription service. In such a case, the two-class problem is to identify subscribers who will stay with the service, and those who will leave.

Another example is fraud detection at a financial institution, where the two-class problem is to identify which transactions are fraudulent, and which are not.

How To Ensure a Strong Predictive Quality (KI)

You must ensure that the predictive quality (Ki) of the model is strong. For example, if the Ki is zero, it means that the model is not trained well and inspires no confidence, since it is equivalent essentially to a random model.

The Ki is directly linked to the amount of information available to predict the target. Therefore, you can improve the KI by increasing the number of useful variables in the model in the following ways:

- Use all variables available.
- Use your domain knowledge to find other sources of information.
- Build variables from the existing ones with data manipulations.
- Use combination of variables by increasing the polynomial degree.

Charts in Model Statistics

You can generate and share charts for classification and regression algorithms in the Model Statistics component. The charts visualize the performance of Classification and Regression algorithms.

Classification charts:

- **Gain(Profit)**: Visualizes the gain or profit that is realized by the model based on a percentage of the target population selection. On the chart the y-axis shows Gain/profit and x-axis shows the Percentage.
- **Lift**: Visualizes the amount of lift the trained model gives as compared to a random model. It allows examination of the difference between a perfect model, a random model and the model created. On the chart the y-axis shows Lift profit and x-axis displays the Percentage.
- **Standardized (KS)**: Visualizes the distance between the distribution functions of the two classes in binary classification (for example, Class 1 and Class 0). The score that generates the greatest separability between the functions is considered the threshold value for accepting or rejecting the target. The measure of seperability defines how well the model is able to distinguish between the records of two classes. If there are minor deviations in the input data, the model should still be able to identify these patterns and diiferentiate between the two. In this way, seperability is a metric of how good the model is; the greater the seperability, the greater the model. Note that the predictive model producing the greatest amount of separability between the two distributions is considered the superior model.
- **Receiving Operating Characteristic (ROC)**: Visualizes the ROC curve, which is generated by plotting the true positive rate (or sensitivity) at various threshold settings against the false positive rate (or the fall-out; calculated as 1 - specificity). The ROC curve is used to derive the metric, Area Under the Curve (AUC). On the chart, the y-axis shows Sensitivity, and X-axis displays Specificity.

Regression chart:

- **Model Accuracy**: Visualizes how many records were correctly predicted in comparison to the actual target values.

Interaction with the Model Compare Component

You can use the Model Statistics component with the Model Compare component to learn the best algorithm for your predictive problem. First the Model Statistics component calculates the performance statistics for either classification or regression algorithm types. After which, the Model Compare component compares the calculated performance statistics to pick the best algorithm of those run at execution.

Note that when you change configurations in the Model Statistics component, it affects the Model Compare component.

In rendering the charts when interacting with Model Compare, the Model Statistics component overlays the partitions atop each and displays different results per partition. The Model Compare component does the same because both components use the same data. Therefore, you should ensure that you configure the KPIs for both exactly the same.

Interaction with the Partition Component

When the Partition component is included before the Model Statistics component in an analysis chain, you receive the option to use three different partitions: Train, Test and Validate. If the Partition component is not included, the Model Statistics component displays a set of statistics and charts for the Train partition only.

15.2.5.1 Generating Performance Statistics

Generate performance statistics for classification and regression algorithms in the Model Statistics component. Use charts in the Model Statistics component to visualize model performance results for Train, Validate and Test datasets and selected KPIs.

To configure the Model Statistics component to generate peformance statistics and visualize results in the Model Statistics component, take the following steps:

1. Launch Expert Analytics, connect to a dataset, and navigate to the *Predict Room*.
2. From the Component List add the *Partition* component.

   > **Note**
   >
   > You must add a Partition component if you want to see the charts when using Model Statistics. The purpose of the charts is to display the curves for different partition datasets.

3. Double-click the *Partition* component, configure the required data fields for the Train, Validate and Test datasets, and click *Done*.
4. In the Algorithms section, drag-and-drop selected algorithms to the analysis editor, and configure them. For example, if solving a classification problem, you might choose three classification algorithms, *Auto Classification*, *R-CNR Tree*, and *Naïve Bayes*.
5. From the Component List add a *Model Statistics* component to the analysis editor for the appropriate algorithm, regression or classifiction.
6. Double-click the *Model Statistics* component to open the configuration options.
7. Click the *Properties* tab to configure the appropriate algorithm type, set the *Target Column* on which to run the algorithm, and then set the *Predicted Column*. Optionally, click the *General* tab and add a component *Alias* and *Description*. Click *Done*.
8. Click the *Run Analysis*  icon.
9. Select the *Results* tab of Model Statistics component to see a summary of the results.
10. Optionally, view the data in the following chart formats:
    a. *Gain/profit*: Y-axis shows Gain/profit and X-axis shows the percentage.

b. *Standardized (KS)*: Y-axis shows Standardized profit and X-axis shows the percentage.
c. *Lift*: Y-axis shows Lift profit and X-axis shows the percentage.
d. *ROC*: Y-axis shows Sensitivity and X-axis shows specificity.

> **Note**
>
> For each chart, you can view Train, Validate and Test curves overlapping on the same chart. There is one curve for each partition and for each chart.

You have generated performance statistics in summary and chart format for the required alogorithm.

15.2.5.1.1 HANA Binning

Properties that can be configured for the Binning Preparation Component in HANA scenarios.

> **Syntax**
>
> Binning also known as discretization, smooths a sorted data value. It divides the range of a numerical variable into sets of subranges called bins, and replaces each value with its bin number. Binning data before running certain algorithms, such as the decision tree algorithm, helps reduce the complexity of the model.
>
> There are four binning methods:
>
> - Equal widths based on number of bins
> - Equal widths based on bin width
> - Equal depth
> - Deviation from mean
>
> Also, there are three methods for smoothing:
>
> - Smoothing by bin means: each value in a bin is replaced by bin value of the mean.
> - Smoothing by bin medians: each bin value is replaced by the bin median.
> - Smoothing by bin boundaries: the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by its closest boundary value.

HANA Binning properties

Property	Description
Independent Column	Select the input source column on which you want to perform binning.

Property	Description
Missing values	Select the method for handling missing values. Possible methods: • Ignore: The algorithm skips the records containing missing values in the independent or dependent columns. • Keep: Retains missing values.
Binning method	Select the *Binning Method*.
Number of Bins	Enter the number of bins needed.
Smoothing Method	Select the *Smoothing Method*.
Binned Column Name	Enter a name for the new column that contains bin numbers.
Smoothed Values Column Names	Enter the name for the new column that contains smoothed values.

Example

Binning of data in a dataset

City	Temperature
Amsterdam	6
Frankfurt	12
Guangzhou	13
Cape Town	15
Waldorf	10
Bangalore	23
Mumbai	24
Miami	30
Rio De Janeiro	32
Sydney	25
Dubai	38

To bin the Temperature column by equal widths based on the number of widths and apply smoothing methods by means, perform the following steps:

1. Drag the *Binning* component onto the analysis editor.
2. Double click *Binning*, or hover the mouse on *Binning* and choose *Configure Properties*.
3. In the *Independent Column* drop down list, select a column, for example, *Temperature*.

Note

You can only select columns that have numerical digit values.

4. In *Missing values* drop down list, choose *Ignore*.
5. In *Binning Method*, choose *Equal widths based on the number of bins*.

6. In number of bins, enter 4.
7. Select *Smoothing Required*.
8. In Smoothing methods, choose *Bin Mean*.
9. Under Enter name for newly added column, in *Binned Column Name*, enter Temperature Bin.

   > **Note**
   >
   > You can name the column based on your preference or analysis requirement. This column contains the binned value.

10. Under Enter name for newly added column, in *Smoothed Values Column Names*, enter Temperature Smooth.

    > **Note**
    >
    > You can name the column based on your preference or analysis requirement. This column contains the smoothed value.

Output Table:

City	Temperature	Temperature Bin	Temperature Smooth
Amsterdam	6	1	8.0
Frankfurt	12	2	13.33333
Guangzhou	13	2	13.33333
Cape Town	15	2	13.33333
Waldorf	10	1	8.0
Bangalore	23	3	25.5
Mumbai	24	3	25.5
Miami	30	3	25.5
Rio De Janeiro	32	4	35.0
Sydney	25	3	25.5
Dubai	38	4	35.0

15.2.6 Normalization

You can configure properties for the Normalization Preparation Component in HANA and non-HANA scenarios.

> **Syntax**
>
> Use this component to normalize the attribute data. HANA Normalization scales the large value attribute data to fall within a specific range, such as -1.0 to 1.0, or 0.0 to 1.0. You can use this component for In-Database analysis. Normalization of data is useful for classification algorithms involving neural networks, or distance measurements such as nearest neighbor classification and clustering.

Note

If you want the processed data to replace the existing column, select *Replace column*.

The normalization component supports the following normalization methods:

- *Min-Max normalization*: Performs a linear transformation on the original data values, and scales each value to fit in a specific range. While performing the Min-Max normalization you can specify *New Maximum value* and *New Minimum value.* This normalization is helpful for ensuring that extreme values are constrained within a fixed range.

  ### Note

  - *New Maximum value* must be greater than *New Minimum value.*

- *Z-score normalization*: Computed based on the `mean` and `standard deviation` for each attribute. This normalization is useful to determine whether a specific value is above or below average, and by how much.
- *Decimal scaling normalization*: The decimal point of the values of each attribute are moved according to its maximum absolute value.

Note

You can select *Replace column*, if you want the normalized data to replace the existing column data, on which normalization is performed.

Example

Normalizing the time taken to cover a certain distance.

Table:

Name	Distance (in meters)	Time (in seconds)
Laura	500	66
Desy	500	360
Alex	500	201
John	500	78
Ted	500	504

To normalize the time column using *Min-Max* normalization, perform the following steps:

1. In the *Predict view*, from the *Component List* choose *Data Preparation* tab.
2. Drag the *HANA Normalization* component onto the analysis editor or Double-click on *HANA Normalization*.
3. Double click *HANA Normalization* , or hover the mouse pointer on *HANA Normalization* and choose *Configure Properties.*
4. Select the columns you want to normalize.

   ### Note

   You can only select columns with numerical values.

For example, Time (in seconds).

5. From Normalization Type drop down, choose *Min-Max*.
6. Enter values for the *New Maximum* and the *New Minimum*.
7. Choose *Done*, and then choose *Run*.

Output table:

Name	Distance (in meters)	Time (in seconds)	Time (in seconds)_Normalized
Laura	500	66	0.05
Desy	500	360	0.30
Alex	500	201	0.17
John	500	78	0.06
Ted	500	504	0.42

Perform same steps for *Z-score* normalization and *Decimal Scaling* normalization as mentioned in *Min-Max* normalization. However, in case of *Z-score* normalization and *Decimal Scaling* normalization, you do not have enter the *New Maximum* and the *New Minimum* value.

Z-score normalization output:

Output table:

Name	Distance (in meters)	Time (in seconds)
Laura	500	-0.49
Desy	500	1.77
Alex	500	0.55
John	500	-0.40
Ted	500	2.88

Decimal Scaling normalization output:

Output table:

Name	Distance (in meters)	Time (in seconds)
Laura	500	0.01
Desy	500	0.04
Alex	500	0.02
John	500	0.01
Ted	500	0.05

15.2.7 Partition

You can configure properties for the Partition component in HANA and non-HANA scenarios.

Syntax

The Partition component partitions an input dataset randomly into three subsets called Train, Test, and Validate. The proportion of each subset is defined as a parameter. The union of three subsets need not be the complete initial dataset.

You can partition the dataset using the following partition methods:

- Random Partition, which randomly divides all the data.
- Stratified Partition, which divides each sub-category randomly.

In the second case, the dataset needs to have at least one categorical attribute (for example, of type varchar). The initial dataset is subdivided according to the different categorical values of this attribute. Each mutually exclusive subset is then randomly split to obtain the Train, Test, and Validate subsets. This ensures that all "categorical values" or "strata" are present in the sampled subset.

Note that when comparing two or more algorithms in the model comparison chain, the Partition component is mandatory.

Partition Properties

Property	Description
Partition Method	Select the method for partitioning data into train, test, and validation sets. • Random • Stratified
Random Seed	Enter a random number using which you want to perform the calculation.
Partition Rows by	Select the method for partitioning rows. • Percentage of Rows • Number of Rows
Train Set	Enter the number of rows or percentage of rows for the train set.
Test Set	Enter the number of rows or percentage of rows for the test set.
Validation Set	Enter the number of rows or percentage of rows for validation set.
Partition Column Name	Enter a name for the new column that contains partitioned values.
Number of Threads	Enter the number of threads the algorithm should use for execution.

15.2.8 Sample

Properties that can be configured for the Sample Preparation Component.

Syntax

Use this component to select a subset of data from large datasets.

The Sample component supports the following sample types:

- First N: Selects the first N records in the dataset.
- Last N: Selects the last N records in the dataset.
- Every Nth: Selects every Nth record in the dataset, where N is an interval. For example, if N=2, the 2nd, 4th, 6th, and 8th records are selected and so on.
- Simple Random: Randomly selects records of size N or N percent of records in a dataset.
- Systematic Random: In this sample type, sample intervals or buckets are created based on the bucket size. The Sample component selects the Nth record at random from the first bucket, and from each subsequent bucket the Nth record is selected.

Sample Properties

Property	Description
Sampling Type	Select the type of sampling.
Limit Rows by	Select the method for limiting the rows.
Number of Rows	Enter the number of rows you want to select.
Percentage of Rows	Enter the percentage of rows you want to select.
Bucket Size	Enter the bucket size within which you want to select a random row.
Step Size	Enter the interval between the rows you want to select.
Maximum Rows	Enter the maximum number of rows you want to select.

Example

Selecting subset of data from a given dataset

Emp ID	Emp Name	DOB	Age
1	Laura	11/11/1986	25
2	Desy	12/5/1981	30
3	Alex	30/5/1978	33
4	John	6/6/1979	32
5	Ted	4/7/1987	24
6	Tom	30/6/1970	41

Emp ID	Emp Name	DOB	Age
7	Anna	24/6/1965	46
8	Valerie	6/7/1990	21
9	Mary	19/9/1985	26
10	Martin	21/11/1986	25

Sample outputs:

1. First N: For N=5

Emp ID	Emp Name	DOB	Age
1	Laura	11/11/1986	25
2	Desy	12/5/1981	30
3	Alex	30/5/1978	33
4	John	6/6/1979	32
5	Ted	4/7/1987	24

2. Last N: For N=4

Emp ID	Emp Name	DOB	Age
7	Anna	24/6/1965	46
8	Valerie	6/7/1990	21
9	Mary	19/9/1985	26
10	Martin	21/11/1986	25

3. Every Nth: Interval=3

Emp ID	Emp Name	DOB	Age
3	Alex	30/5/1978	33
6	Tom	30/6/1970	41
9	Mary	19/9/1985	26

4. Simple Random: For number of rows=2
   The result can be any two rows.

Emp ID	Emp Name	DOB	Age
7	Anna	24/6/1965	46
8	Valerie	6/7/1990	21

5. Systematic Random: Bucket Size=4

Emp ID	Emp Name	DOB	Age
2	Desy	12/5/1981	30
6	Tom	30/6/1970	41

Emp ID	Emp Name	DOB	Age
10	Martin	21/11/1986	25

or

Emp ID	Emp Name	DOB	Age
1	Laura	11/11/1986	25
5	Ted	4/7/1987	24
9	Mary	19/9/1985	26

15.2.9 HANA Data Type Definition

Use the HANA Data Type Definition component to convert one data type to another.

Using the component, you can work with a wide range of SAP HANA data sets in Expert Analytics.

Example:

Suppose you want to build a model chain that uses a data set which contains the unsupported TINYINT data type. You can use the HANA Data Type Definition component to convert the column of TINYINT values to the supported INTEGER data type.

Step Guide:

Take the following steps to perform a data type conversion:

1. In Expert Analytics, connect to a data set on SAP HANA.
2. Go to the *Predict* room.
3. Drag *HANA Data Type Definition* from the component list to the canvas. You can find it in the Data Preparation section.
4. Launch the dialog box to configure the component.
5. In the default Transformations tab, configure the following settings:
   a. *Keep Original Column*: By default the checkbox is unselected which replaces the original column. Select the checkbox to keep the original column in the output data set. Note that you must add an alias name for a new column. (See Alias Name description below for more information.)
   b. *Column Name*: Select the column name that you want to convert. Ensure that all records in the column that you want to convert fit the respective selected output format.
   c. *Convert To*: Select the data type to which you want to convert the selected column. Ensure that your input data fits the selected output format. For example, a BIGINT value of size 3,000,000,000 will **fail** to convert into an INTEGER value, which has a maximum value of INTEGER is 2,147,483,647. However, a BIGINT value of 2,000,000,000 will convert to INTEGER.
   d. *Date Format*: Select the date format that is used in your input data set. This is the date format that represents the data that you want to translate into the required date in SAP HANA.
   e. *Alias Name*: If you want to keep the original column, you must enter an alias name. If you do not keep the original column, the setting is disabled. When the original column is kept, the component auto-generates a prefix to the column that is based on the data type to which the column is being converted. You can the edit the name; however, you must ensure that it does not match an existing column name on the selected data set.

f. *Add more columns*: Click the Add icon to add more columns to convert.
g. *Delete rule*: Click the Delete icon to remove the selected rule.

6. Click *Done*.

The data types are converted from one type to another. When you run the model chain, the HANA Data Type Definition component executes like any other component. These transformations can participate in both Retrain and Apply operations.

15.3 Data Writers

Use data writers to store the results of the analysis in flat files or databases for further analysis.

15.3.1 CSV Writer

Properties that can be configured for the CSV Writer.

Syntax

Use this component to write data to flat files such as CSV, TEXT, and DAT files.

CSV Writer Properties

Property	Description
File Name	Select the file path and enter a name for csv or dat or txt file.
Overwrite, if exists	To overwrite an existing file, select this option.
Column Separator	Select a column delimiter that separates data tokens in the file.
Insert Quotation Character	Select the character for replacing the column separators while writing the data.
Include Column Headers	Select this option to use the first row as column headers.
Encoding	Select the text-encoding method to write the data.
Decimal Separator	Select the character for decimal representation in digit grouping.
Grouping Separator	Select the character for the thousands separator.
Number Format	Enter the number format you want to apply to numerical data.
Date Time Format	Select the date format you want to apply to dates.

15.3.2 HANA Writer

Properties that can be configured for the HANA Writer.

Syntax

Use this component to write data to SAP HANA database tables.

HANA Writer Component

Property	Description
Schema Name	Select a schema.
Table Type	Select the table type of the table to which you want to write data.
Table Name	Enter a name for the table.
Overwrite, if exists	Select this option to overwrite the table if it already exists.

15.3.3 JDBC Writer

Properties that can be configured for the JDBC Writer.

Syntax

Use this component to write data to relational databases such as MySQL, MS SQL Server, DB2, Oracle, SAP MaxDB, and SAP HANA.

JDBC Writer Properties

Property	Description
Database Type	Select the database type.
Database Driver Path	Enter the location of the JDBC driver path. For example, to write to the Oracle database, you need to specify the location of the Oracle JDBC jar (`C:\ojdbc6.jar`)
Database Machine Name	Enter the name of the machine on which the database is installed.
Port Number	Enter the database or service port number.
Database Name	Enter the name of the database.

Property	Description
User Name	Enter the database user name.
Password	Enter the password for the database user.
Table Type	Enter the type of the table. This property is applicable when writing to the SAP HANA database.
Table Name	Enter the table name.
Overwrite, if exists	Select this option to overwrite the table if it already exists.

15.4 Models

Models that you create by saving the state of algorithms are listed under the *Models* section in the *Components* list.

Expert Analytics does not contain predefined models. Therefore, unless you have already saved a configured algorithm as a model, the *Models* section is empty.

Related Information

Creating a Model [page 153]

Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of willful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).



go.sap.com/registration/contact.html

© 2017 SAP SE or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.
Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.
These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.
Please see http://www.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.