



**PUBLIC**

SAP Predictive Analytics

2020-08-28

# Data Manager User Guides and Scenarios

# Content

- 1 About This Guide. . . . . 5**
- 2 Getting Started with Data Manipulation. . . . . 6**
  - 2.1 About Getting Started with Data Manipulation. . . . . 6
  - 2.2 Definition. . . . . 6
  - 2.3 Creating a Data Manipulation. . . . . 7
    - Selecting a Data Source. . . . . 7
    - Data Manipulation Editor. . . . . 11
  - 2.4 Saving a Data Manipulation. . . . . 41
  - 2.5 Using a Data Manipulation. . . . . 42
    - Using a Saved Data Manipulation. . . . . 42
    - Use Case Scenarios. . . . . 43
  - 2.6 Performing a Data Manipulation Transfer. . . . . 45
    - Creating a Data Manipulation. . . . . 45
    - Transferring a Data Manipulation. . . . . 46
    - Using the Transferred Data Manipulation in a New Database. . . . . 47
  - 2.7 Annex. . . . . 48
    - Arithmetic Operators. . . . . 48
    - Boolean Operators. . . . . 50
    - Date Operators. . . . . 53
    - Miscellaneous Operators. . . . . 55
    - String Operators. . . . . 55
    - Conversion Operators. . . . . 58
- 3 Data Manipulation Scenario. . . . . 60**
  - 3.1 About Data Manipulation Scenario. . . . . 60
  - 3.2 Essential Concepts. . . . . 60
    - Data Manager Semantic Layer. . . . . 60
    - Entity. . . . . 60
    - Time-stamped Population. . . . . 61
    - Analytical Record. . . . . 61
    - Analytical Dataset. . . . . 61
    - Temporal Analytical Dataset. . . . . 61
    - Performance Indicator (PI). . . . . 62
    - Methodology. . . . . 62
  - 3.3 Application Scenario: Segmented Cross-sell in Retail Banking. . . . . 63
    - Creating the Performance Indicator. . . . . 64

	Preparing the Data. . . . .	66
	Creating the First Classification Model. . . . .	74
	Model Results Analysis. . . . .	82
3.4	Entity Transfer. . . . .	83
	Transferring an Entity. . . . .	83
	Using the Transferred Entity in a New Database. . . . .	84
<b>4</b>	<b>Event Log Aggregation Scenario. . . . .</b>	<b>85</b>
4.1	About Event Log Aggregation Scenario. . . . .	85
4.2	Event logging: Description. . . . .	85
4.3	Use Scenario: Overview. . . . .	88
4.4	Use Scenario: Introduction. . . . .	91
4.5	Step 1 - Configuring the Data Source. . . . .	92
	Importing CSV Files into a Database. . . . .	93
4.6	Step 2 - Modeling your Data. . . . .	94
	Simple Method: Using Only Reference Data. . . . .	94
	Intermediate Method: Adding Demographic Data. . . . .	102
	Overall Method: Adding Transaction Data. . . . .	104
4.7	Step 3 - Making a Decision and Taking Action. . . . .	115
	Identifying the Customers to Contact. . . . .	115
	Your Marketing and Sales Campaign: Wrap-Up. . . . .	120
<b>5</b>	<b>Sequence Analysis Scenarios. . . . .</b>	<b>122</b>
5.1	Introduction to Application Scenarios. . . . .	122
5.2	Introduction to Sample Files. . . . .	123
5.3	Scenario 1: Segment Visitors to Understand Purchase Behavior Using File Counts. . . . .	123
	Step 1 - Selecting the Data. . . . .	124
	Step 2 - Defining the Modeling Parameters. . . . .	130
	Step 3 - Generating and Validating the Model. . . . .	137
	Step 4 - Analyzing and Understanding the Model. . . . .	138
5.4	Scenario 2: Predict End of Session Using Intermediate Sequences. . . . .	138
	Step 1 - Selecting the Data. . . . .	139
	Step 2 - Defining the Modeling Parameters. . . . .	139
	Step 3 - Generating and Validating the Model. . . . .	140
	Step 4 - Analyzing and Understanding the Model. . . . .	141
<b>6</b>	<b>Text Analysis Scenario. . . . .</b>	<b>143</b>
6.1	About Text Analysis Scenario. . . . .	143
	Before Beginning. . . . .	144
6.2	General Introduction to Scenario. . . . .	146
6.3	Extracting Information from Textual Data. . . . .	148
	Simple Method: Using a Classification Model on the Data. . . . .	148

	Intermediate Method: Adding Information with the Data Manipulations. . . . .	155
	Advanced Method: Using Text Coding to Extract Information from the Textual Variables. . . . .	157
	Advanced Method without Stop Words and Stemming Rules. . . . .	163
	Adapted Method: Defining a Specific Language for the Domain. . . . .	165
6.4	Annex - Regular Expression Reminder. . . . .	171

# 1 About This Guide

This guide provides you with an overview of functionalities and scenarios on how to use Data Manager. It is a collection of previously independent user guides that have been grouped into a single guide.

# 2 Getting Started with Data Manipulation

## 2.1 About Getting Started with Data Manipulation

This section is addressed to people who want to evaluate or use the Automated Analytics and in particular the Data Manager - Semantic Layer feature.

Before reading this section, you should read the sections *Classification/Regression* and *Segmentation/Clustering* that present respectively:

- An introduction to Automated Analytics
- The essential concepts related to the use of Automated Analytics features

No prior knowledge of SQL is required to use data manipulation - only knowledge about how to work with tables and columns accessed through ODBC sources. Furthermore, users must have “read” access on these ODBC sources.

To use SAP Predictive Analytics, users need write access on the tables `KxAdmin` and `ConnectorsTable`, which are used to store representations of data manipulations.

This section introduces you to the main functionalities of the data manipulation feature.

One of the useful features of data manipulation is the ability to declare arguments. Arguments are symbols with associated values that can be changed before executing the data manipulations. They can be used anywhere within data manipulation.

These data manipulations can all be performed by standard SQL engines embedded with all major relational databases. Instead, the data manipulation module can be seen as an object oriented layer that is used to generate data manipulation statements in SQL, which are processed, in turn, by the database server.

## 2.2 Definition

This section provides you with some useful definitions and details the technical requirements to the use of the Data Manager - Semantic Layer feature.

### Data Manipulation

Tabular representation of data made of lines and columns. Each Line represents an “observation”. Roles can be assigned to columns such as “input”, “skip”, “target” or “weight”.

## Data Preparation

Set of operations needed to create a data manipulation. It can be broken up into two separate phases: Semantic Layer and Data Encoding.

### Semantic Layer

Business intended data transformations, such as target definition or dataset filtering.

### Data Encoding

Technically driven data transformations that are automatically handled by the platform.

## 2.3 Creating a Data Manipulation

### 2.3.1 Selecting a Data Source

A data manipulation is based on existing database tables. The first step to create it is to select the database and the table you want to work with.

#### i Note

Throughout this document we will use *table* to globally represent database tables, views and SAP HANA information views.

1. On main menu of the application, click *Create a Data Manipulation* in the *Data Manager* section.  
The *Define New Data Manipulation* panel is displayed.
2. Click the *Browse* button corresponding to the *Database Source* field.  
The *Data Source Selection* window opens.
3. Select the database from which you want to create a new data manipulation.
4. If the database you want to access is password protected, enter the user name in the *User* field and the password in the *Password* field.

#### i Note

If you fail to enter the correct login/password, an error message is displayed when you try to select a table.

5. Click *OK* to validate your selection.

6. Click the *Browse* button corresponding to the *Table* field. The window *Data Source Selection* opens.

The following types of tables can be displayed in the *Data Source Selection* window:

- data manipulations created with the application,
  - standard database tables and SQL views,
  - SAP HANA information views if the datasource is an SAP HANA database.
7. Select in the list the table you want to use in the new data manipulation.
  8. Click *OK* to validate your selection.
  9. In the *New Table Alias* field, enter the name you want to use to refer to the selected table. By default the alias is based on the selected table name.
  10. Click *Next* to display the *Data Manipulation Editor*.

## 2.3.1.1 Data Sources Supported

Automated Analytics supports the following data sources:

- Text files (also called flat files) in which the data are separated by a delimiter, such as commas in *.csv* (Comma Separated Value) files.

### ! Restriction

When accessing data in *.csv* files, Automated Analytics only supports `CR` + `LF` (common on Microsoft Windows) or `LF` (common on Linux) for line breaks.

- Database management systems that can be accessed using ODBC.

### i Note

For the list of supported ODBC-compatible sources, see the SAP Product Availability Matrix (PAM) at <http://service.sap.com/sap/support/pam>.

For more information about using SAP HANA, see the related information below.

To configure Automated Analytics modeling tools to access data in your database management system, refer to the guide *Connecting your Database Management System on Windows* or *Connecting your Database Management System on Linux*.

- SAS files

## Related Information

[SAP HANA as a Data Source \[page 9\]](#)



## 2.3.1.2 SAP HANA as a Data Source

You can use SAP HANA databases as data sources in Data Manager and for all types of modeling analyses in Modeler: Classification/Regression, Clustering, Time Series, Association Rules, Social, and Recommendation.

---

SAP HANA tables or SQL views

found in the [Catalog](#) node of the SAP HANA database

---

All types of SAP HANA views

found in the [Content](#) node of the SAP HANA database.

An SAP HANA view is 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 analytic view is built on a fact table and associated attribute views. A calculation view executes a function on columns when the view is accessed.

### ! Restriction

- Analytic and calculation views that use the variable mapping feature (available starting with SAP HANA SPS 09) are not supported.
  - You cannot edit data in SAP HANA views using Automated Analytics.
-

Thanks to Smart Data Access, you can expose data from remote sources tables as virtual tables and combine them with HANA regular tables. This allows you to access data sources that are not natively supported by the application, or to combine data from multiple heterogeneous sources.

#### ⚠ Caution

To use virtual tables as input datasets for training or applying a model or as output datasets for applying a model, you need to check that the following conditions are met:

- The in-database application mode is not used.
- The destination table for storing the predicted values exists in the remote source before applying the model.
- The structure of the remote table, that is the column names and types, must match exactly what is expected with respect to the generation options; if this is not the case an error will occur.

#### ⚠ Caution

In Data Manager, use virtual tables with caution as the generated queries can be complex. Smart Data Access may not be able to delegate much of the processing to the underlying source depending on the source capabilities. This can impact performance.

## Prerequisites

You must know the ODBC source name and the connection information for your SAP HANA database. For more information, contact your SAP HANA administrator.

In addition to having the authorizations required for querying the SAP HANA view, you need to be granted the `SELECT` privilege on the `_SYS_BI` schema, which contains metadata on views. Please refer to SAP HANA guides for detailed information on security aspects.

### 2.3.1.3 Defining a metadata repository

The metadata repository allows you to specify the location where the metadata should be stored.

1. Choose between storing the metadata in the same place as the data or in a single place by checking the option of your choice.
2. In the list *Data Type*, select the type of data you want to access. For some type of data, you will need a specific license.

3. Use the *Browse* button corresponding to the Folder field to select the folder or database containing the data. In case of a protected database, you will need to enter the user name and the password in the fields *User* and *Password*.
4. Click the button *Edit Variable Pool Content* to edit the parameters of the variables stored in the variable pool.
5. Click *OK* to validate.

## 2.3.2 Data Manipulation Editor

The *Data Manipulation Editor* ribbon offers the following tabs: *Main*, *Edition*, and *Views*. It provides you with all the options needed to create, edit, and view the data manipulation components.

### What Can You Do With the *Main* Tab?

- Create, edit, and delete new computed fields, based on aggregates, free expressions, conditions, lookup tables, normalizations, or SQL expressions.
- Merge tables.
- Create filters.
- Create, list, and edit prompts.

### What Can You Do With the *Edition* Tab?

- Select fields to apply the same modification to several fields at once. Use the *Advanced Selection* tool to select fields based on their alias, their storage, their type, and the table or view they are issued from.
- Rename the selected fields by using standard renaming options such as changing the case, adding a prefix or a suffix, or replacing part of the name.
- Set the selected fields visibility.
- Change the selected fields type.

### What Can You Do With the *Views* Tab?

- List the fields and set their visibility.
- Display the data and their statistics.
- View the SQL expression for the current data manipulation.
- View the documentation for the current data manipulation. It includes a graphic summary, the list of visible fields, and details on any components used in the data manipulation.

## 2.3.2.1 Main

Use this tab to edit the fields by defining an aggregate for example, to merge fields, create filters or create prompts. All the options are also available by right-clicking the field list and selecting the appropriate option in the contextual menu.

### 2.3.2.1.1 Expression Editor

The expression editor allows you to create fields (one by one or several at a time) and to edit filter conditions as you would do with a calculator.

To build your expression, you have at your disposal:

- Functions: they allow to build complex expressions with one or more fields.
- Variables and their values:
  - the *Fields* of your database
  - the *Field sets* you have defined in your data manipulation
  - the *Prompts* you have defined in your data manipulation
  - the defined *Categories* for the existing fields/variables.
- *Field Association*
- *Messages*

#### 2.3.2.1.1.1 Basic Aggregate Functions

The Expression Editor supports the use of basic aggregate functions, which allows you to group your data into columns to simplify and reduce the dataset.

The basic aggregation functions are not displayed in the user interface of the Expression Editor by default. To see the basic aggregates functions, you can:

- Add the option `<-BasicAggregation>` to the command line starting the application as follow: `C:\Program Files\SAP Predictive Analytics\Desktop 3.1\Automated\EXE\Clients\KJWizardJNI\KJWizardJni.exe-BasicAggregation.`
- Or add the following line to the file `KJWizardJni.ini`: `arg.1=-BasicAggregation`. This file is found under the folder: `...\SAP Predictive Analytics\Desktop 3.1\Automated\EXE\Clients\KJWizardJNI\KJWizardJni.ini`.

The following aggregate functions are supported by SAP Predictive Analytics:

Basic Aggregate Functions	Formulas
Sum	AggregateSum()
Min	AggregateMin()
Max	AggregateMax()

Basic Aggregate Functions	Formulas
Count	AggregateCount()
Avg	AggregateAvg()

To use the basic aggregate functions in the Expression Editor, you can either type the formula or select the relevant basic aggregate functions.

### ❖ Example

You have a large dataset containing a lot of data. You would like to aggregate a measure, for example, `<Quantity>` or `<Amount>` by `<Time>` and `<Product>`, and use the results in *Modeler* for *Time Series Forecasting*:

1. Add a new variable `<Total Quantity>` with a basic aggregate function formula:  
`AggregateSum(sales_orders.quantity_ordered)` .
2. Select the keep visible check box for each variable required for the grouping: `<Month Begin Date>` and `<Product Id>`.

## 2.3.2.1.1.2 Creating a Field

1. Enter an expression in the text area located in the upper part of the panel.

You can use the following entry help functions:

- Double-click the name of a function in the *Functions frame* or of a variable (field or pre-defined fields set, prompt and category) in the *Variables* frame to insert them at the prompt location. In case of a function, a pattern is inserted giving information on the parameters to use.

### i Note

To know more about every function, an explanation label is available when moving the mouse over it as shown in the above screenshot as an example for the Arithmetic Operator "Absolute".

- You can also insert these elements in a chosen position by drag-and-dropping them from one of the trees to the text area.
  - When entering a variable name (field, prompt or set), press simultaneously the *Ctrl* and *Space* keys to display a list of variable names beginning with the entered text.
  - The color of the indicator located above the *Messages* area indicates the state of the formula. For further details, go to section Messages.
2. To validate the formula, click the *OK* button.
  3. A pop-up appears to name the new computed field. Enter a name in the *Name* field.
  4. Click the *OK* button.

### 2.3.2.1.1.3 Creating a Named Field Set

Creating several fields by applying the same calculation to various existing fields is a frequent need. The expression editor allows you to do that thanks to the use of field sets.

For example, the use of field sets allows you to sum up a large number of fields, or to compute their maximum.

1. In the *Variables* section, double-click the option *Field Sets*.

A sub-tree is displayed listing the existing field sets.

2. Double-click the option Create Field Set...

A new window opens listing all available fields.

3. In the field *Alias Mask*, enter a mask allowing filtering the fields by their name. A mask is made of a part common to the name of all the fields you want to see displayed, and of the star character (\*) allowing to complete the parts which differ in the field names. The star can be used anywhere in the mask and as many times as needed.
4. Uncheck the fields you do not want to keep in the field set.
5. In the field Set Name, enter a name for the new field set.
6. Click the OK button.

The window closes and the new field set is displayed in the list under the item Field Sets.

### 2.3.2.1.1.4 Editing a Named Field Set

1. Select the field set you want to edit.
2. Right-click the selected field set.  
A contextual menu is displayed.
3. Click the Edit option. The field set edition window opens.

#### i Note

To change the name of a field set amounts to duplicating it.

### 2.3.2.1.1.5 Deleting a Named Field Set

1. Select the field set you want to delete.
2. Right-click the selected field set.  
A contextual menu is displayed.
3. Click the *Remove* option.

### 2.3.2.1.1.6 Creating a Field Set on the Fly

When you want to apply a calculation to fields whose names have a common root, you can create a field set on the fly in the formula text field. A field set created on the fly is defined by a mask, that is a fixed part common to the name of all the wanted fields and a wildcard character representing the part of the names that changes for each field. Three wildcard characters can be used to define the field sets: the at sign (@), the hash sign (#) and the dollar sign (\$). Only one wildcard can be used for each field set, and a same wildcard cannot be used twice in the same formula.

In the formula text field, enter the mask corresponding to the fields you want to apply the calculation to, for example `income_@` and use it as a standard field.

### 2.3.2.1.1.7 Using Several Field Sets

1. Enter a formula using the field sets as explained above. The [Messages](#) area located in the lower part of the panel, indicates the number of fields that will be created.
2. When using several field sets in the same formula, you must select in the drop-down list [Field Association](#) how the fields from these sets will be associated.

#### i Note

To select an option in [Field Association](#), refer to section [Field Association](#) for further explanations and examples.

3. Click the [OK](#) button to validate the fields creation.  
A dialog box requesting you to name the new fields opens.
4. Enter a root common to all the newly created fields.
5. If the formula only uses named field sets, or if you do not need to further control the naming, go to the next step. If you have used one or more wildcard characters, you can use them in the alias to create a naming pattern. The wildcard characters are replaced by the corresponding name parts.
6. To create new fields, go back to the panel [Expression Editor](#).

### 2.3.2.1.1.8 Using a Field Set as a List of Variables

A field set can be used to define a list of arguments for n-ary functions, that is, functions with an undefined number of arguments. When a field set is used as argument of a n-ary function, and in this case only, you need to frame the set name with braces { } to force its interpretation as an argument list.

As an example, let's consider a database table whose fields `income_april`, `income_may`, `income_june` contain the monthly income. If you want to create a field containing the highest monthly income for the quarter, you need to use the following formula `greatestN({income_@})` which uses the `income_@` field set as a list of variables and thus amounts to the formula `greatestN(income_april ,income_may ,income_june)`. However when using the field set as is in the formula `greatestN(income_@)`, three fields are created by the three following formulas: `greatestN(income_april)`, `greatestN(income_may)` and `greatestN(income_june)`, which cannot be used.

1. Create a field set (named or on the fly) as explained above.
2. In the formula text field, enter the formula containing a n-ary function.
3. As argument for the function, enter the field set name between braces { }. For example, the formula `greatestN({incomes})` will create a field whose value is the highest value of the fields contained in the field set named `incomes`.

### 2.3.2.1.1.9 Extracting Categories

Using the existing variable categories to create conditions with interval, equality or inequality relations is a frequent need. The expression editor allows doing that through the category extraction feature.

1. In the *Variables* section, double-click the option *Categories*.  
A sub-tree is displayed with the mention *Extract Categories...*
2. Double-click the option *Extract Categories...*  
A new window that lists all available fields opens.
3. In the field *Alias Mask*, you can enter a mask allowing filtering the fields by their name. A mask is made of a part common to the name of all the fields you want to see displayed, and of the star character (\*) allowing to complete the parts which differ in the field names. The star can be used anywhere in the mask and as many times as needed.
4. Check the fields you want to extract categories from.
5. In the *Sample Size* section, check the option *Compute statistics over the xxxx first lines* and specify the number of first lines in case you do not want to extract categories from the whole dataset. Keep the default option otherwise.

#### i Note

Some categories of the dataset may be missing in the expression editor in case you choose to perform category extraction *over the xxxx first lines*.

6. Click the *Extract* button. A progress bar is displayed.
7. When the progress bar closes, click the *Close* button. Two tree items have been added in the *Categories* section: *Nominal Variables and Other Variables*.
8. Choose the type of variable, the variable and the category that you want to use. In the example below, *Nominal*, *client\_type* and *OWNER* have been chosen respectively.

### 2.3.2.1.1.10 Using Advanced Settings for Category Extraction

The advanced settings let you choose the way categories are extracted from the dataset in a finer way.

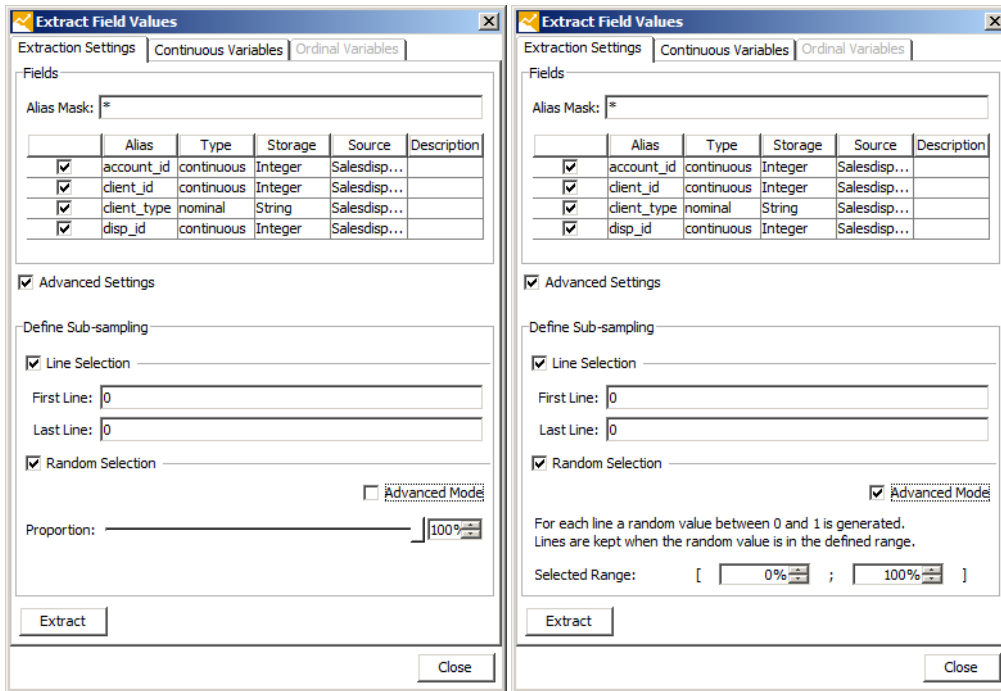
To use advanced settings for category extraction:

1. In the *Variables* section, double-click the option *Categories*.  
A subtree is displayed with the mention *Extract Categories...*
2. Double-click the option *Extract Categories...*



A new window that lists all available fields opens.

- In the field *Alias Mask*, you can enter a mask allowing filtering the fields by their name. A mask is made of a part common to the name of all the fields you want to see displayed, and of the star character (\*) allowing to complete the parts that differ in the field names. The star can be used anywhere in the mask and as many times as needed.
- Select the fields you want to extract categories from.
- Check the *Advanced Settings* box. Depending on the selected options, one of the following screen is displayed:



### i Note

The *Sub-sampling* mode replaces the *Sample size* mode available in case the box *Advanced Settings* is not checked. Please refer to the procedure *To Extract Categories* (see "Extracting Categories") for details about the *Sample size* mode.

- Check the right combination of options in the *Define Sub-sampling* section, depending on how category extraction must be performed. The table below describes how the category extraction is processed for each of possible combinations.

Combination of options	Description of category extraction process
<i>Line Selection</i>	The categories are extracted in the range of lines in the dataset defined by the values typed in the <i>First Line</i> and <i>Last Line</i> fields.

### Random Selection

The categories are extracted from lines picked up randomly in the dataset.

The number of lines used to extract categories equals to the proportion of the total number of lines in the dataset, defined using the *Proportion* slider.

---

### Random Selection - Advanced Mode

The categories are extracted from lines picked up randomly in the dataset.

A random value between 0 and 1 is attributed to each line. Lines with a value in the *Selected Range* (value / 100) are used for category extraction. Other lines are skipped.

---

### Line Selection + Random Selection

The categories are extracted in the range of lines in the dataset defined by the values typed in the *First Line* and *Last Line* fields.

The categories are extracted from lines picked up randomly within this range of the dataset.

The number of lines used to extract categories equals to the proportion of the total number of lines within the defined range, defined using the *Proportion* slider.

---

### Line Selection + Random Selection - Advanced Mode

The categories are extracted in the range of lines in the dataset defined by the values typed in the *First Line* and *Last Line* fields.

The categories are extracted from lines picked up randomly within this range of the dataset.

A random value between 0 and 1 is attributed to each line in the defined range. Lines with a value in the *Selected Range* (value / 100) are used for category extraction. Other lines are skipped.

#### **i** Note

Whatever the chosen combination, some categories of the dataset may be missing in the expression editor in case you choose to perform category extraction on a small portion of the dataset.

7. Click *Extract*. A progress bar is displayed.
8. When the progress bar closes, click *Close*. Two tree items have been added in the *Categories* section: *Nominal Variables* and *Other Variables*.
9. Choose the Type of variable, the variable and the category that you want to use.

## 2.3.2.1.1.11 Field Association

When using several field sets in the same formula, you must select in the drop-down list *Fields Association* how the fields from these sets will be associated:

- *Associate by Position*: the fields from each set are associated depending on their position in the database table.

- *Associate by Value*: the fields from each set are associated depending on the value represented by the wildcard characters used to define the field sets.
- Do Cartesian Product: all the fields from one set are associated to all the fields from the other set.

Examples:

- Let's consider a database table containing the following fields ordered as displayed:

income_january
income_april
income_february
income_march
expenses_march
expenses_january
expenses_april

- The following formula `income_@ - expenses_#` uses two field sets, one grouping all the fields starting with `income_`, and the other all the fields starting with `expenses_`:

income_@	expenses_#
income_january	expenses_march
income_april	expenses_january
income_february	expenses_april
income_march	

1. The option *Associate by Position* results in the following calculations:

Calculation	Position
income_january - expenses_march	1
income_april - expenses_january	2
income_february - expenses_april	3

2. If the option *Associate by Value* is selected, the application will try to match the value represented by @ and the value represented by #, resulting in the following calculations:

Calculation	@ and # Values
income_january - expenses_january	january
income_april - expenses_april	april
income_march - expenses_march	march


3. The option *Do Cartesian Product* results in the following calculations:

Calculation

income_january - expenses_march
income_january - expenses_january
income_january - expenses_april
income_april- expenses_march
income_april - expenses_january
income_april - expenses_april
income_february - expenses_march
income_february - expenses_january
income_february - expenses_april
income_march - expenses_march
income_march - expenses_january
income_march - expenses_april

## Messages

The color of the indicator located above the [Messages](#) area indicates the state of the formula.

If the indicator is...		the formula...
	red	contains an error, which is reported in the message area. It is not possible to validate it (the Next button is disabled).
	yellow	can be validated but some inconsistencies may occur and are reported in the message area.
	green	is valid.

In the case of an error or a warning, the [Messages](#) area located in the lower part of the panel provides details to help you understand the problem.



### 2.3.2.1.2 Fields

Use the [Fields](#) tab to display the fields from the source table and add your own fields.

You can modify the way this table is displayed by:

- selecting the columns to be displayed,
  - sorting the fields by column,
  - changing the column order.
1. To select the columns to be displayed:
    1. Right-click the table heading. A contextual menu is displayed.
    2. Check the column you want to display or uncheck it if you want to hide it.
    3. Repeat step 2. until the table displays the information you want.
  2. To sort the fields, click the heading of the column you want to use as the sorting parameter. One click sorts it in ascending order, two clicks in descending order, three clicks unselect the column as the sorting parameter. A sign in the column heading indicates the sorting order that is applied.
  3. To modify the columns order, click the heading of the column you want to move and drag it to its new place.

### 2.3.2.1.2.1 Operations on Existing Fields

The following operations are available on existing fields:

- Modify a field alias.
- Set a field visibility, that is select if the field will be added to the data manipulation or not. Fields that are set to invisible do not appear in the *Data and Statistics* tab but can be used for a merge or a filter, for example.
- Add a description to a field
- Edit a field information. Only computed fields created by the user can be edited. Non-editable fields appear grayed.

### 2.3.2.1.2.2 Modifying a Field Alias

Each field can be referred to by an alias. Aliases are usually used to differentiate fields having the same name but coming from different tables. By default, a field alias is the field name but you can change it.

1. Double-click the alias you want to modify.
2. Enter the new alias.
3. Click another cell to validate your change.

### 2.3.2.1.2.3 Setting a Field Visibility

Field visibility allows you to choose which fields appear in the dataset.

There are two ways to set a field visibility:

Click the *Visibility* checkbox corresponding to the selected field.

Or:

1. Right-click the line corresponding to the selected field.  
A contextual menu is displayed.
2. Select the *Set Visibility* option.
3. Select the option you want to apply to the field.

#### 2.3.2.1.2.4 Adding a Description to a Field

A field description allows you to add comments on this field to make it easier to use.

1. Double-click the *Description* cell corresponding to the selected database field.
2. Enter the field description.
3. Click another cell to validate your change.

#### 2.3.2.1.2.5 Editing a Field Information

You can only edit computed fields created by the user.

There are two ways to edit a field information:

Double-click a non-editable cell of the selected field. The panel *New Computed Field* is displayed.

Or :

1. Right-click the selected field.  
A contextual menu is displayed.
2. Select the *Edit* option. The panel *New Computed Field* is displayed.

#### 2.3.2.1.2.6 Creating a New Computed Field

New computed fields are useful when you want to add new variables that are not saved in any table but can be derived from other data on demand. You should add variables when you think doing so can render more information available to a predictive or descriptive model. For example, a ratio between two variables that has business meaning, or the conversion of a birth date to an age are useful transformations that would be difficult for modeling software to infer automatically.

Technically, this can be decomposed into six types of variable creation:

- Aggregate,
- Condition,
- Lookup Table,
- Normalization,

- SQL Expression,
- Function.

You can also use the *Expression Editor* provided by the application to define new fields as you want (see section *Expression Editor*).

There are two ways to create a new computed field: by clicking *New* or by right-clicking the

1. Click the *+ New* button.

The list of available field types is displayed.

2. Select the type of field you want to create.

The corresponding edition panel is displayed.

### 2.3.2.1.2.7 Aggregates

Use the *Aggregate* option to create aggregates similar to those automatically created by the event logging feature. Generate many different aggregates thanks to the event logging feature, and create a regression or classification model and identify the aggregates most relevant to your business issue. You can then create a new dataset containing only the most relevant aggregates.

To Create a New Aggregate

In the *New* list, select the *New Aggregate* option.

The *Define an Aggregate* panel is displayed.

### 2.3.2.1.2.8 Defining the Aggregate Settings

To select the events table, in the section *Events Table Selection*:

1. In the *Table* list, select the table containing the events linked to your reference table.
2. In the list *Date Column*, select the field indicating the events order.

To Specify the Join Keys, in the section *Join Keys*:

1. In the list *Reference Table Key*, select the field corresponding to the join key in the reference table.
2. In the list *Events Table Key*, select the field corresponding to the join key in the events table.

To specify the aggregate operation, in the section *Aggregate Operation Specification*:

1. In the *Function* list, select the type of function to use. The following functions are available:

Functions	Description	Returned Values
<i>Count</i>	computes the number of occurrences	number of occurrences
<i>Sum</i>	compute the sum	sum
<i>Average</i>	compute the mean	mean
<i>Min</i>	identifies the minimum value	minimum value

Functions	Description	Returned Values
<i>Max</i>	identifies the maximum value	maximum value
<i>Exists</i>	checks if at least one event exists for the current reference	0 if no event has been found 1 if at least one event has been found
<i>NotExists</i>	checks if no event exists for the current reference	0 if at least one event has been found 1 if no event has been found
<i>First</i>	identifies the first occurrence  Note that this function needs a date column.	value of the first chronological occurrence for the current reference
<i>Last</i>	identifies the last occurrence  Note that this function needs a date column.	value of the last chronological occurrence for the current reference

- In the list *Target Column*, select the variable on which you want to apply the selected function. When you select the Count operation, an additional option (\*) allows you to avoid selecting a specific column. It is possible to select more than one Target Column.

## 2.3.2.1.2.9 Defining the Period Settings

You can create aggregates on the whole dataset, or filter it by date. As in event logging, dates used as filter can be constants, variables, or prompts.

- Select the tab *Period Settings*.
- Check the box *Define Periods* to select the start and end dates of the time window used to filter the events.
- Select if you want to define a *Single Period* or *several Successive Periods*.
  - For a Single Period:
    - Select the type of input to use as start date. Three types are available:
      - the Field type allows you to use a date stored in the database and thus proper to each customer (such as a date of first purchase or a date of churn).
      - the Constant type allows you to select a fixed date (such as a marketing campaign launch date).
      - the Prompt type allows asking the user to fill the information when the model is generated.
    - Select the storage format to use (DateTime or Date). If you have selected the Field type, only the fields corresponding to the selected storage format are displayed in the list.
    - Select the value to use as the start date. Note that the start date is included in the defined period.
      - In case of a field, select the field in the displayed list.
      - In case of a constant, enter the constant value or use the calendar button located to the right.
      - In case of a prompt, select the prompt to use or click the + button to create a new prompt.
    - Repeat steps 2 to 4 for the end date. Note that the end date is excluded from the defined period.
  - For Successive Periods:



Define the number of successive periods you want, their length and the starting date by using the hyperlinks, which are underlined in blue or green. The start date is included in the periods.

## 2.3.2.1.2.10 Defining the Filters and Pivots

The *Filter* option allows you to filter your data depending on the variables values. Each defined filter generates a variable.

The *Pivot* option allows you to create one variable for each selected value. Creating a pivot amounts to creating a filter on one category for each selected category. The pivot is always applied to the Event Table.

Both options can be complementary.

### 2.3.2.1.2.10.1 Defining a Filter

Two types are available:

- Filter *Event Table* shows the list of filters performed on the Event Table;
  - Filter *Reference Table* shows the list of filters performed on the Reference Table.
1. Click the *Add* button to define a complex condition using the expression editor.
  2. Select and click the *Edit* button to amend the previously defined condition.
  3. Click the *Remove* button to delete the previously defined condition

### 2.3.2.1.2.10.2 Defining a Pivot

1. Select the variable to filter by in the *Variables* drop-down list.
2. To add categories to the table, you can:
  - automatically extract the variable categories by clicking the magnifier button located next to the list and then select the values to keep or exclude by checking the corresponding Selection box.
  - enter a value in the field *New Category* and click the + button.
  - load a list of categories from files by clicking the open file button located at the right of the Categories table list and then select the values to keep or exclude by checking the corresponding *Selection* box.

#### i Note

The number of created variables is indicated at the bottom of the panel. This number grows exponentially when filtering by pivot. The higher the number of variables, the longer the model learning.

3. Save the selected categories by clicking the *save* button located at the right of the *Categories table* list.

### i Note

As mentioned above, the [Save](#) button only saves the selected categories that are the ones with the [Selection](#) box checked.

4. Check the box [Also create aggregates without filtering](#) to create an unfiltered aggregate variable.

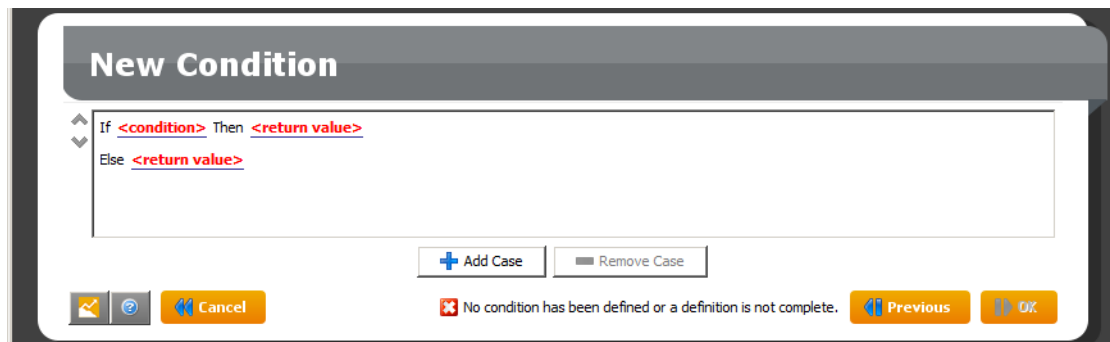
## 2.3.2.1.2.11 Defining a Condition

A condition allows you to define a field value depending on another field value case by case.

To Create a Condition:

1. In the [New](#) list, select the option [New Condition](#).

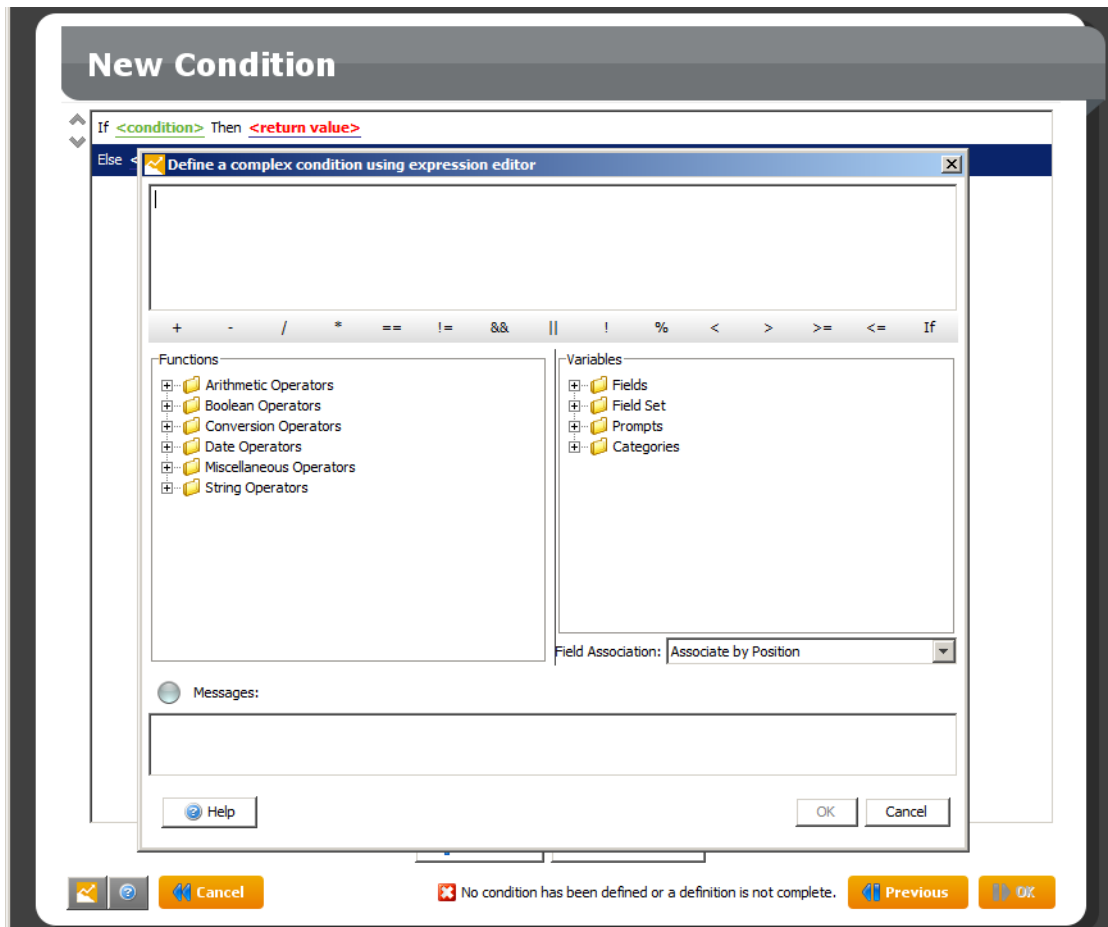
The panel [New Condition](#) is displayed.



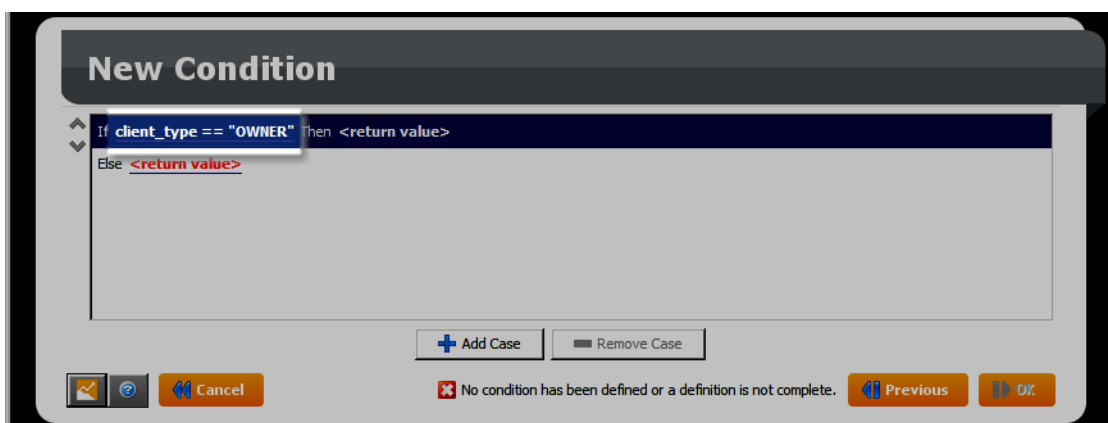
The screenshot shows a dialog box titled "New Condition". Inside the dialog, there is a text area with the following text: "If <condition> Then <return value>" and "Else <return value>". Below the text area, there are two buttons: "+ Add Case" and "- Remove Case". At the bottom left of the dialog, there are three icons: a chart icon, a refresh icon, and a "Cancel" button. At the bottom right, there is a status message: "No condition has been defined or a definition is not complete." followed by "Previous" and "OK" buttons.

2. In the text field, click `<condition>`.

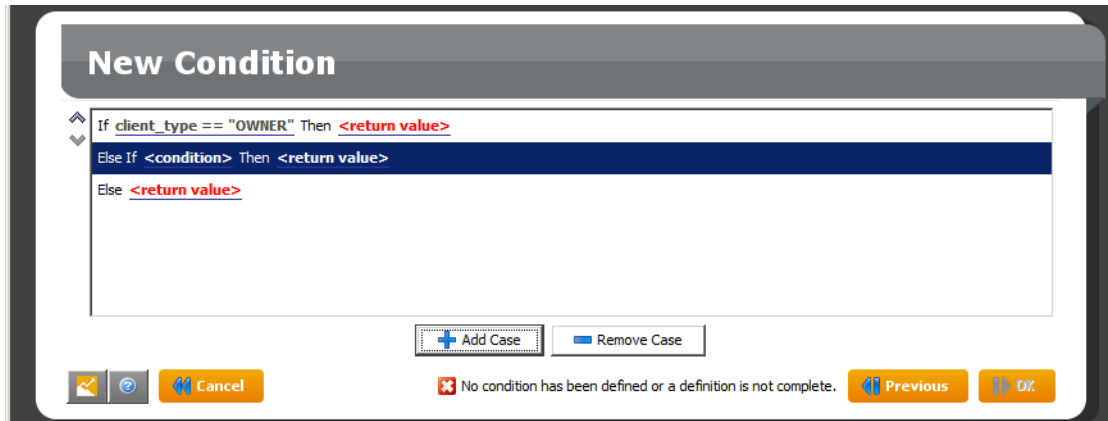
The expression editor opens.



3. Set the condition parameters as detailed in the section Expression Editor.
4. Click the *OK* button. `<condition>` is replaced by the condition you have defined.



5. Click `<return value>` to define the value the new computed field will take when the condition is true.
6. Click the button *Add Case* to add the new case to the list.



7. Repeat steps 4 and 5 for each new case. The order in which the cases are listed is important since the first true condition will determine the computed field value.
8. Use the buttons and located on the left to order the cases.  
You can delete a case from the list by selecting it and clicking the button Remove Case.
9. On the line Else, click `<return value>` to define the value to be used when none of the defined cases are true.
10. Click the *Next* button. The pop-up *Enter the Computed Field Name* opens.
11. Enter the new field name in the Name field.

#### i Note

If you enter an existing name, a message is displayed and the OK button is deactivated.

12. Click the *OK* button. The Fields list is displayed containing the newly created field.

## 2.3.2.1.2.12 Creating a Lookup Table

The user specifies cases, each of which is made up of a list of discrete values and a corresponding label. A classic example is a “look-up table” that is used as a dictionary to translate values from identifiers into strings, or to group values representing fine distinctions into a smaller number of more general bins.

1. In the *New* list, select the option *New Lookup Table*.  
The panel *Add a New Lookup Table* is displayed.
2. In the *Field* list, select the field from which the value will come from.
3. Select the type of results you want to generate in the list Output Storage.
4. In the column *IF*, double-click the value `<Undefined>` to enter the first value you want to add as an entry.
5. In the column *THEN*, double-click the value `<Undefined>` to enter the result value corresponding to that first input value.
6. To add another set of values, click the **+** button located on the right.  
A new line with undefined values is displayed.
7. Repeat steps 4 and 5 for this new entry.
8. In the field *Other Values*, enter the value that will correspond to the field values not set as specific entries.

9. Click the *Next* button. The pop-up *Enter the Computed Field Name* opens.
10. Enter the new field name in the Name field.

### **i** Note

If you enter an existing name, a message is displayed and the *OK* button is deactivated.

11. Click the *OK* button. The Fields list is displayed containing the newly created field.

## 2.3.2.1.2.12.1 Deleting Entries from the Lookup Table

1. In the list of entries, select the one you want to delete.

2. Click the - button located on the right.  
The selected entry is deleted.

## 2.3.2.1.2.13 Using Variable Categories to Fill the Lookup Table

You can extract the categories from a nominal variable to fill the lookup table. Be aware that extracting categories can be lengthy if the number of categories is high.

### **⚠** Caution

If the nominal variable you have selected contains too many categories (for example an Id variable), the categories will not be extracted.

To Fill the Lookup Table with Variable Categories:

1. Click the *binoculars* button.  
A contextual menu is displayed.
2. If you have already extracted once the categories from the selected variable, select the option *Select Reference Variable*.

3. Else, select the option *Extract values and fill*. The pop-up window *Extract Field Values* opens.
4. Use the tab Extraction Settings to define the extraction parameters:
  - Fields: the fields for which you want to extract the categories,
  - Sample size: the size of the dataset sample from which you want to extract the categories.
5. Click the *Extract* button. The list of values in the lookup table is filled.
6. Click the *Close* button.
7. Define for each value the corresponding output value.

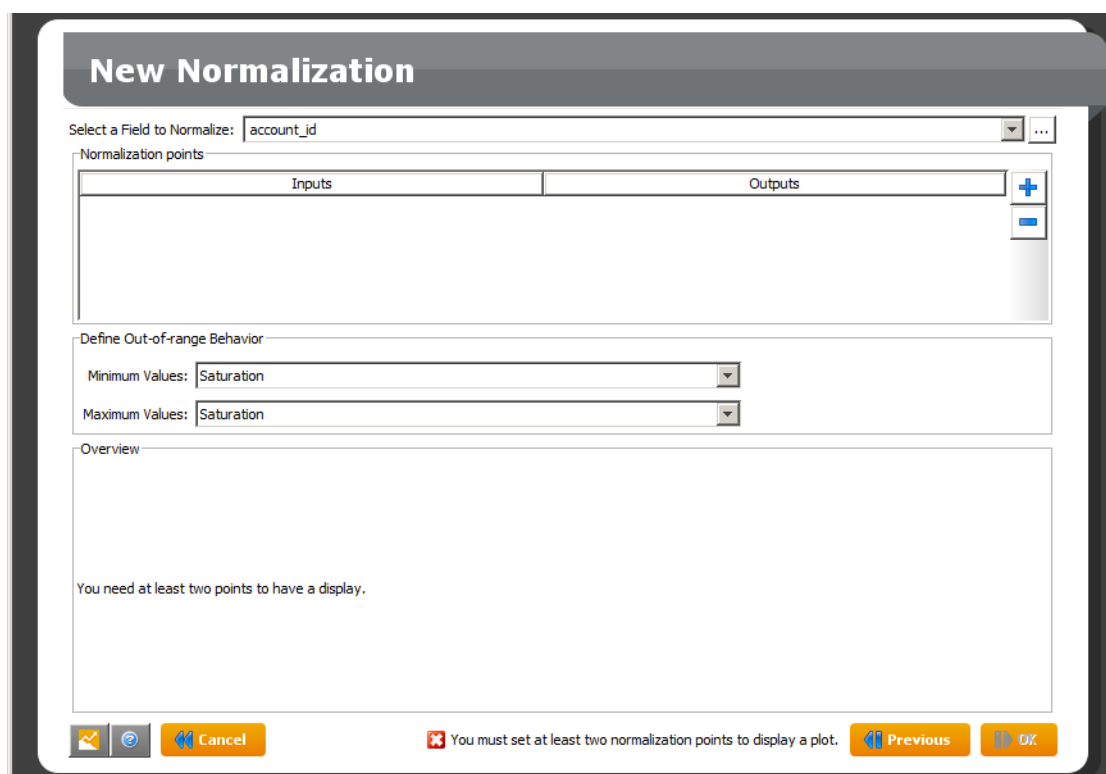
## 2.3.2.1.2.14 Defining a Normalization

Normalization is a standard Semantic Layer primitive that appears in PMML (Predictive Model Markup Language), a data mining specification language defined by the Data Mining Group (DMG). Normalization is frequently applied to numeric variables prior to data mining and consists of a piece-wise linear transform with the resultant variable typically ranging from 0 to 1. This can be used for rank transformations, where the output represents magnitude in terms of the approximate proportion (percentile) of values below the input value. Alternatively, a field may be converted based on how many standard deviations a value is from the field's mean. Part of normalization is also specification of what value to use when a numeric input value is unknown or out of the range seen in the training data.

To create a new normalization:

1. In the *New* list, select the option *New Normalization*.

The panel *New Normalization* is displayed.



**New Normalization**

Select a Field to Normalize:

Normalization points

Inputs	Outputs

Define Out-of-range Behavior

Minimum Values:

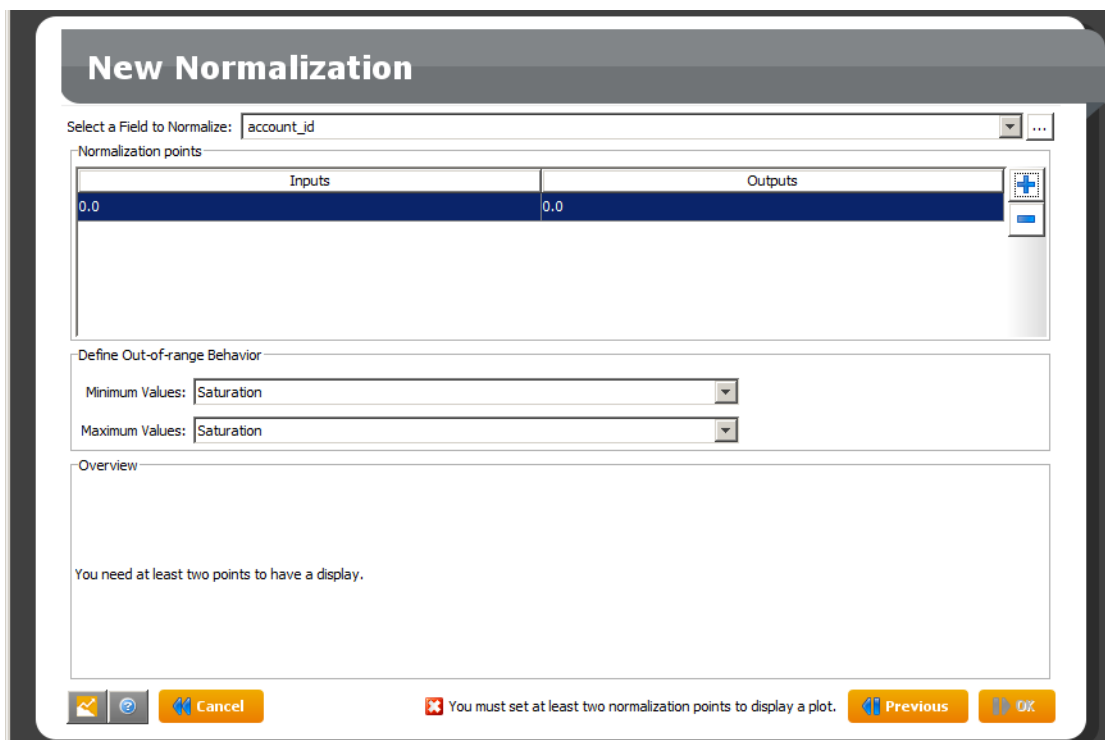
Maximum Values:

Overview

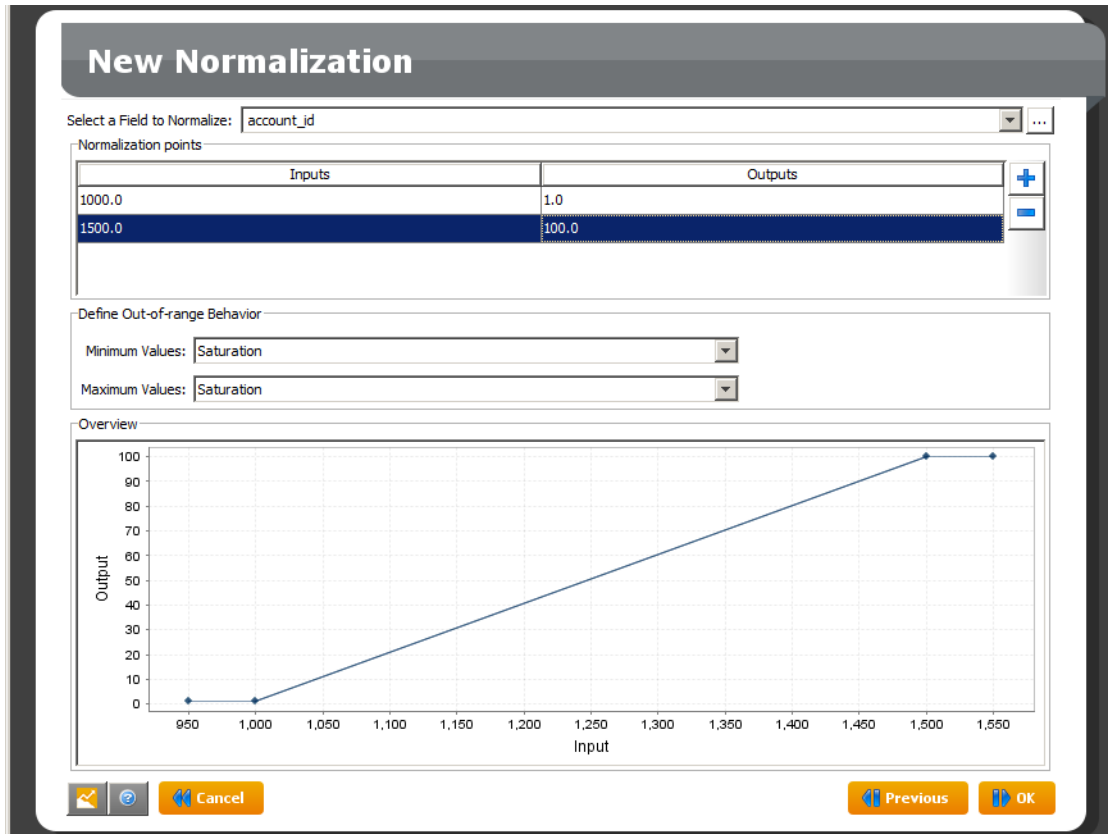
You need at least two points to have a display.

✖ You must set at least two normalization points to display a plot.

2. Select a field in the list *Select a Field to Normalize*. Only integer or number fields are displayed.
3. In the section *Normalization Points*, click the + button to add a normalization point.



4. Double-click the cell in the *Inputs* column to define the lowest value of the range.
5. Double-click the cell in the *Outputs* column to define the corresponding output value.
6. Repeat steps 3 to 5 to create the highest point of the normalization. The corresponding plot is displayed in the *Overview* section at the bottom of the panel.



You need to define at least two points, but you can add more if needed by repeating steps 3 to 5 for each new point.

- In the list *Minimum Values* of the section *Define Out of Range Behavior*, select the behavior that should be applied to values lower than the lowest point previously set. The available values are detailed in the following table:

Behavior	Out of Range Values Correspond to...	Corresponding Plot
<i>Saturation</i>	the value of the range bounds	
<i>Slope</i>	the continuation towards infinity of the straight line leading to the last bound.	



Behavior	Out of Range Values Correspond to...	Corresponding Plot
<i>User Defined</i>	a user defined value. To set a user defined value, enter the value in the text field and click the Refresh button to actualize the plot.	
<i>Null Value</i>	the Null value. Meaning that they are not displayed on the plot.	

8. Click the *Next* button. The pop-up *Enter the Computed Field Name* opens.
9. Enter the new field name in the *Name* field.

#### **i** Note

If you enter an existing name, a message is displayed and the *OK* button is deactivated.

10. Click the *OK* button. The Fields list is displayed containing the newly created field.

## 2.3.2.1.2.15 Defining an SQL Expression

An SQL expression field allows you to use predefined SQL queries.

To Create an SQL Expression:

1. In the *+New* list, select the option *New SQL Expression*.  
The panel New SQL Expression is displayed.
2. Enter a valid SQL expression in the text field. If the expression is not correct, an error will be displayed when displaying the tab *View Data*.
3. Select the type of result the SQL expression will return in the list *Result Type*.
4. In the *Type* list, select the value type of the result.
5. Click the *OK* button. The pop-up *Enter the Computed Field Name* opens.
6. Enter the new field name in the *Name* field.

#### **i** Note

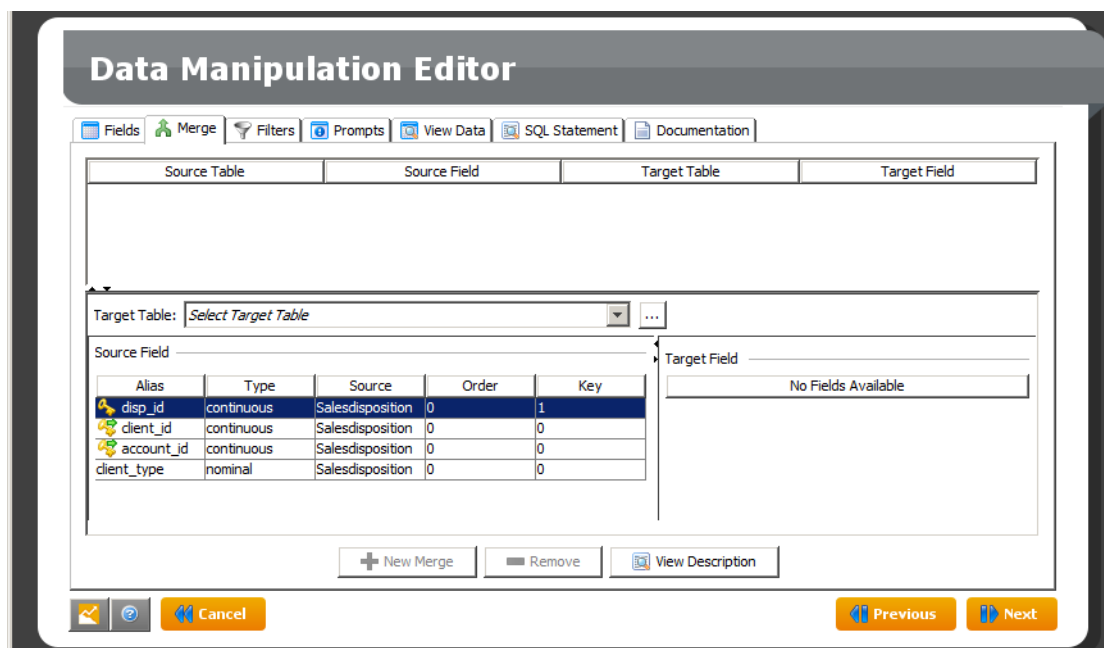
If you enter an existing name, a message is displayed and the *OK* button is deactivated.

Back to the *Data Manipulation Editor* panel, the *Fields list* is displayed containing the newly created field.

7. Click the Next button to continue or repeat steps 1 to 6 to create a new SQL expression.

## 2.3.2.1.3 Merge

The Merge tab allows you to create a merge between the source table and another table in your data base, that is to add information contained in another table when the selected field from the source table is equal to the selected field of the target table.



### 2.3.2.1.3.1 Creating a Merge

1. Select the table to be joined in the list *Target Table*.
2. Click the + button.
3. Select the source field in the list Source Field.

If the selected table contains fields corresponding to the source field, they are displayed in the list Target Field, else the message *No Fields Available* is displayed.

4. Select the target field in the list *Target Field*.
5. Click *OK* to create the merge. This button is only activated when all the elements needed for the merge are selected.

Once a merge has been created, all the fields of the target table are added to the list Source Field allowing you to create new merges from them.

When merging new fields, you can define the naming by yourself by adding a prefix or a suffix to the aliases.

- In the *Prefix* field, enter the prefix of your choice followed by an underscore.
- In the *Suffix* field, enter the suffix of your choice preceded by an underscore.

## 2.3.2.1.3.2 Removing a Merge

To remove a merge:

1. Select the merge to remove in the list.
2. Click the *Remove* button.

## 2.3.2.1.3.3 Creating a Composite Key

After the merge of two fields from different tables, you have the possibility to create a composite key by joining two other fields of these same tables.

1. Select the source field in the list *Source Field*.
2. Select the target field in the list *Target Field*.
3. Click the button *New Merge*.

A contextual menu opens.

4. Choose the option *Add a Key Pair*.

The source and target fields of the existing merge displayed in the upper part of the panel, are updated.

## 2.3.2.1.3.4 Removing a Composite Key

1. Select the merge with the composite key to remove in the list.
2. Click the *Remove* button.

A contextual menu opens.

3. Choose the option *Remove Key Pairs*.

A pop-up window is displayed.

4. Select the key pair you want to remove.
5. Click OK.

The source and target fields of the persisting merge, displayed in the upper part of the panel, are updated.

## 2.3.2.1.4 Filters

The Filters tab allows you to select only the records of interest to answering your business question. You can compare a field value to another field, a constant or a prompt and keep only the corresponding observations.

To display the Filters List, click the *Filters* tab. The *Filters* panel is displayed.

## Applying the Filters

You can choose if you want the selected records to match all the filters (which corresponds to the logical operator AND) or at least one of the listed filters (which corresponds to the logical operator OR).

To Select How to Apply the Filters, in the section *Keep Only Records* that, check the radio button corresponding to the option you want to apply.

## Operations on Existing Filters

To Create a Filter:

1. Click the button *New Condition*. The *Expression Editor* opens.
2. Set the parameters as detailed in the section Expression Editor.

## Removing a Filter

To Remove a Filter:

1. Select the filter you want to remove in the list.
2. Click the *Remove* button.

### 2.3.2.1.5 Prompts

A prompt allows you to require a value from the user when using the data manipulation in any feature of the application.

You can use two types of prompt in data manipulations: authored prompts or inherited prompts.

- **Authored prompts** are created in the current data manipulation with Data Manager. They can be edited and deleted directly and can be used in field, filter, and merge definitions in the current data manipulation.
- **Inherited prompts** are inherited from an SAP HANA view or from another data manipulation. They cannot be edited or deleted directly. You can edit a prompt inherited from another data manipulation by opening that data manipulation. Prompts inherited from SAP HANA views cannot be edited with Data Manager. Inherited prompts cannot be used in field, filter, or merge definitions.

#### 2.3.2.1.5.1 Authoring Prompts

There are two ways to create a prompt: either in the *Prompt* tab or when creating a new field or filter.

### 2.3.2.1.5.1.1 Creating a Prompt

1. On the *Prompt* tab, click *New Prompt*. The *Prompt Editor* opens.
2. Enter a name for the prompt in the *Name* field. This name will allow you to select the prompt as a value when creating a field or filter. Select the type in the *Type* list.
3. Enter the default value in the *Value* field. When prompting the user, this value will be suggested by default.
4. Enter the sentence that will ask the user for a value in the *Description* field. For example: "What is the minimum age required?"
5. Click *OK* to create the new prompt.

### 2.3.2.1.5.1.2 Editing a Prompt

You can edit only authored prompts.

1. On the *Prompt* tab, click *Edit*. is displayed.
2. In the *Prompt Editor*, modify the information as needed. To prevent errors, only the default value and the description can be modified. If you want to modify a prompt name or type, you need to create a new prompt.
3. Click *OK* to validate the modifications. A message box asks you to confirm the modifications.
4. Click *Yes*.

### 2.3.2.1.5.1.3 Removing a Prompt

You can delete only authored prompts that are not used in a field or a filter.

1. In the *Prompt* tab, select the prompt you want to delete.
2. Click *Remove*. If the prompt is used in the dataset, the prompt is not deleted and a warning message appears.

### 2.3.2.1.5.2 Associating Prompts

If a data manipulation is based on another data manipulation or on an SAP HANA information view, it inherits the prompts they contain. Prompts can be associated and used as a single prompt if they have the same name and storage value. You cannot associate prompts from different sources: you cannot associate an authored prompt with an inherited prompt, or a prompt inherited from an SAP HANA view with a prompt inherited from a data manipulation. However you can associate prompts inherited from different SAP HANA views, or prompts inherited from different data manipulations.

### ⚠ Caution

Prompts are considered similar when their name, storage value, and source are identical. However, other characteristics may be different. Make sure that they actually refer to the same thing and that the same value is expected for all of them before associating them.

1. In the *Main* tab of the *Data Manipulation Editor*, click *Prompts*.

The *Prompts* editor opens.

2. Select the *Association Policies* tab.

Prompts with the same name, storage value, and source appear as one item in the list. They are referred as a set of prompts in the following steps of the procedure.

3. Select the set of prompts you want to define an association policy for.
4. Click the *Policy* cell.
5. Select the policy you want to apply to that set of prompts.

Name	Description
<i>One value for all</i>	The prompt is displayed once. The value entered by the user is used for all prompts in this set.
<i>One value for each</i>	All prompts in this set are displayed. The user must enter a value for each prompt.

6. Click *Close* to save your changes.

## 2.3.2.2 Edition

Use this tab to select fields and apply modifications to several fields at once such as renaming, changing their visibility or their type.

### 2.3.2.2.1 Selection

The *Selection* section allows you to select fields and apply batch modifications on several fields at once. All the options are also available by right-clicking the field list and selecting the *Select* option in the contextual menu.

Option	Description
<i>Advanced selection</i>	Selects fields according to various criteria such as their alias, type, storage, or the table they are issued from.
<i>Select All</i>	Selects all visible fields.
<i>Toggle Selection</i>	Reverses the current selection. The selected fields become unselected and the unselected fields become selected.

## 2.3.2.2.2 Field Modification

Modification	Procedure
Rename fields	<ol style="list-style-type: none"><li>1. Select the fields you want to rename.</li><li>2. Click <i>Rename</i>.</li><li>3. Select the modification you want to apply in the drop-down list. The following options are available:<ul style="list-style-type: none"><li>○ <i>To Upper Case</i></li><li>○ <i>Truncate</i></li><li>○ <i>Add Prefix</i></li><li>○ <i>Add Suffix</i></li><li>○ <i>Replace String</i></li></ul></li></ol>
Set field visibility	<ol style="list-style-type: none"><li>1. Select the fields whose visibility you want to modify.</li><li>2. Click <i>Set Visibility</i>.</li><li>3. Select the option you want to use in the dropdown list.</li></ol> <p>By default, the list displays only visible fields. To see invisible fields, uncheck the <i>Display only visible fields</i> option below the field list.</p>
Set field type	<ol style="list-style-type: none"><li>1. Select the fields whose type you want to modify .</li><li>2. Click <i>Set Type</i>.</li><li>3. Select the new type in the dropdown list. The following types are available:<ul style="list-style-type: none"><li>○ <i>nominal</i></li><li>○ <i>ordinal</i></li><li>○ <i>continuous</i></li><li>○ <i>textual</i></li></ul><p>Make sure that the type you select matches the actual content of the fields you have selected.</p></li></ol>

## 2.3.2.3 Views

Use this tab to display the data and their statistics, and to view the SQL expression and the generated documentation corresponding to the current data manipulation.

### 2.3.2.3.1 Data and Statistics

With the option *Data and Statistics*, you can display the data manipulation content and check if the results correspond to what you expect. You can also compute statistics on the data and generate graphics. If the data manipulation contains prompts, you are asked to provide their values so that the data can be displayed.

### 2.3.2.3.1.1 Sorting the Data

1. A first click on a column heading sorts the data by ascending order.
2. A second click on the same column heading sorts the data by descending order.
3. A third click deactivates the sort option.

### 2.3.2.3.1.2 Selecting the Number of Rows to Display

1. In the field *First Row Index*, enter the number of the row above which you want to start displaying the data.
2. In the field *Last Row Index*, enter the number of the last row you want to display.
3. Click the *Refresh* button to display the selected rows in the table above.

### 2.3.2.3.1.3 Searching a Variable

To search for a specific variable in your dataset and display it, use the *Search* button located at the bottom right corner of the *Data and Statistics* panel.

1. Click the *Search* button.  
The *Search window* is displayed.
2. Select the type of search you want to do.

Search Type	Description
<i>Index</i>	Search a variable by index number. This number can be found in the first column of the <i>Fields</i> tab.
<i>Name</i>	Search a variable by name.

3. Click *OK* when you have entered the index number or selected the variable name.  
The selected variable column is highlighted.

### 2.3.2.3.2 SQL Statements

The *Generated SQL* tab displays the SQL query corresponding to the data manipulation being built.

### 2.3.2.3.3 Documentation

The *Documentation* tab allows you to get an overview of your data manipulation. It contains all the options selected for your data manipulation like filters, merges, prompts, or expressions.



This screen shows:

- Graphic Summary
- Visible / Invisible Fields
- Prompt
- Expressions
- Filters

This overview can be customized with the [Settings](#) option:

- Section Settings
- Field Settings

Use the [Overview Settings](#) to choose the overview format both for viewing and exporting. The generated file can be saved in `.txt`, `.html`, and `.rtf` file formats.

## 2.4 Saving a Data Manipulation

The application provides three ways to save a data manipulation that can be used concurrently:

- as an Automated Analytics data manipulation,
- as a table or a view,
- as a KxShell script.

### Save a Data Manipulation

1. When you have set all the parameters for the new data manipulation, click [Next](#). The [Save and Export](#) panel is displayed.
2. Set the parameters.
3. Click [Save](#). A dialog box is displayed confirming that the dataset has been saved.
4. Click [Cancel](#) to go back to the main menu.

### Save as a Standard Data Manipulation

1. Check the [Save](#) box.
2. In the field [Data Manipulation Name](#), enter a name for the newly created dataset. This name allows you to recognize and select the dataset in your database.
3. You can enter additional information in the [Description](#) field.

## Save as a Table or View

1. Check the box *Save as table or view*.
2. Choose if you want to *Save as a Table* or to *Save as a View* by selecting the appropriate option.
3. Enter the name of the new table or view in the field *Name of the Table/View*.

## Export as a KxShell Script


1. Check the box *KxShell Script Export*.
2. Use the *Browse* button corresponding to the *Folder* field to indicate where the script is to be saved.
3. In the field *KxShell Script*, enter the name of the script file.

## 2.5 Using a Data Manipulation

### 2.5.1 Using a Saved Data Manipulation

Now that you have saved your data manipulation, it can be used in another Automated Analytics feature.

1. In the application main menu, select the feature you want to use  
The *Select a Data Source* panel is displayed.
2. In the *Data Type* list, select *Database*.
3. With the *Browse* button next to the *Folder* field, select the database where you have saved the data manipulation created with the corresponding feature. If necessary, enter the login and password granting you access to the database.
4. With the *Browse* button next to the *Estimation* field, select the data manipulation.

In the database, the data manipulations created with the Semantic Layer feature are represented by the following icon: 

5. Click *Next*. If the data manipulation uses prompts, they are displayed at this time.  
You can either keep the default value, or enter a new value corresponding to your needs.
6. Click *Save* to validate the change. The *Cancel* button automatically keeps the default value.
7. Once all the prompts have been validated, the description panel is displayed.

You can now use the feature as you would with a standard data source.

## 2.5.2 Use Case Scenarios

There are three types of use cases:

- Decoration
- Filtering
- Variable Creation

### Decoration (Star Schema Use Case)

The user wants to create a model on “customer” to predict the response to a mailing campaign, but the information on “customer” is spread over several tables. One table contains reference information about the customer. One field in this table represents the region, which is in fact an identifier (foreign key) that links to another table containing demographic information about the region. A second field in the customer reference table contains an identifier that links to another table describing the associated contract. The user would like to complete the customer dataset with demographics about the customer's region and characteristics of the contract in order to see if this information can help to predict the customer response.

Technically, if the two tables can be accessed through the same database, this operation is a join. More specifically, we are interested here in “outer left” joins, where the number of lines of the reference table is not changed by the fact that information exists in the “decoration” table. If the information does not exist, then the extra fields should be brought back empty.

A different type of use case is that of aggregation, where there may be more than one line in the decoration table that can be associated with each “event” in the reference table. This situation occurs when dealing with “transactions”. An example is a “fact table” that consists of records indicating that a given customer bought a certain product on a specific date for a specified price. Of course, there can be several of such occurrences per customer. Accordingly, there is necessary to summarize them in one or more ways (for example into fields that extend a customer table) if the data is to be used in predictive modeling or segmentation. This is accomplished via aggregation (sometimes called pivoting, or transposition) and is addressed via the application aggregation modules, event logging and sequence coding.

### Filtering

The user wants to create a model predicting churn only for customers associated with a prepaid agreement. In this case, one table contains reference information about the customer; one field contains an identifier that links to another table containing information about his contract and the contract type. The user would like to keep only customers with prepaid contracts to train the model.

Technically, this can be seen as a “where” clause in an SQL “select” statement. Sometimes, the same “where” clause needs to be used for both training and apply datasets, and sometimes, the “where” clause can be used to separate the training dataset from the apply dataset. In the latter case, it is important to be able to set the value of an argument at runtime.

## Variable Creation

The user wants to add new variables that are not saved in any table but can be derived from other data on demand. When using SAP Predictive Analytics, variable creation should be primarily business-driven since the software already knows how to transform data to accommodate its algorithms. Users should add variables when they think this could provide more information for a predictive or descriptive model. For example, a ratio between two variables that has business meaning or the conversion of a birth date to an age are useful transformations that would be very difficult to infer automatically with a modeling software.

Technically, this can be decomposed into several types of variable creation:

- Using a predefined function
- Normalization
- Case-based processing

### Using a Predefined Function

A very wide range of potential functions could interest users. However, most of these functions are built using one or more of the following elements:

- Mathematical operators: such as + , \* , - , / , %
- Logical operators: and , or
- String manipulations: like , sub-string , trim , left-trim , right-trim , upper-case , lower-case .
- Mathematical functions: log , exp
- Date manipulations: year , month , day , time , minute , second , date addition , date difference

#### i Note

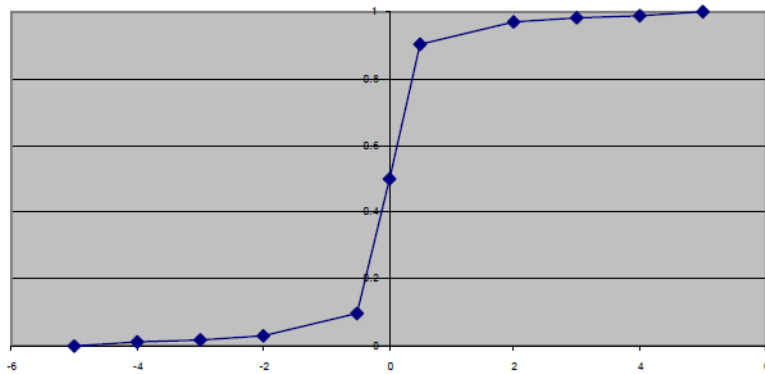
The DayOfTheWeek function returns an integer that varies from 1 to 7 where 1 stands for Sunday, 2 for Monday, ... and 7 for Saturday. This is the standard behavior of major RDBMS (including DB2, ORACLE, SQLServer, ACCESS).

A list of basic semantic layers is provided by SQL.

### Normalization

Normalization is a standard Semantic Layer primitive that appears in PMML (Predictive Model Markup Language), a data mining specification language defined by the Data Mining Group (DMG). Normalization is frequently applied to numeric variables prior to data mining and consists of a piece-wise linear transform with the resultant variable typically ranging from 0 to 1. This can be used for rank transformations, where the output represents a magnitude in terms of the approximate proportion (percentile) of values below the input value. Alternatively, a field may be converted based on how many standard deviations a value is from the fields mean. Part of normalization is also the specification of what value to use when a numeric input value is unknown or out of the range seen in the training data.

An example of normalization is graphed below. The X-axis represents the original data, the Y-axis the normalized value. The minimum and maximum X values correspond to the 0 and 1 values on the Y-axis, respectively. Each point is defined based on the normalization method. Normalization values corresponding to data values falling inbetween normalization points on the X-axis are calculated using straight-line interpolation. The graph is shaped like a S, indicating that the normalization method is designed to enhance (spread out) differences between values close to the mean while diminishing the magnitude of extreme (very high or very low) values.



## Case-based Processing

This can be decomposed into sub-operations. A lot of user-defined manipulations are based on “cases”. A classic example is a “look-up table” that is used as a dictionary to translate values from identifiers into strings, or to group values representing fine distinctions into a smaller number of more general bins. A different example is the application of complex conditions to generate outcome indicators or to segment a continuous value into sub-ranges with corresponding segment identifiers. Sometimes, such manipulations can be done through joins with a decoration table, but often it is easier to define them directly. In SAP Predictive Analytics, we have focused on the following cases:

- Look-up table  
The user specifies cases, each of which is made up of a list of discrete values and a corresponding label.
- Numeric case  
The user specifies ranges (minimum and maximum values) and corresponding labels.
- Generic case  
The user specifies different cases, where each is defined through a possibly-complex Boolean expression. The result of each expression is associated with either a user-defined variable, a field from the database, a constant or a prompt.

## 2.6 Performing a Data Manipulation Transfer

### 2.6.1 Creating a Data Manipulation

It is possible to migrate a data manipulation while performing a data transfer. This option is available via the *Toolkit* feature.

The data transfer process encompasses various steps starting from the Data Manipulation creation in *Data Manager* to performing its transfer in the Toolkit.

To Create a Data Manipulation:

1. In the *Start Panel*, select the feature *Create a Data Manipulation* in the *Data Manager* section.  
The panel *Define a New Data Manipulation* appears.
2. Select:

- a Data Type,
  - a Database Source,
  - and a Table.
3. Click *Next*.
- The panel *Data Manipulation Editor* appears.
4. Add as many options as you want to create your Data Manipulation.
  5. Click *Next*.
  6. Save the new Data Manipulation by selecting the options Save and Save as Table or View.
  7. Click Save.

The data manipulation is now ready to be transferred to another database.

### **i** Note

For further details to perform this step, *Creating a Data Manipulation*.

## 2.6.2 Transferring a Data Manipulation

1. In the *Toolkit* section of the start panel, select the feature *Perform a Data Transfer*.

The panel *Select Dataset Source* is displayed.

2. Select the source of your data by indicating the following information.

Field Name	Step
<i>Data Type</i>	Indicate whether the data you is in a text file, in a database, in a SAS file.
<i>Folder</i>	Select the folder or database where the data is located.
<i>Dataset</i>	Select the file, table or view containing the data.

3. Click *Next*.
4. In the pop-up window *Transfer a Data Manipulation*, select *Transfer Definition*.

The screen *Data Manipulation Transfer* is displayed.

5. In the area *Target Settings*, specify:

Field Name	Description
<b>Target Database</b>	database where you want to transfer your data manipulation
<b>Transfer</b>	the data manipulation

6. To define the data manipulation source in the Table Mapping area, click *Guess*.

If the new database has a similar table with a similar name, the application automatically shows the correct target table.

### ⚠ Caution

The new database must have the same source tables as the existing database. For example, if the data manipulation is derived from TableA and TableB, both tables must be migrated to the new database before performing the data manipulation transfer.

7. Edit the correct field mapping for the target by clicking the *Edit Field Mapping* icon. The *Edit Field Mapping* screen appears. By default, no mapping is defined.
8. If you have selected a correct target table at the previous step, click *Guess*.

The *Target Fields* are automatically filled in.

9. Click *OK*. The green arrow on the *Edit Field Mapping* icon indicates that the mapping is done.
10. Click *Check*

A pop-window indicates that the mapping for the transfer is correct.

11. Click *OK*.
12. Click *Next*. A pop-up message indicates that the transfer has been completed successfully.
13. Click *OK*.

The transferred data manipulation is now ready to be re-used in a new database.

## 2.6.3 Using the Transferred Data Manipulation in a New Database

1. In the Start Page, click on the option *Load an Existing Data Manipulation* in the *Data Manager* section.

The panel *Opening a Data Manipulation* appears.

2. Select:
  - *Data Base* in the field *Data Type*,
  - and in the field *Database Source*, the new database (where the Data Manipulation has been transferred) - in this example, the new database is `kxsrvmulti2_sqlserver2005`.

The transferred data manipulation is automatically displayed.

3. Select the transferred *Data Manipulation* and click *Open*.

The data manipulation loads.

4. Automatically the data manipulation interface opens with the previously defined data manipulation.

## 2.7 Annex

### 2.7.1 Arithmetic Operators

Function	Syntax/Example
<i>Absolute</i>	Returns the numerical value regardless of its sign <code>absolute(numeric)</code> <i>Example:</i> <code>absolute(1) =&gt; 1</code>
<i>Add</i>	<code>add(numeric)</code> <i>Example:</i> <code>add(1,2,3) =&gt; 6</code>
<i>Division</i>	<code>divide(numeric, numeric)</code> <i>Example:</i> <code>divide(64,2) =&gt; 32</code>
<i>Exponential</i>	<code>exp(numeric)</code> <i>Example:</i> <code>exp(16) =&gt; 8886110</code>
<i>Logarithm</i>	<code>ln(numeric)</code> <i>Example:</i> <code>ln(13) =&gt; 2,56495</code>
<i>Max.</i>	<code>greatest2(numeric, numeric)</code> <i>Example:</i> <code>greatest2(12,13) =&gt; 13</code>
<i>Maximum of a List of Values</i>	<code>greatestN(list of numerics)</code> <i>Example:</i> <code>greatestN(15,78,96,53,46,105) =&gt; 105</code>
<i>Min</i>	<code>least2(numeric, numeric)</code> <i>Example:</i> <code>least2(12,125) =&gt; 12</code>



Function	Syntax/Example
<i>Minimum of a List of Values</i>	<p>leastN(list of numerics)</p> <p><i>Example:</i></p> <p>leastN(15,78,96,53,46,105) =&gt; 15</p>
<i>Modulo</i>	<p>Returns the remainder of an integer division</p> <p>modulo(ModRightOperand:integer, ModLeftOperand:integer)</p> <p><i>Example:</i></p> <p>modulo(117,17) =&gt; 15</p>
<i>Multiply</i>	<p>multiply(numeric)</p> <p><i>Example:</i></p> <p>multiply(2,4) =&gt; 8</p>
<i>Negate</i>	<p>Returns the opposite value</p> <p>negate(numeric)</p> <p><i>Example:</i></p> <p>negate(3) =&gt; -3</p> <p>negate(-2) =&gt; 2</p>
<i>Ntile</i>	<p>Returns the quantile number</p> <p>ntile(numeric, integer)</p> <p><i>Example:</i></p> <p>ntile(age,4)</p> <p>age is the variable, 4 is the number of quantiles.</p>
<i>Random Value</i>	<p>Returns a random number between 0 and 1</p> <p>random()</p>
<i>Round</i>	<p>round(number)</p> <p><i>Example:</i></p> <p>round(15.2) =&gt; 15</p>
<i>Round Down</i>	<p>floor(number)</p> <p><i>Example:</i></p> <p>floor(156.3) =&gt; 156</p>

Function	Syntax/Example
<i>Round Up</i>	ceil(number)  <i>Example:</i>  ceil(156.3) => 157
<i>Sign</i>	sign(numeric)  <i>Example:</i>  sign(123456) => 1 sign(-123456) => -1
<i>Square Root</i>	sqrt(numeric)  <i>Example:</i>  sqrt(16) => 4
<i>Subtract</i>	diff(numeric, numeric)  <i>Example:</i>  diff(15, 8) => 7
<i>Zero if NULL</i>	Replaces the NULL value by zero  ZeroIfNull(number)  <i>Example:</i>  If the database contains NULL values, they are converted to 0 .  1+NULL becomes 1+0

## 2.7.2 Boolean Operators

Function	Syntax/Example
<i>And</i>	Returns "1" if all the operands are true, otherwise returns "0"  and(<boolean>)  <i>Example:</i>  and(age > 18)
<i>Does not Start with</i>	Returns "1" if the first string does not start with the second one  notStartsWith(Test:string, Candidate:string)  <i>Example:</i>  notStartsWith("KxTimestamp", "Kx") => 0

Function	Syntax/Example
<i>Equal</i>	<p>equal(LeftOperand:any, RightOperand:any)</p> <p><i>Example:</i></p> <p>equal(15,15) =&gt; 1</p>
<i>Greater than</i>	<p>Returns "1" if the left operand is greater than the right one</p> <p>greater(LeftOperand:any, RightOperand:any)</p> <p><i>Example:</i></p> <p>greater(15,16) =&gt; 0</p> <p>greater(16,15) =&gt; 1</p>
<i>Greater than or Equal to</i>	<p>greaterOrEqual(any, any)</p>
<i>Is Null</i>	<p>isNull(any)</p> <p><i>Example:</i></p> <p>isNull("or") =&gt; 0</p>
<i>Is True</i>	<p>isTrue(boolean)</p> <p><i>Example:</i></p> <p>isTrue(age&lt;30)</p> <p>Checks if age &lt; 30 is true.</p> <p>If true, it returns "1" otherwise it returns "0".</p>
<i>Is in List</i>	<p>Returns "1" if the selected field is in the selected values</p> <p>isInList(Field:any, &lt;Values:any&gt;)</p>
<i>Is in Range</i>	<p>Checks if a numeric operand is between two numeric values</p> <p>isInRange(Value Operand:numeric, Low Border Operand:numeric, High Border Operand:numeric)</p>
<i>Is not Null</i>	<p>isNotNull(any)</p>
<i>Is not in List</i>	<p>Returns "1" if the selected field is not in the selected values</p> <p>notIsInList(Field:any, &lt;Values:any&gt;)</p> <p><i>Example:</i></p> <p>notIsInList(dispatch_id, 12, 16)</p> <p>if dispatch_id = 1 the result is true; if dispatch_id = 12 or dispatch_id = 16 the result is false.</p>
<i>Is not in Range</i>	<p>notIsInRange(Value Operand:numeric, Low Border Operand:numeric, High Border Operand:numeric)</p>

Function	Syntax/Example
<i>Like</i>	<p>Indicates whether the given string matches the given pattern</p> <p>Use % to determine the place of the pattern in the string.</p> <p>For example, if the pattern ends with %, you are looking for a string that starts with the pattern. If the pattern starts with %, you are looking for a string that ends with the pattern. You can use more than one % in the pattern.</p> <p>like(string, pattern)</p> <p><i>Example:</i></p> <p>like(my_variable, 'kxen%')</p> <p>if my_variable = "kxen_model" the result is true; if my_variable = "model_kxen" the result is false.</p>
<i>Not</i>	not(boolean)
<i>Not Equal</i>	<p>Returns "1" if the operands are different</p> <p>different(Left Operand:any, Right Operand:any)</p>
<i>Not Like</i>	<p>Returns "1" if the given string does not match the given pattern</p> <p>Use % to determine the place of the pattern in the string.</p> <p>For example, if the pattern ends with %, you are looking for a string that starts with the pattern. If the pattern starts with %, you are looking for a string that ends with the pattern. You can use more than one % in the pattern.</p> <p>notLike(Test Operand:string, Candidate Operand:string)</p>
<i>Or</i>	<p>Returns "0" if all operands are false, otherwise returns "1"</p> <p>or(&lt;boolean&gt;)</p>
<i>Smaller than</i>	less(Left Operand:any, Right Operand:any)
<i>Smaller than or Equal to</i>	lessOrEqual(any, any)
<i>Start with</i>	<p>Returns "1" if the first string starts with the second one</p> <p>startsWith(Test:string, Candidate:string)</p>

## 2.7.3 Date Operators

Function	Syntax/Example
<i>Builds Date</i>	Builds a date from a day, a month and a year <code>dateToImplode(y,m,d)</code> <i>Example:</i> <code>dateToImplode(1,6,8) =&gt; 2001-06-08</code>
<i>Current Date</i>	Returns the current DateTime <code>now()</code> <i>Example:</i> <code>now() =&gt; 2012-09-28 10:17:12</code>
<i>Day Extraction</i>	<code>datePartDay(yyyy-mm-dd)</code> <i>Example:</i> <code>datePartDay(2012-02-08) =&gt; 8</code>
<i>Days Difference</i>	<code>dateDiffNbDays(yyyy-mm-dd, yyyy-mm-dd')</code> <i>Example:</i> <code>dateDiffNbDays(2012-02-09, 2012-02-02) =&gt; 7</code>
<i>Hour Extraction</i>	<code>datePartHour("hh:mm:ss")</code> <i>Example:</i> <code>datePartHour("10:05:30") =&gt; 10</code>
<i>Minutes Extraction</i>	<code>datePartMinute("hh:mm:ss")</code> <i>Example:</i> <code>datePartMinute("10:05:30") =&gt; 5</code>
<i>Month Extraction</i>	<code>datePartMonth(yyyy-mm-dd)</code> <i>Example:</i> <code>datePartMonth(2010-02-05) =&gt; 2</code>
<i>Seconds Extraction</i>	<code>datePartSecond("hh:mm:ss")</code> <i>Example:</i> <code>datePartSecond("10:05:30") =&gt; 30</code>

Function	Syntax/Example
<i>Shifts Date by Day</i>	Adds/subtracts n days to/from a given date <code>dateAddDay(yyyy-mm-dd, integer)</code> <i>Example:</i> <code>dateAddDay(2010-03-02,20) =&gt; 2010-03-22</code>
<i>Shifts Date by Hour</i>	Adds/subtracts n hours to/from a given date <code>dateAddHour(yyyy-mm-dd, integer)</code> <i>Example:</i> <code>dateAddHour(2012-02-05,35) =&gt; 2012-02-06 23:00:00</code>
<i>Shifts Date by Minutes</i>	Adds/subtracts n minutes to/from a given date <code>dateAddMinute(yyyy-mm-dd, integer)</code> <i>Example:</i> <code>dateAddMinute(2012-02-05, 35) =&gt; 2012-02-5 12:35:00</code>
<i>Shifts Date by Month</i>	Adds/subtracts n months to/from a given date <code>dateAddMonth(yyyy-mm-dd, integer)</code> <i>Example:</i> <code>dateAddMonth(2012-02-05,2) =&gt; 2012-04-05</code>
<i>Shifts Date by Seconds</i>	Adds/subtracts n seconds to/from a given date <code>dateAddSecond(yyyy-mm-dd, integer)</code> <i>Example:</i> <code>dateAddSecond(2012-02-05, 3) =&gt; 2012-02-05 12:00:03</code>
<i>Shifts Date by Year</i>	Adds/subtracts n years to/from a given date <code>dateAddYear(yyyy-mm-dd, integer)</code> <i>Example:</i> <code>dateAddYear(2012-02-05, 3) =&gt; 2015-02-05</code>
<i>Weekdays Extraction</i>	<code>datePartWeekDay(yyyy-mm-dd)</code> <i>Example:</i> <code>datePartWeekDay(2012-02-05) =&gt; 1</code>
<i>Year Extraction</i>	<code>datePartYear(yyyy-mm-dd)</code> <i>Example:</i> <code>datePartYear(2012-02-05) =&gt; 2012</code>

## 2.7.4 Miscellaneous Operators

Function	Syntax/Example
<i>Coalesce</i>	Returns the first non null value in a list  coalesce(<any>)  <i>Example:</i>  coalesce("", "", 2,3) => 2
<i>Constant</i>	Creates a new constant  litteral(any)
<i>Void Date</i>	Returns an empty column of date storage type  nullAsDate()
<i>Void DateTime</i>	Returns an empty column of DateTime storage type  nullAsDateTime()
<i>Void Integer</i>	Returns an empty column of DateTime storage type  nullAsInteger()
<i>Void Number</i>	Returns an empty column of number storage type  nullAsNumber()
<i>Void String</i>	Returns an empty column of string storage type  nullAsString()
<i>If Valid</i>	Returns the value of the second argument if the first argument is true, otherwise returns null  ifValid(boolean, any)

## 2.7.5 String Operators

Function	Syntax/Example
<i>Concatenate</i>	concat(<string>)  <i>Example:</i>  concat("data", "mining") => datamining
<i>Does not Start with</i>	Returns true if the first string does not start with the second one  notStartsWith(string, string)

Function	Syntax/Example
<i>Left Pad</i>	<p>Returns the string argument, left-padded with spaces</p> <p>Note - The spaces are not displayed when viewing the data.</p> <p><code>lpad(string, integer)</code></p> <p><i>Example:</i></p> <p><code>lpad("example",4) =&gt; "mple"</code></p> <p><code>lpad("kxen",7) =&gt; " kxen"</code></p>
<i>Left Trim</i>	<p>Removes spaces before a string</p> <p><code>lTrim(string)</code></p>
<i>Leftmost Substring</i>	<p>Returns the specified leftmost number of characters</p> <p><code>left(string, integer)</code></p> <p><i>Example:</i></p> <p><code>left("example",2) =&gt; ex</code></p>
<i>Like</i>	<p>Indicates whether the given string matches the given pattern</p> <p>Use % to determine the place of the pattern in the string.</p> <p>For example, if the pattern ends with %, you are looking for a string that starts with the pattern. If the pattern starts with %, you are looking for a string that ends with the pattern. You can use more than one % in the pattern.</p> <p><code>like(string, pattern)</code></p>
<i>Lower Case</i>	<p>Converts all characters from a string to lower case</p> <p><code>lowerCase(string)</code></p> <p><i>Example:</i></p> <p><code>lowerCase("exampleE") =&gt; example</code></p>
<i>Not Like</i>	<p>Returns "1" if the given string does not match the given pattern</p> <p>Use % to determine the place of the pattern in the string.</p> <p>For example, if the pattern ends with %, you are looking for a string that starts with the pattern. If the pattern starts with %, you are looking for a string that ends with the pattern. You can use more than one % in the pattern.</p> <p><code>notLike(string, pattern)</code></p>
<i>Position</i>	<p>Returns the index of the first occurrence of the substring</p> <p><code>position(posOperand:string, posOperandSearch:string)</code></p>



Function	Syntax/Example
<i>Repeat</i>	Repeats a string value a specified number of times repeat(string, integer) <i>Example:</i> repeat("example", 3) => exampleexampleexample
<i>Replace</i>	Replaces all occurrences of a specified string value with another string value replace(tested string, string to replace, replacement string)
<i>Right Pad</i>	Returns the string argument, right-padded with spaces Note - The spaces are not displayed when viewing the data. rpad(string, integer) <i>Example:</i> rpad("example", 2) => "ex" rpad("kxen",7) => "kxen "
<i>Right Trim</i>	Removes spaces after a string rTrim(string)
<i>Rightmost Substring</i>	Returns the specified rightmost number of characters right(string, integer) <i>Example:</i> right("example",2) => "le"
<i>Start with</i>	Returns true if the first string starts with the second one startsWith(StartWithStringTest:string, StartWithStringCandidate:string)
<i>String Length</i>	Returns the number of characters in a string stringLength(string)
<i>Substring</i>	Selects the n characters of a string from a selected character subStr(string, SubStrIntegerStart:integer, SubStrIntegerLength:integer) <i>Example:</i> subStr("example", 2,3) => xam
<i>Trim</i>	Removes spaces around a string trim(string)

Function	Syntax/Example
<i>Upper Case</i>	Converts all characters from a string to upper case <pre>upperCase(string)</pre> <i>Example:</i> <pre>upperCase("example") =&gt; EXAMPLE</pre>

## 2.7.6 Conversion Operators

Function	Syntax	Example
<i>Converts Boolean into an Integer</i>	<code>booleanToInt (boolean)</code>	<pre>booleanToInt ("True") =&gt; 1</pre> <pre>booleanToInt ("False")</pre> <pre>=&gt; 0</pre>
<i>Converts Boolean to String</i>	<code>booleanToString (boolean)</code>	<pre>booleanToString(1)</pre> <pre>=&gt; true</pre>
<i>Converts Date to DateTime</i>	<code>dateToDateTime (yyyy-mm-dd)</code>	<pre>dateToDateTime (1900-01-01)</pre> <pre>=&gt; 1900-01-01 00:00:00</pre>
<i>Converts Date to Integer</i>	Returns the number of days elapsed since 1900-01-01 <code>dateToInteger (yyyy-mm-dd)</code>	<pre>dateToInteger (2010-01-01)</pre> <pre>=&gt; 40177</pre>
<i>Converts Date to String</i>	<code>dateToString (yyyy-mm-dd)</code>	
<i>Converts DateTime to Date</i>	<code>dateTimeToDate (yyyy-mm-ss hh:mm:ss)</code>	<pre>dateTimeToDate (1900-01-01 00:00:00)</pre> <pre>=&gt; 1900-01-01</pre>
<i>Converts DateTime to String</i>	<code>dateTimeToString (yyyy-mm-dd hh:mm:ss)</code>	
<i>Converts Integer to Boolean</i>	<code>intToBoolean (integer)</code>	<pre>intToBoolean (0) =&gt; 0, else =&gt; 1</pre>
<i>Converts Integer to Number</i>	<code>intToNumber (integer)</code>	
<i>Converts Integer to String</i>	<code>intToString (integer)</code>	
<i>Converts Number to Integer</i>	<code>numberToInt (number)</code>	<pre>numberToInt (56.3) =&gt; 56</pre> <p><b>Note</b> - The values are rounded down.</p> <pre>numberToInt (56,9) =&gt; 56</pre>
<i>Converts Number to String</i>	<code>numberToString (number)</code>	
<i>Converts String to Boolean</i>	<code>stringToBoolean (string)</code>	<b>Note</b> - If "true" => 1, else => 0

Function	Syntax	Example
<i>Converts String to Date</i>	<p><code>stringToDate (string)</code></p> <p><b>Note</b> - If the content of the string does not correspond to a date, it leads to an error.</p>	
<i>Converts String to DateTime</i>	<p><code>stringToDateTime (string)</code></p> <p><b>Note</b> - If the content of the string does not correspond to a DateTime, it leads to an error.</p>	
<i>Converts String to Integer</i>	<p><code>stringToInt (string)</code></p> <p><b>Note</b> - If the content of the string does not correspond to an integer, it leads to an error.</p>	
<i>Converts String to Number</i>	<p><code>stringToNumber (string)</code></p> <p><b>Note</b> - If the content of the string does not correspond to a number, it leads to an error.</p>	
<i>Converts an ABAP date in string format to a Date object</i>	<p><code>abapDateToDate (string)</code></p> <p><b>Note</b> - If the content of the string does not correspond to a date in ABAP format (YYYYMMDD), it leads to an error.</p>	<code>abapDateToDate ("20161231")</code> => 2016-12-31
<i>Converts an ABAP time in string format to a DateTime object</i>	<p><code>abapTimeToDateTime (string)</code></p> <p><b>Note</b> - If the content of the string does not correspond to a time in ABAP format (HHMMSS), it leads to an error.</p>	<code>abapTimeToDateTime ("100504")</code> => 10:05:04
<i>Converts an ABAP timestamp in string format to a DateTime object</i>	<p><code>abapTimestampToDateTime (string)</code></p> <p><b>Note</b> - If the content of the string does not correspond to a timestamp in ABAP format (YYYYMMDDHHMMSS), it leads to an error.</p>	<code>abapTimestampToDateTime ("20161231100504")</code> => 2016-12-31 10:05:04

# 3 Data Manipulation Scenario

## 3.1 About Data Manipulation Scenario

This section presents the data manipulation functions offered by through a scenario. The first part provides definitions of essential concepts as well as the methodology to be applied when using these functions. The second part guides you through the steps defined in the methodology by using a practical scenario as an example.

## 3.2 Essential Concepts

### 3.2.1 Data Manager Semantic Layer

The Data Manager semantic layer is made of three elements:

- Data manipulation features such as filters, join attributes, new attributes computation, aggregates, performance indicators definition
- Analytical dataset methodology
- Metadata management, which allows storing, sharing and easily re-using the data descriptions

### 3.2.2 Entity

An **entity** is the object of interest of any analytical task. It can be a customer, a product, a store, and is usually identified with a single identifier that can be used throughout the data repositories. Entities are usually associated with a state model describing the life cycle of such an analytical object of interest.

#### i Note

This is a technical constraint: entities **MUST** be uniquely identified.

### 3.2.3 Time-stamped Population

A **time-stamped population** is a set of entities at a particular point in time. For example all customers as defined by `customer_ID` (the entities) as of January 1, 2008 (the time stamp).

It is a list of pairs <identifiers; time stamps>: the semantic meaning of such a construct can be associated with snapshots of the entities and a given time: in general terms, a given entity may be represented at different time stamps in a single time-stamped population.

### 3.2.4 Analytical Record

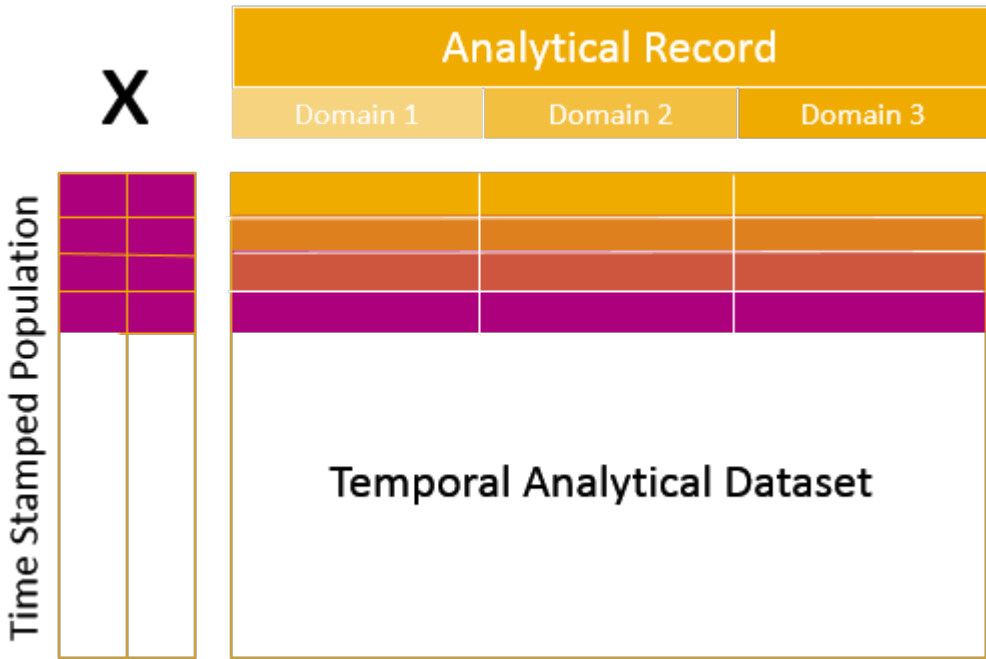
An **analytical record** is a logical view of all attributes corresponding to an **entity**. An analytical record may be decomposed into domains that group **attributes** related to each other. For example, in CRM, an analytical record can have a demographic domain and a behavioral domain.

### 3.2.5 Analytical Dataset

An analytical dataset is a dataset generated by manipulating data through merges, creation of new fields, application of filters, and so on.

### 3.2.6 Temporal Analytical Dataset

A **temporal analytical dataset** is a special case of analytical dataset. It is the product of a **time-stamped population** by an **analytical record**, the result of this operation can be seen as a virtual table containing attributes values associated with identifiers in relation with the time stamp.



In other words, a temporal analytical dataset contains 'photos' or snapshots of a given list of entities taken at a given time. This time can be different for each entity, and an entity can be associated with several 'photos'.

#### i Note

The analytical datasets are used to train predictive/descriptive models and to apply these models.

### 3.2.7 Performance Indicator (PI)

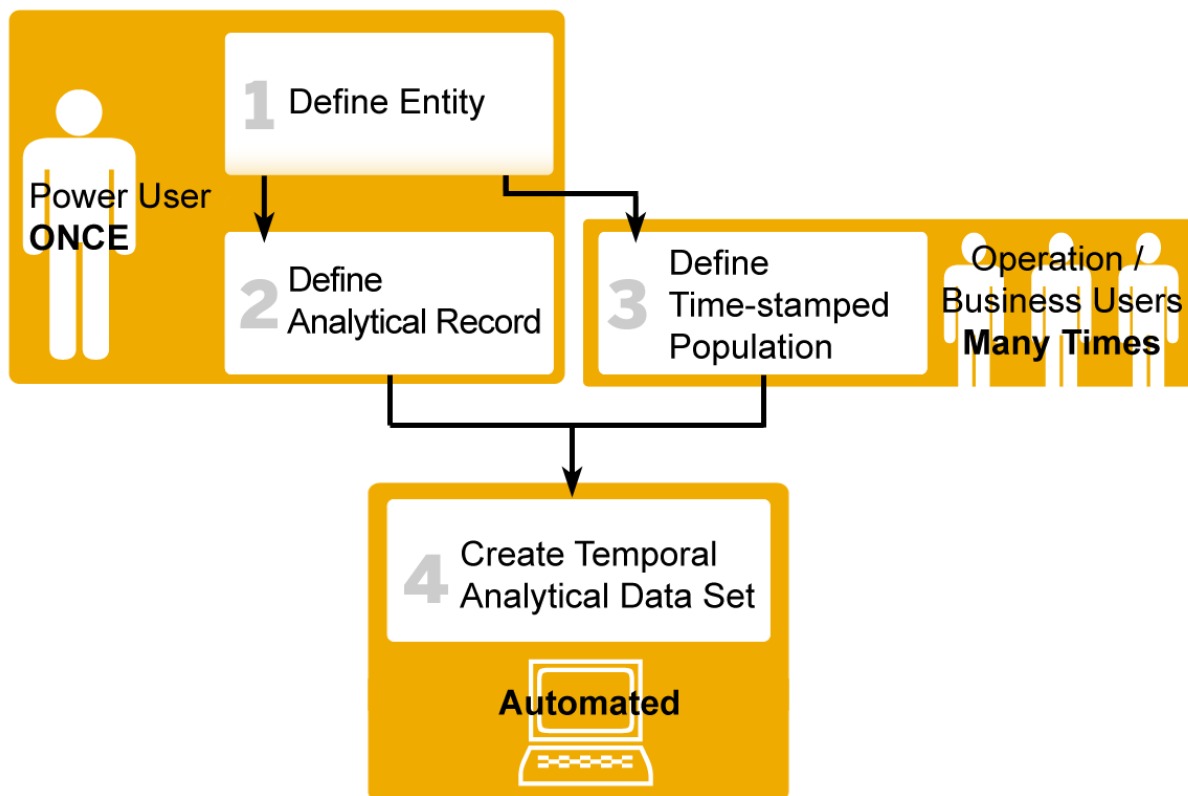
The **Performance Indicators (PIs)** help organizations achieve organizational goals through the definition and measurement of progress.

The purpose of defining PIs is to have a common definition of a metric across multiple projects. A metric like "customer value" could easily be defined in several different ways, leading to confusing or contradictory results from one analysis to the next. Shared PIs ensure consistency across analysts and projects over time. The key indicators are agreed upon by an organization and are indicators which can be measured and will reflect success factors. The PIs selected must reflect the organization's goals, they must be the key to its success, and they must be measurable.

### 3.2.8 Methodology

The methodology used in Data Manager has been designed around systematization and automation of the creation of temporal analytical datasets. The key is to define all attributes describing a customer through expressions that are not based on basic information only, but also on a **time stamp**. In other words, all attributes are expressions that can be computed in order to get the value of that attribute for a given customer **at a given time**.

This methodology is made of four steps:



1. **Define the entity** that will be the main object of the analytical tasks.
2. Define the **analytical record** that describes the entity's attributes on which predictive and descriptive models can be trained.

#### i Note

The first two steps only need to be defined **once** by a power user.

3. **Define the time-stamped population** to be analyzed. This task will be executed by **operation or business analysts** and can be redone **several times** to adapt the population to your needs.
4. The last step is the **automated** creation of the temporal analytical dataset consisting of the product between the analytical record, which determines the number of columns, and the time-stamped population, which determines the number of lines. You can also **create a target on the fly** when it does not exist in the original data.

### 3.3 Application Scenario: Segmented Cross-sell in Retail Banking

In this scenario, you are a data mining analyst in a large retail bank. Your role is to translate any business question (in this case, cross-sell) into successive data mining functions/models. You are part of a center of excellence in analytics. This group provides support for teams such as the marketing team when developing

advanced analytics projects. Your project is to increase the number of credit cards subscribed by present customers.

At the start of year 2008 (January 7th), the marketing management team decided on an objective to increase the number of credit cards subscribed by present customers of the bank. They decided to achieve this objective through the use of advanced analytics. Their goal is to engage the marketing operations of the retail bank on the path to 'competing on analytics'.

## 3.3.1 Creating the Performance Indicator

Creating a performance indicator will allow you to measure the performance of your company regarding the goal you want to achieve.

### 3.3.1.1 Creating the Calendar Table

For a given date, the Calendar table allows getting specific information relative to the date that is complex to compute such as the day of the week, the week number in the year, the quarter and so on. This information is needed to compute the performance indicators.

#### i Note

To execute the scenario, use the following settings:

- **ODBC Sources:** the database in which you have imported the PKDD data.
- **Start Date:** 1990-01-01 00:00:00
- **End Date:** 2020-01-01 00:00:00

1. In the *Windows Start* menu, select the option **Programs > SAP Business Intelligence > Predictive Analytics > Control Panel**.
2. In the list *ODBC Sources*, select the database in which you want to create the date look-up table.
3. If the database is password protected, enter the user name and password respectively in the fields *User* and *Password*.
4. Use the fields *Start Date* and *End Date* to define the time range that should be covered by the date look-up table. This time range needs to cover the content of your database.
5. Click the button *Create Table*. A message box is displayed indicating that the table is being created.
6. Close the *Control Panel*.

### 3.3.1.2 Creating the Performance Indicator

A simple performance indicator can be created to visualize the evolution of the credit cards sales each month. You will need to create an aggregate counting the number of credit cards bought in the past month, and shift it one month forward every month.



## i Note

To execute the scenario, use the following settings:

Shifting Aggregate:

- *Event Table*: **credit\_card**
- *Date Column*: **card\_date**
- *Function*: **count**
- *Target Column*: **card\_id**
- *Aggregation Period*: **1 Month in the Past**
- *Name*: **count\_creditcard**

Period:

- *Compute for a period of*: **730 days**
- *Ending at*: **1998-08-01 12:00:00**

Advanced filters:

- Create the condition: **date\_day == 1**
- Performance indicator name: **PI\_CountCreditCard**

To create a performance indicator, proceed as follows:

1. In Automated Analytics start panel, select the option *Create or Edit Data Manager Objects* in the *Data Manager* section.
2. Click the *Browse* button to select the database in which your data are stored.
3. If the database is protected, enter the username in the *User* field and the password in the *Password* field. Click *OK* to validate your selection.
4. Click *Next* to display the *Data Manager* panel.
5. In the + menu, select **Performance Indicator > Based on the Calendar Table** to open the *Performance Indicator Editor*.
6. In the *Fields* tab, click the + button and select *New Shifting Aggregate*.
7. In *Event Table*, select the table containing the data for which you want to create the aggregate.
8. In *Date Column*, select the column which contains the date the aggregate should be based on.
9. In the *Function* drop-down list, select the aggregation operation you want to apply. The following operations are available:

Functions	Description	Returned Values
<i>Count</i>	computes the number of not null values found in the selected column of the Event table	number of not null occurrences in the selected column

## i Note

This may differ on some databases and behave as a count (no operand).

Functions	Description	Returned Values
<i>Count(no operand)</i>	computes the total number of lines in the Event table	total number of occurrences
<i>Sum</i>	compute the sum	sum
<i>Average</i>	compute the mean	mean
<i>Min</i>	identifies the minimum value	minimum value
<i>Max</i>	identifies the maximum value	maximum value

10. In *Target Column*, select the column on which you want to apply the desired aggregation function.
11. In *Aggregation Period*, select the period on which you want to compute the aggregate.
12. Click *Validate*.  
A pop-up window opens up, letting you to set a name for the new field you are creating.
13. In *Name* field, enter the name you want to give to the field.
14. Click the *Filters* tab.
15. Check *Period*.
16. Use the provided fields to select the period over which you want to compute the performance indicator.  
By default, the application will compute one point for each line in the table used to create the performance indicator, that is, either the Calendar table or the time series you have based your indicator on. If you want to compute only certain points, for example one point every month, you will have to create an advanced filter to keep only the dates that interest you. As an example, if you want a point at the beginning of each month, you will create the filter `date_day == 1`.
17. Check *Advanced Filters*.
18. Click *New Condition* to open the expression editor.
19. Define the filter you want to apply.
20. Click *OK*.  
The expression editor closes and the defined condition appears in *Filter Condition List*.
21. Click the tab *View Data* to display the performance indicator as a graph.
22. Click *Next* to open the *Save and Export* panel.
23. In *Data Manipulation Name*, enter a name for your performance indicator.
24. Use the *Description* field to describe the performance indicator and click *Save*.

## 3.3.2 Preparing the Data

### 3.3.2.1 Creating the Entity

In this particular case, even if the problem statement for the project speaks about selling credit card to '*customer*', since credit cards are associated with accounts, you have to determine, together with everyone involved in this project, if the entity for this project is the '*account*' (on which the credit card is attached), the '*customer*' (the only one that can receive a piece of mail describing the offer), or the association between the

account and the customer (which is called a '*Disposition*' in the database schema), describing the role of the person with the account. There are two roles in our example: one is the 'owner' of the account, and another possible role is a 'user' of the account.



To define an entity, you must take into account the fact that this entity makes business sense, that it can be characterized in terms of attributes, that it can be associated with predictive metrics in relation with the tasks you want to perform (in this case the number of gold credit cards). Defining an entity is not a minor operation. Entities may be used in many projects and cannot be changed without an impact analysis on all the deployed processes using this entity.

The entity selected is the '*Disposition*'.

### **i** Note

To execute this scenario, use the following settings:

- *Name of the entity*: **entity\_disposition**
- *Table*: **disposition**
- *Id Field*: **disp\_id**
- Do not define a *Filter*

1. In the **+** menu of the *Data Manager* panel, select *Entity*.  
The *Edit Entity* panel opens up.
2. In *Name*, enter the name of the new entity.
3. In *Description*, enter a description of the new entity.
4. To select the table on which you want to build the new entity, click *Browse* and select a table in the dialog box that appears. Click *OK* to validate your choice.
5. In the *Id Field* drop-down list, select the column containing the entity identifier.
6. **Optional:** To add a filter to the entity in order to remove some lines depending on a specific condition, click *Edit Filter*. The existence of a filter is indicated as listed below:
  - : no filter has been defined.
  - : a filter already exists.
7. Click *Next* to validate the creation or changes on the entity.
8. If an entity with the same name already exists, a dialog box appears asking you to choose what should be done.
  - Click *Yes* to save the entity as a new version of the existing one.
  - Click *No* to overwrite the existing entity.

The *Data Manager* panel appears, listing the newly created entity and the default time-stamped population associated with it.

### Caution

**Because of technical constraints**, the default time-stamped population must NOT be deleted **as long as the entity exists**.

## 3.3.2.2 Editing a Filter

1. Click *Edit Filters*.  
The panel allowing you to define filters opens.
2. Click *New Condition* to add a new filter.  
The expression editor appears.
3. You can either use the offered *Functions* and *Variables* by double-clicking them, or you can enter directly a valid expression. The indicator above the *Messages* field indicates whether the expression is valid or not.
4. Click *OK* to create the condition and display the panel listing the conditions.
5. Once all the conditions you need have been created, click *OK* to validate the filter.

## 3.3.2.3 Creating the Analytical Record

Advanced analytics best practice next step consists of creating an overall view of the entities (when this entity is a customer, it is sometimes called 'Customer Analytical Record', or '360 degree view of the customer', or even 'Customer DNA'). This view characterizes entities by a large number of attributes (the more, the better) which can be extracted from the database or even computed from events that occurred for each of them.

The list of all attributes corresponds to what is called an 'Analytical Record' of the entity 'disposition'. This analytical record can be decomposed into domains as listed below:

- Demographic
- Geo-demographic
- Complaints history
- Contacts history
- Products history
- Loan history
- Purchase history (coming from the transaction events)
- Segments
- Scores

### i Note

To execute this scenario, use the following settings:

- Analytical Record *Name*:

```
AR_ClientGeoDemo
```

- *Based on Entity*: *entity\_disposition*.
- Edit the following attributes under the *Merge* tab:

Add information on...	Target table	Source Field	Target Field
the disposition	<i>disposition</i>	<i>kxld</i>	<i>disp_id</i>

Add information on...	Target table	Source Field	Target Field
the client	<i>client</i>	<i>client_id</i>	<i>client_id</i>
geographic and demo-graphic data	<i>geodemo</i>	<i>home_geocode_id</i>	<i>geocode_id</i>
the credit card information	<i>credit_card</i>	<i>disp_id</i>	<i>disp_id</i>
the loan information	<i>loan</i>	<i>account_id</i>	<i>account_id</i>

1. In the *Data Manager* panel, click the + menu and select *Analytical Record*. The *Edit Analytical Record* panel appears.
2. Enter the name and the description of the new analytical record.
3. In the *Based on Entity* drop-down list, select the entity that will be used to create the new analytical record.
4. Click *Edit Attributes* to add computed fields or fields from other tables to the new analytical record. For a more detailed procedure, see **Adding Fields from Other Tables: Creating a Merge** below.
5. **Optional:** you can associate the analytical record attributes to specific domains. By default, the attributes are associated to a domain named after the table they are originated from.
6. Click *Next* to validate the creation or edition of the analytical record.
7. If an analytical record with the same name already exists, a dialog box is displayed asking you to choose what should be done:
  - Click *Yes* to save the analytical record as a new version of the existing one,
  - Click *No* to overwrite the existing analytical record.

### 3.3.2.4 Adding Fields from Other Tables: Create a Merge

1. Click the *Merge* tab.
2. Select the source field in the *Source Field* list.
3. Select the table to be joined in the *Target Table*.  
If the selected table contains fields corresponding to the source field type, they are displayed in the *Target Field* list, otherwise a message *No Fields Available* pops up.
4. Select the target field in the list.
5. Click *New Merge* to create the merge.
6. Repeat the procedure for each merge you want to create. The existing merges are displayed in the upper part of the panel.

### 3.3.2.5 Editing a Time-stamped Population

A default time-stamped population, named `entity_dispositionPopulation`, has been automatically created when you have created the `entity_disposition` entity. In this scenario, you need to modify this population so that it contains all active dispositions for which no cards have been issued on January 1, 1998. An

active disposition is a disposition for which at least one transaction has been performed in the 6 months prior to the selected date.


### i Note

To execute the scenario, use the following settings:

- *Name*:

```
ActiveDispositionNoCard@Date
```

- *Based on Entity*: entity\_disposition.
- *Time Stamp*: KxTimeStamp

1. In the *Data Manager* panel, select the time-stamped population you want to edit.
2. Click *Edit* () to open the *time-stamped population edition panel*.
3. In *Name*, enter the name of the time-stamped population you want to create.
4. Click *Edit Filters* to open the *Time-stamped Population Editor*.

## 3.3.2.6 Editing the Time Stamp Prompt

This time stamp created in the default time-stamped population is a prompt for which the default value is the date of its creation.


### i Note

To execute this scenario, use the following settings:

- *Default value*:

```
1998-01-01 12:00:00
```

- *Prompt Heading*: do not change the default value.

1. In the *Time-stamped Population Editor*, click the *Prompts* tab.
2. Select the prompt you want to edit.
3. Click *Edit* () to open the *Prompt Editor*.
4. In *Default Value*, enter the value you want.
5. In *Prompt Heading*, enter the text you want to see when the prompt is displayed. Click *OK* to validate the changes.  
A dialog box appears, asking you if you want to overwrite the existing prompt.
6. Click *Yes* to apply the changes to the prompt.

### 3.3.2.7 Creating an Aggregate

To select active dispositions, you need first to create an aggregate counting the number of transactions there has been for each disposition on the 6 months before the date of interest.

#### Note

<i>Events Table</i>	transaction
<i>Events Date Column</i>	date_transaction
<i>Reference Table Key / Events Table Key</i>	account_id
<i>Function</i>	Count
<i>Target Column</i>	trans_id
<i>Period Settings</i>	<i>Single Period</i>
<i>Period Settings / Start Date</i>	create a new field KxTimeStampMinus6M by using the Date Operator Shift Date By Month on the field KxTimeStamp, and shift back 6 months. You should get the following formula: dateAddMonth(KxTimeStamp, -6).
<i>Period Settings / End Date</i>	KxTimeStamp
<i>Name</i>	TransactionsNumberLast6M

To execute this scenario, use the following settings:

1. Click the *Fields* tab.
2. In the + menu, select *New Aggregate*.  
The *Define an Aggregate* panel appears
3. In *Event Tables Selection* under the *Aggregation Settings* tab, use the *Table* dropdown list to select the table that contains the event data you want to aggregate.
4. If needed, enter an alias for the table in *Alias*.
5. In *Date Column*, select the column that contains the event dates.
6. In the *Join Keys* section, select the reference table to add in the *Reference Table Key* dropdown list.
7. Select the events table to add in the *Events Table Key* dropdown list.
8. In the *Aggregate Operation Specification* section, use the *Function* dropdown list to specify the operation you want to do in the aggregate.  
The following operations are available:

Functions	Description	Returned Values
<i>Count</i>	computes the sum	sum
<i>Average</i>	computes the mean	mean
<i>Min</i>	identifies the minimum value	minimum value
<i>Max</i>	identifies the maximum value	maximum value

Functions	Description	Returned Values
<i>Exists</i>	checks if at least one event exists for the current reference	0 if no event has been found 1 if at least one event has been found
<i>NotExists</i>	checks if no event exists for the current reference	0 if at least one event has been found 1 if no event has been found
<i>First</i>	identifies the first occurrence  <b>i Note</b> Needs a date column	value of the first chronological occurrence for the current reference
<i>Last</i>	identifies the last occurrence  <b>i Note</b> Needs a date column	value of the last chronological occurrence for the current reference

9. In *Target Column* dropdown list, select which column you want to use for the aggregate.
10. If you want to define a time window for your aggregate, click the *Period Settings* tab.
11. Check *Define Periods*.
12. Choose if you want to define a *Single Period* or several *Successive Periods*.
  - For a single period, select a *Start Date* and an *End Date* for the period over which you want to aggregate the data. You can either use an existing field (*Field*), a constant (*Constant*) or an authored prompt (*Prompt*). If the field you want to use does not exist, you can create it on the fly by clicking the + button located on the right of the date field.

**i Note**

The start date is included in the defined period, whereas the end date is excluded from it.

- For successive periods, define the number of successive periods you want, their length and the starting date, using the hyperlinks (underlined in blue).

**i Note**

The starting date is included in the periods.

13. Click *OK* and enter the name of the new field.
14. Click *OK*.

### 3.3.2.8 Creating a Condition to Filter the Population on a Field

Filtering the population allows you to keep only the records of interest for your business issue. First, you need to filter the population to keep only the active dispositions, meaning that you will use the aggregate you just created to select only those having at least one transaction in the last 6 months. Then you will need to keep only the records for which there is no credit card on January 1, 1998; meaning the records for which the card issued date is either empty or is later than January 1, 1998.



### i Note

To execute this scenario, use the following settings:

Keeping only...	Creates the condition...
active dispositions	TransactionsNumberLast6M > 0
dispositions without credit cards	isNull(issued_date)    issued_date > KxTimeStamp

1. Click [New Condition](#).  
The expression editor appears.
2. Use the offered functions and variables by double-clicking them or enter directly a valid expression. The indicator above the [Messages](#) field indicates whether the expression is valid or not.
3. Click [OK](#).

## 3.3.2.9 Creating the Variable to be Used as the Target

You want to create a model able to predict who is likely to buy a credit card in the following month. The time-stamped population you have created contains only dispositions that have no credit card associated on January 1, 1998, so you need to create a target indicating who has bought a credit card between January 1, 1998 and February 1, 1998.

### i Note


To execute this scenario, use the following settings:

- In The [Expression Editor](#), enter the expression

```
KxTimeStamp<=issued_date && issued_date<dateAddMonth(KxTimeStamp, 1)
```

- [Computer Field Name](#): enter

```
creditcard_bought
```

1. In the [Data Manager](#) panel, select the time-stamped population to which you want to add a target.
2. Click  ([Edit](#)) to open the [Filtered Time-stamped Population](#) panel.
3. Click [Edit Filters](#) to open the [Time-stamped Population Editor](#).
4. In the + menu, click [Expression Editor](#).  
The [Expression Editor](#) appears.
5. You can either enter a valid expression directly in the upper text field, or you can insert functions, variables, field sets, and authored prompts by double-clicking to insert them at the cursor location.
6. Click [OK](#).
7. Enter the name of the variable.

8. Click *OK* and *Next* to define the variable.

### 3.3.2.10 Defining the Target

1. In the *Filtered Time-stamped Population* panel, click the *Target* tab.
2. In the *Target* dropdown list, select the target variable you have just created and click *Next*.

### 3.3.2.11 Copying Objects to Another Metadata Repository

The *Metadata* button allows you to specify the location where the metadata should be stored. The Copy option is useful when you are using different metadata depending on the team you are working with.

1. Select the objects you want to copy.
2. Right-click on the selection and choose *Copy to another metadata repository* in the contextual menu.
3. The *Target Metadata Repository* window appears.
4. Select a *Data Type* and a *Folder*.
5. Click *Next*.
6. Click *OK* to confirm or *Previous* to change your former selection. If you confirm the summary, the objects will be saved in the new Metadata Repository.
7. To check that the procedure has been correctly carried out, select the target folder as the metadata repository in the *Data Manager* panel. If the objects have been successfully saved, they will be displayed in the objects list.

## 3.3.3 Creating the First Classification Model

You will create a first classification model to see if the analytical record you have created contains sufficient information to predict who will be likely to buy a credit card in the following month.

### 3.3.3.1 Selecting the Data to be Modeled

You need to select the data that will be used to create the model.

#### ❖ Example

For this scenario, use the following settings:

- Select the *Data Set Factory Mode*.
- *Database Source*: select the database containing the data.

- **Analytical Record:** select AR\_ClientGeoDemo.
- **Time-stamped Population:** select ActiveDisposition@Date.
- Validate the default date for the prompt.

1. In the *Modeler* section of the start panel, click *Create a Classification/Regression Model*.
2. Check the mode you want to use to select the data to be modeled. There are two ways to select the data:
  - *Use a File or Database Table*, which allows you to select a standard data source such as database table, a text file, or a data manipulation created with Automated Analytics Data Management feature.
    - a. Select the type of the data source
    - b. Use the *Browse* button corresponding to the *Folder* field to select the folder or database containing the data. If you want to access a protected database do not forget to enter the username and password when selecting the database.
    - c. Use the *Browse* button corresponding to the *Data Set* field to select the file or table containing the data.
  - *Use Data Manager*, which allows you to directly select the analytical record and the time-stamped population.
    - a. Use the *Browse* button corresponding to the field *Database Source* to select the database containing the analytical record and time-stamped population. If you want to access a protected database do not forget to enter the username and password when selecting the database.
    - b. Use the *Browse* button corresponding to the field *Analytical Record* to select the analytical record you want to use. In the window *Data Source Selection*, all the existing *Analytical Records* are located under the tree item *Analytical Record*.
    - c. Use the *Browse* button corresponding to the field *Time-stamped Population* to select the time-stamped population you want to use. In the window *Data Source Selection*, all the existing Time-stamped Populations are located under the tree item *Time-stamped Population*.
3. If need be, create the target.
4. Click *Next*. Depending on the size of the query sent to the database, an additional warning may be displayed, asking you to validate its execution. For more information see *Special Case: Data Stored in Databases - the Explain Mode*.
5. If the time stamp for the population is a prompt, it is displayed asking you to either validate the default date, or to enter another date.
6. Click *OK*.

### 3.3.3.1.1 SAP HANA as a Data Source

You can use SAP HANA databases as data sources in Data Manager and for all types of modeling analyses in Modeler: Classification/Regression, Clustering, Time Series, Association Rules, Social, and Recommendation.

SAP HANA tables or SQL views

found in the *Catalog* node of the SAP HANA database

## All types of SAP HANA views

found in the [Content](#) node of the SAP HANA database.

An SAP HANA view is 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 analytic view is built on a fact table and associated attribute views. A calculation view executes a function on columns when the view is accessed.

### ! Restriction

- Analytic and calculation views that use the variable mapping feature (available starting with SAP HANA SPS 09) are not supported.
- You cannot edit data in SAP HANA views using Automated Analytics.

---

## Smart Data Access virtual tables

Thanks to Smart Data Access, you can expose data from remote sources tables as virtual tables and combine them with HANA regular tables. This allows you to access data sources that are not natively supported by the application, or to combine data from multiple heterogeneous sources.

### ⚠ Caution

To use virtual tables as input datasets for training or applying a model or as output datasets for applying a model, you need to check that the following conditions are met:

- The in-database application mode is not used.
- The destination table for storing the predicted values exists in the remote source before applying the model.
- The structure of the remote table, that is the column names and types, must match exactly what is expected with respect to the generation options; if this is not the case an error will occur.

### ⚠ Caution

In Data Manager, use virtual tables with caution as the generated queries can be complex. Smart Data Access may not be able to delegate much of the processing to the underlying source depending on the source capabilities. This can impact performance.

## Prerequisites

You must know the ODBC source name and the connection information for your SAP HANA database. For more information, contact your SAP HANA administrator.

In addition to having the authorizations required for querying the SAP HANA view, you need to be granted the `SELECT` privilege on the `_SYS_BI` schema, which contains metadata on views. Please refer to SAP HANA guides for detailed information on security aspects.

### 3.3.3.2 Managing Performance when Using Databases

Before requesting data stored in a Teradata<sup>(1)</sup>, Oracle<sup>(2)</sup> or SQLServer 2005 database, the application uses a feature, called the **Explain mode**, which categorizes the performances of SQL queries in several classes defined by the user. In order to be as fast and as light as possible, this categorization is done **without actually executing the full SQL query**.

#### i Note

- (1) For all versions of Teradata.
- (2) For all versions above and including Oracle 10.

The objective is to allow estimating the workload of the SQL query before executing it and then deciding -- possibly thanks to an IT Corporate Policy-- if the SQL query can actually be used.

For example, an IT Corporate Policy may favor interactivity and then define 3 classes of SQL queries, each with its maximum time:

- *Immediate*:  $duration < 1 s$ . The query is accepted and executed immediately.
- *Batched*:  $1s \leq duration < 2 s$ . The query is accepted but will be executed on next idle time.
- *Rejected*:  $2s \leq duration$ . The query will never be executed.

The number, names and limits of classes are defined by the user in order for these values to match the current DBMS configuration and DBMS usage policy.

## The Explain Mode has been Configured

If the Explain mode has been configured by your DBMS administrator, there are two possible outcomes to a query:

- **the query is accepted and executed**: this is completely transparent. The application accesses the data without further input from the user.
- **the query needs to be validated before being executed**: a pop-up window opens displaying a message configured by the DBMS administrator. A query that needs validation can be categorized in two ways:

medium-sized  or huge  .

If the query is categorized as medium-sized, you will probably have to check with your administrator which action to take:

- If the administrator authorizes the query, click *Continue*. The pop-up window closes and the requested action is carried out.
- If the administrator does not authorize the query, click *Stop Query*, the pop-up window closes, but no action is executed.

If the query is categorized as huge, it means that the query will take too much time and resources. In that case, the behavior of the *Continue* button depends on the configuration set by the DBMS Administrator (for example, it can automatically refuse queries that are considered too heavy). In any case, you should check with them to know the line of action to follow.

## The Explain Mode has not been Configured

If your DBMS Administrator has not configured the Explain mode, a pop-up window opens when you try to access the data.

You need to contact your Administrator who will tell you which action to take and configure the Explain mode.

If the Administrator validates the execution of the query, you may want all queries with the same duration to be executed without validation. In that case, check the box *Do not request validation anymore for similar requests*. The validation message will then only appear for larger queries. This configuration will only be used for the current session, when closing the application, it will be lost. For a permanent configuration, see your DBMS Administrator.

### 3.3.3.3 About Data Description

In order for Automated Analytics components to interpret and analyze your data, the data must be described.

To put it another way, the description file must specify the nature of each variable, determining their:

- **Storage** format: number (*number*), character string (*string*), date and time (*datetime*) or date (*date*).
- **Type**: *continuous*, *nominal* or *ordinal*.

For more information about data description, see **Types of Variables** and **Storage Formats** in the *Classification, Regression, Segmentation and Clustering Scenarios - Automated Analytics User Guide*.

To describe your data, you can:

- Either use **an existing description file**, that is, taken from your information system or saved from a previous use of Automated Analytics components,
- Or **create a description file using the *Analyze* option**, available to you in SAP Predictive Analytics. In this case, it is important that you validate the description file obtained. You can save this file for later reuse. If you name the description file **KxDoc\_<SourceFileName>**, it will be automatically loaded when clicking *Analyze*.

### ⚠ Caution

The description file obtained using the *Analyze* option results from the analysis of the first 100 lines of the initial data file. In order to avoid all bias, we encourage you to mix up your dataset before performing this analysis.

Each variable is described by the fields detailed in the following table:

The Field...	Gives information on...
<i>Name</i>	the variable name (which cannot be modified)
<i>Storage</i>	the type of values stored in this variable: <ul style="list-style-type: none"><li>• <i>Number</i>: the variable contains only "computable" numbers (be careful a telephone number, or an account number should not be considered numbers)</li><li>• <i>String</i>: the variable contains character strings</li><li>• <i>Datetime</i>: the variable contains date and time stamps</li><li>• <i>Date</i>: the variable contains dates</li></ul>
<i>Value</i>	the value type of the variable: <ul style="list-style-type: none"><li>• <i>Continuous</i>: a numeric variable from which mean, variance, etc. can be computed</li><li>• <i>Nominal</i>: categorical variable which is the only possible value for a string</li><li>• <i>Ordinal</i>: discrete numeric variable where the relative order is important</li></ul>
<i>Key</i>	whether this variable is the key variable or identifier for the record: <ul style="list-style-type: none"><li>• <i>0</i> the variable is not an identifier;</li><li>• <i>1</i> primary identifier;</li><li>• <i>2</i> secondary identifier...</li></ul>
<i>Order</i>	whether this variable represents a natural order.  There must be at least one variable set as Order in the Event data source.  <div data-bbox="820 1487 1396 1684" data-label="Complex-Block"><h3>⚠ Caution</h3><p>If the data source is a file and the variable stated as a natural order is not actually ordered, an error message will be displayed before model checking or model generation.</p></div>
<i>Missing</i>	the string used in the data description file to represent missing values (e.g. "999" or "#Empty" - without the quotes)
<i>Group</i>	the name of the group to which the variable belongs
<i>Description</i>	an additional description label for the variable

Use the following settings:

- Use the *Analyze* feature.

- Set all the identifiers to *nominal*.

### 3.3.3.4 Selecting the Target Variable

By default, and with the exception of key variables (such as KxIndex), all variables contained in your dataset are taken into consideration when generating the model. You may exclude some of these variables.

For the first analysis of your dataset, we recommend that you retain all variables. It is particularly important to retain even the variables that seem to have no impact on the target variable. If indeed these variables have no impact on the target variable, the model will confirm this. Otherwise, the model will allow you to recognize previously unidentified correlations between these variables and the target variable. By excluding variables from the analysis based on simple intuition, you take the risk of depriving yourself of one of the greatest value-added features of Automated Analytics models: the discovery of non-intuitive information.

To create your model, you use data where the target is already known. However to be able to predict correctly your target when you apply the model, you need to exclude any information that allowed you to create the target from the explanatory variables. In this scenario, it means that you need to remove from the explanatory variables all the variables coming from the table in which the credit card information is stored.

#### i Note

To perform this task, use the following setting:

- *Target Variable*: creditcard\_bought

1. In the *Selecting Variables* window, in the *Explanatory Variables Selected* section (left hand side), select the variables you want to use as target variables. By default, the last variable of the dataset is selected as the target variable.

#### i Note

On the *Selecting Variables* screen, variables are presented in the same order as that in which they appear in the table of data. To sort them alphabetically, select the option *Alphabetic sort*, presented beneath each of the variables list.

2. Click the > button located on the left of the screen section *Target(s) Variable(s)* (upper right hand side). The variable moves to the screen section Target(s) Variable(s).
3. Select a variable in the screen section *Target(s) Variable(s)* and click the < button to move the variables back to the screen section *Explanatory Variables Selected*.

### 3.3.3.5 Selecting a Weight Variable

1. On the *Selecting Variables* screen, in the *Explanatory Variables Selected* section (left hand side), select the variables you want to use as a weight variable.



### i Note

On the *Selecting Variables* screen, variables are presented in the same order as that in which they appear in the table of data. To sort them alphabetically, select the option *Alphabetic sort*, presented beneath each of the variables list.

2. Click the > button located on the left of the *Weight Variable* screen section (middle right hand side). The variable moves to the *Weight Variable* screen section.
3. Select a variable in the *Weight Variable* screen section and click the < button to move the variables back to the *Explanatory Variables Selected* screen section.

## 3.3.3.6 Excluding Variables

Exclude variables whenever they are not relevant to the model you are trying to set up.

### i Note

To perform this task, use the following settings:

- *Excluded Variables*: KxIndex\_1, card\_id, card\_type, issued\_date

1. On the *Selecting Variables* screen, in the *Explanatory variables selected* section (left hand side), select the variables to be excluded.

### i Note

On the *Selecting Variables* screen, variables are presented in the same order as that in which they appear in the table of data. To sort them alphabetically, select the option *Alphabetic sort*, presented beneath each of the variables list.

2. Click the > button located on the left of the *Excluded Variables* screen section (lower right hand side). The variable moves to the *Variables excluded* screen section.
3. Select a variable in the *Variables excluded* screen section and click the button < to move the variables back to the *Explanatory variables* screen section selected.
4. Click *Next*.  
The *Parameters of the Model* panel appears.

## 3.3.3.7 Model Parameters

The *Summary of Modeling Parameters* panel allows you to check the modeling parameters before generating the model.

The name of the model is automatically filled. It corresponds to the name of the target variable, followed by the underscore sign ("\_") and the name of the data source, minus its file extension.

### i Note

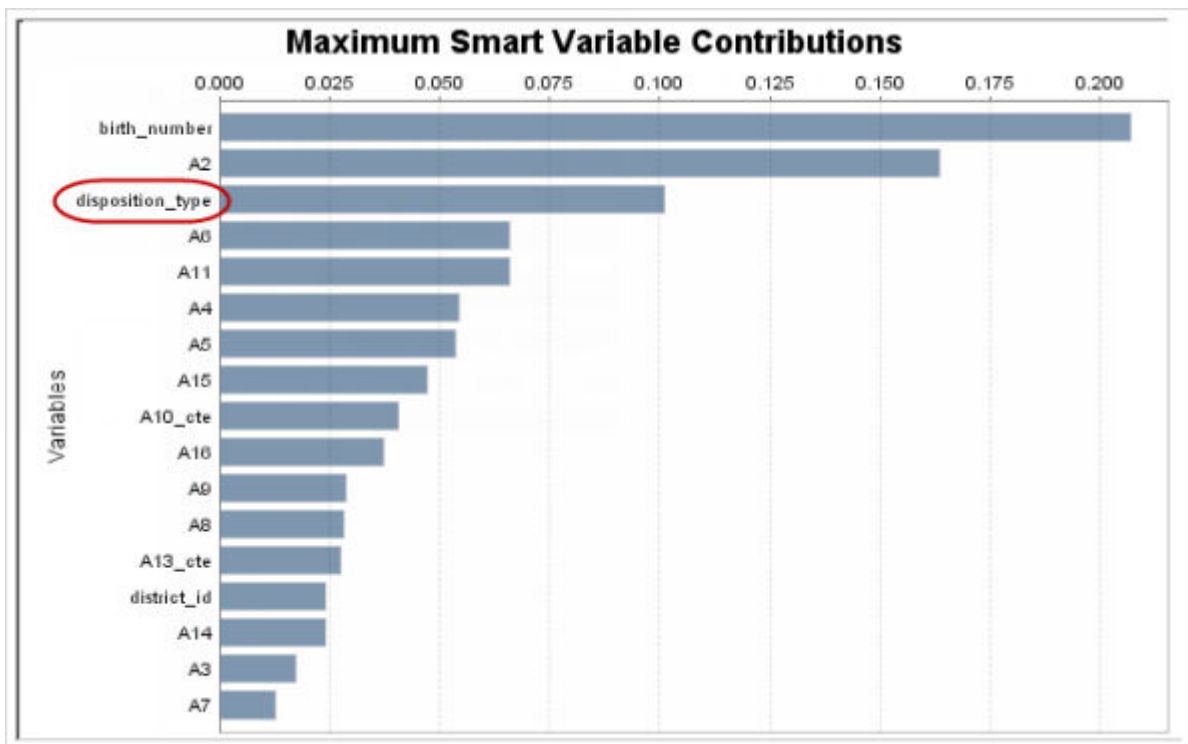
The *Summary of Modeling Parameters* panel contains an *Advanced* button. By clicking this button, you access the *Specific Parameters of the Model* panel, where you can set the polynomial order of the model to

be generated. For more information, see the Classification/Regression section of the *Classification, Regression, Segmentation and Clustering Scenarios - Automated Analytics User Guide*.

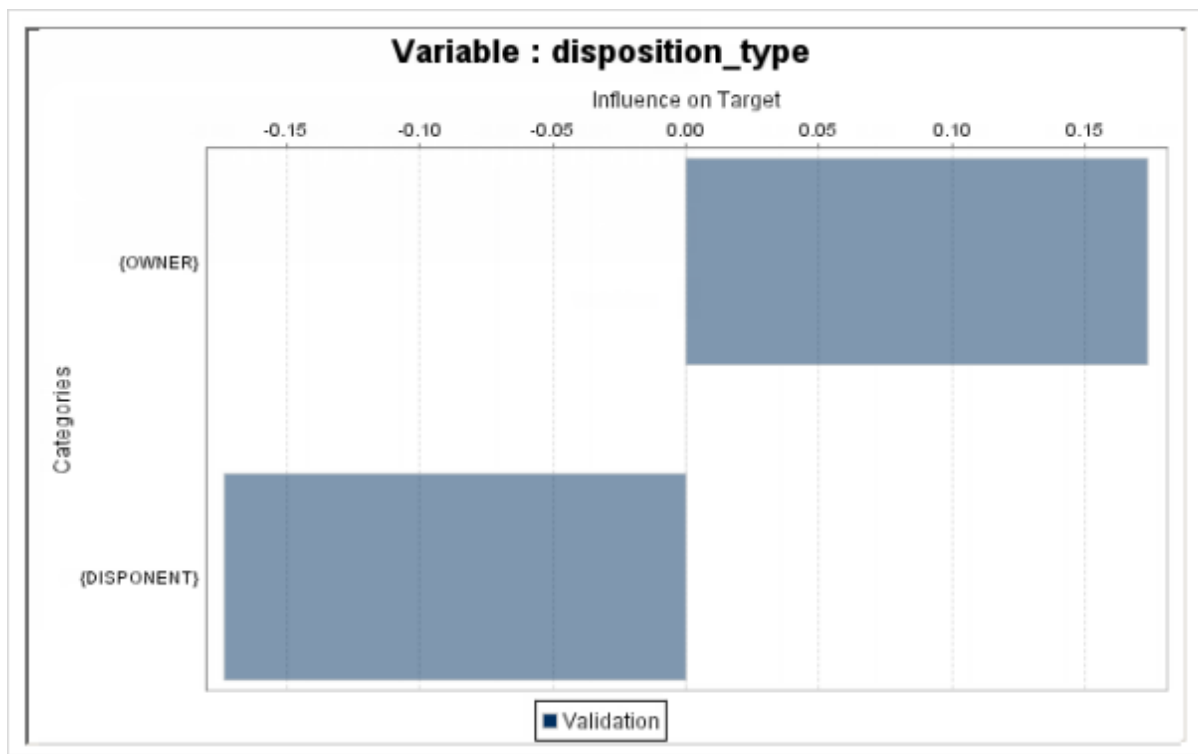
### 3.3.4 Model Results Analysis

If you look at your model results, you can see that the performance indicators are not very high. The quality of your model, represented by the KI, can be improved by adding information to the analytical record. The analytical record you have created contained only geographic and demographic data, however you have a lot of information available on the customers behavior, such as the amount withdrawn from an account or the amount checked in the account, the balance at the end of each month, and the tenure of the account. You could also add the product history, with the number of loans taken in the last year for example, and the loan history.

However, this first model already reveals some information. If you look at the Contributions by Variables (see graphic below), you can see that one of the most important variables is the disposition type.



If you click this variable on the graph, you will display the categories significance, that will show that a customer who is owner of an account is more likely to buy a credit card, than one who is only a user of the account.



## 3.4 Entity Transfer

It is possible to migrate an entity and its time-stamped population while performing a data transfer. This option is available via the **Toolkit** feature.

The data transfer process encompasses various steps starting from the creation of the entity and its related time-stamped population in **Data Manager** to performing their transfer in the **Toolkit**.

### 3.4.1 Transferring an Entity

1. In the start panel, select the feature *Perform a Data Transfer* in the Toolkit section. The *Select Dataset Source* panel appears.
2. Select a *Data Type*.
3. Select a *Folder*.
4. Select a *Dataset*, which is the entity you want to transfer into a new database (it is preceded by the icon with a green arrow in the middle).
5. Click **OK > Next**.
6. In the *Transfer a Data Manipulation* pop-up window, click *Transfer Definition*. The *Data Manipulation Transfer* panel appears.

7. In *Target Settings*:
  - select the *Target Database*, that is, the database where you want to transfer your data manipulation,
  - in *Transfer as*, enter the name the data manipulation should have in the target data base.
8. Click *Guess* to define the data manipulation source in the Table Mapping area.

If the new database has a similar table with a similar name, the application automatically shows the correct *Target Table*.

#### Caution



The new database must have the similar source table(s) as the existing database, that is, if the Data Manipulation is derived from sources TableA and TableB, both TableA and TableB must be migrated to the new database before performing the Data Manipulation transfer.

9. Click *Edit Field Mapping* to edit the correct field mapping.

The panel Edit Field Mapping appears. By default, there is no mapping defined.
10. If you have selected a correct Target Table in the previous step, click *Guess*.

The *Target Fields* are automatically filled in.
11. Click *OK*.

The green arrow on the icon *Edit Field Mapping* indicates that the mapping is done.
12. Click *Check*.

A pop-window indicates that the mapping for the transfer is correct.
13. Click  *OK > Next* .
14. Click *OK*.

The transferred entity is now ready to be re-used in a new database.

## 3.4.2 Using the Transferred Entity in a New Database

1. In the **Start Page**, click *Create or Edit Data Manager Objects* under *Data Manager*.

The *Connect to Data Manager* panel appears.
2. Select:
  - *Data Base* in *Date Type*.
  - In *Database source*, the database where the data manipulation has been transferred.
3. Click *Next*.

The transferred entity is automatically displayed.
4. Make changes and/or click *Next*.

The *Data Manager* panel appears, listing the newly created entity and the default time-stamped population associated with it.

# 4 Event Log Aggregation Scenario

## 4.1 About Event Log Aggregation Scenario

The primary goal of this section is to show the added value that event logging – an Automated Analytics data manipulation feature – can bring to your data mining activity. At the same time, it will serve as a tutorial for people who want to evaluate and get started with:

- Event Logging,
- Modeler - Regression/Classification.

This section first gives a description of the application. Then, a use scenario takes you through the steps of creating data models first using the regression/classification model , then using event logging combined with the regression/classification model . All the data modeling tasks are performed using SAP Predictive Analytics.

At the end of the scenario, you will understand how event logging can help you make the most out of your transaction data.

To perform the scenario presented in this section, we recommend that you have a basic knowledge of predictive analytics concepts. To give you a few examples, you must understand what the following concepts stand for: "target variable", "predictive power", "prediction confidence", "profit curves", and so on.

## 4.2 Event logging: Description

### What

Event logging is a data manipulation feature that builds a “mineable” representation of an event history. It merges reference information from a table with information from history tables, which is aggregated automatically per period of time.

### Why

The information necessary to build predictive models is often spread across a table containing “static” information such as customer demographics or equipment specifications and a log of transactions such as purchase history, service call history or equipment alarms. To build predictive models, this data must be compressed and combined into a single row per analytic record, representing both the static reference information and the event history.

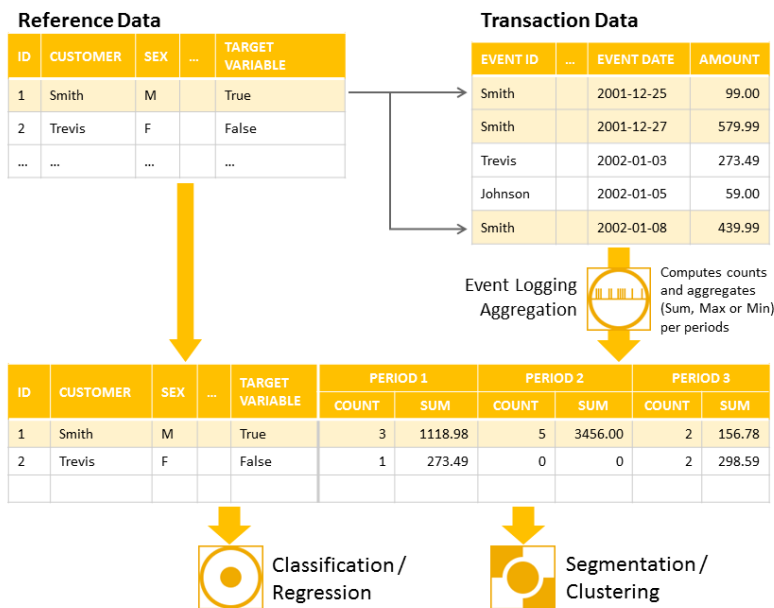
## How

Event logging creates aggregates on user defined periods. Period length can be day, week, month, and so on. They are computed from a reference date that can be fixed or specific to each of the reference cases (for example, date of first purchase for a customer). When a specific reference date is selected, the application creates what is called Relative Aggregation, that is an aggregation not based on calendar or even fiscal periods, both of which are static for all data. The details and timing of event data is extremely important in understanding the customer's behavior. The following figure shows how selecting a specific reference date can reveal a pattern in the customers' behavior.



The application is programmable and you can specify the aggregates (min, max, sum, count, and so on).

The figure below illustrates how the application works.



## Benefits for the business user

Event Logging does not require programming to perform this sophisticated aggregation. Due to the speed of the application, several aggregation options can be tested ad-hoc to find the most meaningful solution.

## Benefits for the Data Mining expert

Event Logging enables the Data Mining professional to include additional historical data in the analysis process, resulting in better models. The application is fast and can handle very large datasets.

## Benefits for the Integration specialist and IT

Only one pass of the log table is required, using an efficient internal data representation. Building transactional aggregates can be done in minutes instead of days, and can be used to prototype permanent ETL processes. No changes to the underlying schema are required.

## Examples

For Customer Relationship Management (CRM), the most valuable information is how a customer has interacted with a company and its products. This information is typically stored as a purchase history, or call

center log. When performing an analysis to predict customer churn, a customer's actions with respect to the time they left can be critical for maximizing model quality. This requires an event aggregation based on the churn date. Customers churn at different times, so aggregating on a fixed date, such as January 2001, is not necessarily meaningful for the analysis. In this case, the count of purchases and complaint calls, and the sum of purchases could be automatically aggregated for each month in the year before the churn date. Once this is done by event logging, a regression/classification model could be used to predict churn.

In a different scenario, when predicting machine part failure, the static information about a particular piece of equipment (lot number, manufacture date, and so on) is not nearly as important as how the equipment has been used. The operating logs, with conditions such as temperature and pressure can be utilized by the application. A series of alarms in a new machine can be very different than the same set of alarms in a ten-year-old machine. Alarm counts along with maximum pressure and temperature for each quarter over the first five years of service life could be automatically created by the application. In this case, a segmentation/clustering model might be used in addition to create segments of equipment with high risk and low risk for failure.

## 4.3 Use Scenario: Overview

### A Scenario in Three Main Steps

The use scenario will take you through the three following steps:

1. Importing the data samples provided to you as CSV files into your DBMS, in order to create a database whose tables can be indexed and joined.
2. Creating an ODBC connection to the database created in step 1, so that Automated Analytics can access it.
3. Creating three different predictive models, by taking into account different data sources and by using only a regression/classification model for the first two models, and then event logging combined with a regression/classification model for the third model. You will then be able to compare the results obtained with each model according to the data sources used.

### Technical Requirements

#### Use Scenario: Configuring Tasks

The first two tasks of this scenario – Importing the Sample Data into your DBMS and Creating an ODBC connection – are configuring tasks. The details related to performing these tasks depend on your technical environment.

For more information, see the two documents *Import Flat Files into your DBMS - Support Document* and *Connect to your DBMS - Support Document*, or please ask your system administrator.

#### Use Scenario: SQL Queries

In this scenario, you will have to use SQL queries. Versions of these queries have been tested and will work on the following DBMS:

- Microsoft Access,



- IBM DB2,
- Oracle 8i.

## Introduction to Sample Data Files

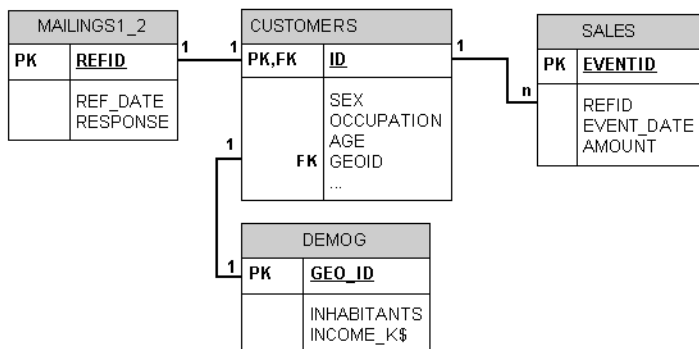
You can download the sample files from the SAP Help Portal page at <http://help.sap.com/pa>.

The File...	Contains...
Customers.csv	<p><i>reference data</i> about your company's customers.</p> <p>It lists 360,000 customers, who are described by 13 variables, such as their gender, their age or their occupation, and so on.</p>
Demog.csv	<p><i>demographic data</i>.</p> <p>It lists 200 geographic areas that are described by their number of inhabitants, the average income per inhabitant and other continuous variables.</p>
Sales.csv	<p><i>transaction data</i> of the customers referenced in the Customers.csv file.</p> <p>It lists about 3,400,000 purchases, that are described by an identifier, an amount and a date. About a dozen transactions or so are referenced for each customer.</p>
Mailings1_2.csv	<p>This file contains the <i>responses</i> of the first 60,000 customers of the Customers.csv file to the incentive mailing that your marketing department sent. It also contains the date the mailing was sent on.</p>
Mailing3.csv	<p>This file contains the <i>actual responses</i> to the marketing incentive mailing of the 300,000 customers you will need to target. You will not use this file to perform the scenario. You could use it at the end of the scenario in order to check the efficiency of the customer targeting that you will have done thanks to your data mining model.</p>

### ⚠ Caution

We strongly recommend that you do not change the names that we specify for data tables and other files. Otherwise, you will have to adapt SQL queries and other settings to your particular situation.

The diagram below shows the data tables used for the scenario and the relationships between these tables.



### i Note

On the diagram above, PK stands for "Primary Key", FK stands for "Foreign Key" and I stands for "Index".

## Additional Sample Files

If you want to quickly test Event logging before starting with the scenario provided in this user guide, another set of sample files is available in the folder `Samples\KedData` with a readme text file describing how to use these samples.

The folder `Samples\KedData` is located:

- for Windows, in the installation sub-folder `\Automated\`.
- for Linux, in the folder where you have decompressed the KXENAF archive file (that is `.tar.Z` or `.tar.gz`).

### Note about Date and Datetime Variables

Internally all dates are converted to datetime. This allows comparing and mixing dates with different formats, either date or datetime.

Duration computations also follow this behavior. When performing Event Log Aggregation or Sequence Analysis, the periods defined in the settings (such as "3 periods of 2 weeks before the reference date") are converted as bounds of datetime ranges.

When a date is converted to datetime, the time is set by default to noon (12:00), instead of midnight (0:00). This is to avoid problems when converting back to date from datetime (as a one second delta may change the date from one day). For example, if you look at a table containing date values which description is forced to datetime, you will see the dates with a time set to 12:00:00.

### → Tip

In the user interface, to indicate a datetime compatible with a date value, enter it with the time set to noon (12:00:00).

## 4.4 Use Scenario: Introduction

### Your Objective

You are a member of the Sales Management team in a large retail bank.

The current date is July 02, 2007. Your Sales Director has just asked you to generate additional revenues of \$1,500,000 before September 01, 2007. You must find ways to sell more "Credit++" – the new product of consumer credit that has been developed within your company for the last six months.

### Your Means and Constraints

A few months back, your marketing department has sent incentive mailings to people referenced in its customer database, offering them to apply for the new consumer credit product.

The two waves of mailings were sent one month from one another, each one to a sample of 30,000 people randomly selected among the overall population referenced in the customer database.

According to the CRM processes in force in your company, any person who has not responded to a mailing within two months is considered to have responded negatively. Taking such an hypothesis into account, the average response rate obtained for the first two mailings amounts to 4.99%. In other words, 2,994 out of the 60,000 people contacted have eventually applied for the new credit product. Only the first two waves – that is, 60,000 people – can indeed be taken into account with respect to the time required to identify non-responders. The average revenue generated from a person that responds positively to a mailing is \$900.

For all the 360,000 people referenced in your customer database, you have at your disposal:

- Reference data,
- Transaction data
- Demographic data, that your marketing department purchased. This data provides information such as the standard of living or the number of inhabitants for every demographic area.

You also have at your disposal Response Data, that is a table containing a list of the 60,000 people who were sent the marketing incentive mailing, 2,994 of which responded positively with regard to the new consumer credit product. For each of these 2,994 responders, you also know the amount of the credit they were granted.

### Your Business Issue

Based on the average revenue per person – that is, \$900 – 1667 responders need to be contacted to generate an income of \$1,500,000. Based on the average response rate – that is, 4.99% – a mailing should be sent to 33,400 people.

Your problem is that you will not be able to find the time to contact over 33,000 people due to the operational constraints you face within your company. In order to maximize your response rate, you need to develop a targeted marketing campaign to predict which customers are the most likely to respond to the incentive mailing among the 300,000 referenced in your database.

## Your Solutions

Your solution here is Data Mining: You will generate a predictive model supervised by the known responses of the 60,000 customers already contacted. You will then apply this model on the other 300,000 customers referenced in your database. As a result of the model application, you will obtain a file containing the customers who are the most likely to respond positively to your incentive mailing.

So, your point here is not to select the best Data Mining tool available on the market, but to select the best method to build the predictive model among the three following ones:

- A simple method, that consists of using only the customers' reference data.
- A intermediate method, that consists of using the customers' reference data combined with the purchased demographic data.
- An overall method, which consists of using the customers' reference data, combined with the purchased demographic data and the customers' transaction data.

For the first two methods, you will use the regression/classification engine. The third method is where event logging comes into play, enabling you to pre-process the transaction data so that it can be processed by the regression/classification engine.

## 4.5 Step 1 - Configuring the Data Source

To enable the joining and indexing of data tables, you need to import the data samples into your database management system (DBMS). Data Toolkit provides you with a Data Transfer feature that will allow you to easily import the provide csv files into your database.

### Data Sources Supported

The following data sources are supported:

- Flat files (text files) in which the data are separated by a delimiter, such as commas in .csv (Comma Separated Value) format file. For instance, the sample file Census01.csv, used for the regression/classification and segmentation/clustering application scenarios, is a .csv file.

#### ! Restriction

When accessing data in .csv files, Automated Analytics only supports `CR` + `LF` (common on Microsoft Windows) or `LF` (common on Linux) for line breaks.

- ODBC-compatible data sources.

#### ⚠ Caution

SAP HANA information views are not supported by Data Manager, only standard tables or views can be used as data source.

### i Note

For the list of supported platforms, refer to the [Product Availability Matrix](#).

## Creating an ODBC Connection

Before using the Data Transfer feature to import the sample files in your database, you need to create an ODBC connection so that Automated Analytics can access it.

To know how to create an ODBC connection, see the document [Connecting to your Database Management System on Windows or Linux](#), or ask your system administrator.

## Importing CSV Files into your DBMS Using Data Transfer

To import the csv files provided as sample files into a database, you can use the Data Transfer feature. The table below lists the files to import in the database and the settings to apply for the import.

When Importing the file..	Use the description...	Specify the following field as primary key...	Index the following field...
Customers.csv	Customers_desc.txt	ID	GeoID
Demog.csv	Demog_desc.txt	GEO_ID	-
Sales.csv		EVENTID	REFID
Mailings1_2.csv	mailings_desc.txt	REFID	-
Mailings3.csv	mailings_desc.txt	REFID	-

### i Note

An index speeds up queries on the indexed fields as well as sorting and grouping operations. A primary key field holds data that uniquely identifies each record in a table.

## 4.5.1 Importing CSV Files into a Database

1. On the start menu, in the *Toolkit* section, click the option *Perform a Data Transfer*.  
The panel *Select Dataset Source* is displayed.
2. In the list *Data Type*, select the option *Text Files*.
3. Use the *Browse* button located next to the *Folder field* to select the folder where you have saved the event logging sample files.
4. Use the *Browse* button located next to the field *Dataset* to select the file you want to import.

5. Click the *Next* button. The panel *Describe Dataset* is displayed.
6. If a description file exists for the file you have selected as a source:
  1. Click the button *Open Description*.
  2. Select the description file in the window *Load a Description for ...*
  3. Click the *OK* button.
  4. Else click the *Analyze* button.

The file description is displayed.

#### **i** Note

The application automatically indexes all fields that have their key field set to 1 in the description.

7. Click the *Next* button. The panel *Create Dataset Copy* is displayed.
8. In the list *Data Type*, select the option *Data Base*.
9. Use the *Browse* button located next to the Folder field to select the data base in which you want to import the data.

The pop-up window Data Selection opens.

#### **i** Note

If it has not been done yet, contact your administrator to set up the database connection on your computer.

10. If the selected database is password-protected, enter the login information in the *User* and *Password* fields.
11. Click the *OK* button.
12. In the field *Output Dataset*, enter the name of the table to create. It should have the same name as the file from which the data have been imported.
13. Click the *Next* button. The panel *Data Transfer* is displayed. A progress bar indicates the advancement of the transfer.
14. Once the transfer is over, click the *Next* button to go back to the starting menu.
15. Repeat the whole procedure for each file to import.

## **4.6 Step 2 - Modeling your Data**

### **4.6.1 Simple Method: Using Only Reference Data**

#### **4.6.1.1 Simple Method: Description**

To tackle your issue, you decide to first try and see if the customers' reference data you own contain enough information to allow for an effective targeting of your sales campaign.

Using the regression/classification engine, you will generate a predictive model in order to determine the way prospects will respond to incentive mailings.

This model will be generated by using the reference data of the 60,000 customers who received the first two waves of mailings sent by your marketing department.

It means that you will need to:

- Join the table containing the first 60,000 customers' responses (`Mailings1_2.csv`) with the table containing their customers' reference data (`Customers.csv`).

## 4.6.1.2 Simple Method: Modeling Process

The regression/classification engine allows you to create explanatory and predictive models.

The first step in the modeling process consists of defining the modeling parameters:

1. Select a partition strategy.
2. Select a data source to be used as training dataset.
3. Describe the dataset selected.
4. Select the target variable, and possibly a weight variable.
5. Select the explanatory variables.

### Summary of the Modeling Settings to Use

The table below summarizes the modeling settings that you must use for the simple method. It should be sufficient for users who are already familiar with SAP Predictive Analytics.

For detailed procedures and more information, see the following sections.

Task(s)	Screen	Settings
Creating an analytical dataset merging two tables	<a href="#">▶ Data Manipulation ▶ Define a New Analytical Dataset ▶ Merge ▶</a>	<ul style="list-style-type: none"> <li>• source table: mailings1_2</li> </ul> <p>Merge:</p> <ul style="list-style-type: none"> <li>• source field: REFID / target table: customers / target field: ID</li> </ul> <p>Update Fields:</p> <ul style="list-style-type: none"> <li>• Set GEOID and REFID as integer and nominal .</li> <li>• Set REF-DATE as datetime and continuous .</li> <li>• Check that REFID is a key variable.</li> </ul> <p>Save as:</p> <ul style="list-style-type: none"> <li>• KEL_ADS_SimpleMethod</li> </ul>

- Specifying the data source *Data to be Modeled*
  - Select the option ODBC Source .
  - In the ODBC Source field, specify the data source to be used
  - In the Training field, select the analytical dataset KEL\_ADS\_SimpleMethod , created in the previous step.

Describing the data	<i>Data Description</i>	Create the data description using the Analyze button.
Selecting the Target Variable and a weight variable	<i>Selecting the Target Variable</i>	Select the variable RESPONSE as your target variable.
Selecting explanatory variables	<i>Selecting Variables</i>	Exclude the variables REFID and REF_DATE from the list of variables to be used for modeling.

## 4.6.1.2.1 Creating the Analytical Dataset

1. In the Start menu, double-click the option *Define a New Analytical Dataset*.
2. Select the table *mailings1\_2*.
3. Click *Next*. The panel *Edit Temporal Analytical Dataset* is displayed.
4. Click the *Merge* tab.
5. Select *REFID* as the Source Field.
6. Select *customers* as the Target Table.
7. Select *ID* as the Target Field.
8. Click the button *New Merge*. The newly created merge is displayed in the upper part of the panel.
9. Click the *Fields* tab.
10. Click the type corresponding to *REFID* and set it to nominal.
11. Repeat step 10 for *GEOID*.
12. Click the type corresponding to *REF\_DATE* and set it to continuous.
13. Check that *REFID* is identified as a key.
14. Click the *Next* button. The panel *Save and Export* is displayed.
15. In the field *Analytic Dataset Name*, enter *KEL\_ADS\_SimpleMethod* as the name of the new analytical dataset.
16. Click the *Save* button.
17. Click the *Cancel* button to go back to the Start menu.

## 4.6.1.2.2 Selecting a Data Source

To generate the model, you first need to combine your customers' reference data (*Customers table*) with the responses data (*Mailings1\_2 table*) validated by your CRM processes, that is, the responses of the customers who were contacted in the first two waves of mailings.



You can join these tables:

- Either in your DBMS.
- Or directly in SAP Predictive Analytics by creating an analytical dataset in the Data Manipulation feature.

For this Scenario, join the table containing the 60,000 customer responses (`Mailings1_2.csv`) with the table containing their customers' reference data (`Customers.csv`).

To Select a Data Source:

1. On the screen *Select a Data Source* in the list *Data Type*, select the *ODBC* option.

#### **i** Note

By default, the *Text files* option is selected.

2. Click the *Browse* button.

Depending on the option you selected at step 1, the dialog box *Data Source Selection* will appear.

3. Select the folder or database where your data is stored. If the database is password protected, enter the login information in the fields *User* and *Password*.
4. Click the *Browse* button associated with *Training field*.

A selection window will appear.

5. Select the data file to be used.

The name of the file will appear in the Training field.

6. If you have selected the *Customized* partition strategy, repeat steps 4 and 5 for the *Validation* and *Testing* fields.
7. To select only part of the dataset, use the *Advanced Settings*.

8. Click the *Next* button.

The screen Data Description will appear.

9. Go to the section *Describing the Data Selected*.

## 4.6.1.2.3 Describing the Data

### How to Describe Selected Variables

To describe your data, you can:

- Either use an existing description file, that is, taken from your information system or saved from a previous use of SAP Predictive Analytics,
- Or create a description file using the *Analyze* option, available to you in the *Modeling Assistant*. In this case, it is important that you validate the description file obtained. You can save this file for later re-use. If you name the description file `KxDoc_<SourceFileName>`, it will be automatically loaded when clicking the Analyze button.

## ⚠ Caution

The description file obtained using the Analyze option results from the analysis of the first 100 lines of the initial data file. In order to avoid all bias, we encourage you to mix up your dataset before performing this analysis.

Each variable is described by the fields detailed in the following table:

The Field...	Gives information on...
<i>Name</i>	the variable name (which cannot be modified)
<i>Storage</i>	the type of values stored in this variable: <ul style="list-style-type: none"><li>• <i>Number</i>: the variable contains only "computable" numbers (be careful a telephone number, or an account number should not be considered numbers)</li><li>• <i>String</i>: the variable contains character strings</li><li>• <i>Datetime</i>: the variable contains date and time stamps</li><li>• <i>Date</i>: the variable contains dates</li></ul>
<i>Value</i>	the value type of the variable: <ul style="list-style-type: none"><li>• <i>Continuous</i>: a numeric variable from which mean, variance, etc. can be computed</li><li>• <i>Nominal</i>: categorical variable which is the only possible value for a string</li><li>• <i>Ordinal</i>: discrete numeric variable where the relative order is important</li></ul>
<i>Key</i>	whether this variable is the key variable or identifier for the record: <ul style="list-style-type: none"><li>• <i>0</i> the variable is not an identifier;</li><li>• <i>1</i> primary identifier;</li><li>• <i>2</i> secondary identifier...</li></ul>
<i>Order</i>	whether this variable represents a natural order.  There must be at least one variable set as Order in the Event data source.
<i>Missing</i>	the string used in the data description file to represent missing values (e.g. "999" or "#Empty" - without the quotes)
<i>Group</i>	the name of the group to which the variable belongs
<i>Description</i>	an additional description label for the variable

## ⚠ Caution

If the data source is a file and the variable stated as a natural order is not actually ordered, an error message will be displayed before model checking or model generation.

To Create a Description File:

For this Scenario, create the data description by clicking the *Analyze* button.

1. On the screen *Data Description*, click the *Analyze* button.  
The data description will appear.
2. Check that the description obtained is correct.  
If your initial data file contains variables that serve as keys, they are not automatically recognized. Describe them manually, as described in the procedure To Specify that a Variable is a Key.
3. Once the data description has been validated, you can:
  - o Save it by clicking the *Save* button.
  - o Click the *Next* button to go to the following step.  
The screen Selecting the Target Variable will appear.
4. Go to the section *Selecting a Target Variable*.

## To Specify that a Variable is a Key

1. In the *Key* column, click the box corresponding to the row of the key variable.
2. Type in the value "1" to define this as a key variable.

**Guessed Description**

Index	Name	Storage	Value	Key /	Order	Missing	Group	Description	Structure
1	age	integer	continuous	0	0				
2	workclass	string	nominal	0	0				
3	fnlwgt	integer	continuous	0	0				
4	education	string	nominal	0	0				
5	education-...	integer	ordinal	0	0				
6	marital-status	string	nominal	0	0				
7	occupation	string	nominal	0	0				
8	relationship	string	nominal	0	0				
9	race	string	nominal	0	0				
10	sex	string	nominal	0	0				
11	capital-gain	integer	continuous	0	0				
12	capital-loss	integer	continuous	0	0				
13	hours-per-...	integer	continuous	0	0				
14	native-cou...	string	nominal	0	0				
15	class	integer	nominal	0	0				
16	KxIndex	integer	continuous	1	0			Automatical...	

Add Filter in Data Set

Analyze Open Description Save Description View Data

Cancel Previous Next

## Selecting the Target Variable and a Weight Variable

For this Scenario:

1. Select the variable RESPONSE as your target variable.
2. Do not select any weight variable.

To Select Target Variable:

1. On the screen *Selecting Variables*, in the section *Explanatory Variables Selected* (left hand side), select the variable you want to use as Target Variable.

### i Note

On the screen *Selecting Variables*, variables are presented in the same order as that in which they appear in the table of data. To sort them alphabetically, select the option *Alphabetic sort*, presented beneath each of the variables list.

2. Click the button > located on the left of the screen section *Target(s) Variable(s)* (upper right hand side). The variable moves to the screen section *Target(s) Variable(s)*. Also, select a variable in the screen section *Target(s) Variable(s)* and click the button < to move the variables back to the screen section *Explanatory variables selected*.

## Selecting Explanatory Variables

By default, and with the exception of key variables (such as KxIndex), all variables contained in your dataset are taken into consideration for generation of the model. You may exclude some of these variables.

### Event Logging Select Explanatory Var > For this Scenario

For this Scenario

- Exclude the variables *REFID* and *REF\_DATE* from the list of variables to be used for modeling.

### i Note

These two variables, representing the customer unique identifier (*REFID*) and the date on which the different mailings were sent (*REF\_DATE*), are excluded since their values are sure to be completely different from the data found in the apply dataset. This can be tested by using the deviations analysis in the menu *Using the Model*.

- Retain all the other variables.

To Select Variables for Data Analysis

1. On the screen *Selecting Variables*, in the section *Explanatory variables selected* (left hand side), select the variable to be excluded.

### i Note

On the screen *Selecting Variables*, variables are presented in the same order as that in which they appear in the table of data. To sort them alphabetically, select the option *Alphabetic sort*, presented beneath each of the two parts of the screen.

2. Click the button > located in the center of the screen. The variable moves to the screen section *Variables excluded*. Also, click the button < to move the variables to the screen section *Explanatory variables selected*.
3. Click the *Next* button. The screen *Summary of Modeling Parameters* will appear.

### 4.6.1.3 Simple Method: Results

#### Model Performance Indicators

Once the model has been generated, you must verify its validity by examining the performance indicators:

- The predictive power is a quality indicator that allows you to evaluate the explanatory power of the model, that is, its capacity to explain the target variable when applied to the training dataset. A perfect model would possess a predictive power equal to 1 and a completely random model would possess a predictive power equal to 0.
- The prediction confidence is a robustness indicator that defines the degree of robustness of the model, that is, its capacity to achieve the same explanatory power when applied to a new dataset. In other words, the degree of robustness corresponds to the predictive power of the model applied to an application dataset.

To see how the predictive power and the prediction confidence are calculated, see *Predictive Power, Prediction Confidence and Profit Curves* in the document *Classification, Regression, Segmentation and Clustering Scenarios - Automated Analytics User Guide*.

#### i Note

Validation of the model is a critically important phase in the overall process of data mining. Always be sure to assign significant importance to the values obtained for the predictive power and prediction confidence of a model.

The model generated on the customers' reference data gives the following results:

- *Predictive power = 0.198*
- *Prediction confidence = 0.976*

#### Presentation of the User Menu

Once the model has been generated, click the *Next* button. The screen *Using the Model* will appear.

The screen *Using the Model* presents the various options for using a model, that allow you to:

- Display the information relating to the model just generated or opened (*Display* section), referring to the model curve plots, contributions by variables, the various variables themselves, HTML statistical reports, table debriefing, as well as the model parameters.

- Apply the model just generated or opened to new data, to run simulations, and to refine the model by performing automatic selection of the explanatory variables to be taken into consideration (*Run* section).
- Save the model, or generate the source code (*Save/Export* section).

## Taking a Closer Look at the Model

From the screen *Using the Model*, you can display a suite of plotting tools that allow you to analyze and understand the model generated in details. Three useful tools are described in the table below.

On the screen...	You can observe and analyze...
<i>Profit Curves</i>	The performance of the model with respect to a hypothetical perfect model and a random type of model
<i>Contributions by Variables</i>	The contribution of each of the explanatory variables with respect to the target variable
<i>Significance of Categories</i>	The significance of the various categories of each variable with respect to the target variable

On the screen *Contributions by Variables* (see below), you notice that the GEOID variable is one of the variables that contribute the most to the explanation of the target variable. This result lead you to believe that taking the demographic data might significantly improve the predictive power, thus obtaining a better model. This leads you to the intermediate method.

## 4.6.2 Intermediate Method: Adding Demographic Data

### 4.6.2.1 Intermediate Method: Description

The results of the simple method shed light on the importance of the *GEOID* variable. So, after using only the customers' reference data, you decide to now combine them with the externally purchased demographic data to build your predictive model.

Your goal is to check whether this data may contribute to a better targeting of your sales campaign, compared to the simple method.

As the table of demographic data (*Demog*) you own contains no dynamic data (or events), you can still use the sole regression/classification engine to generate the model.

As with the simple method, you will need to join the tables to be used - that is, the *Customers*, *Mailings1\_2* and *Demog* tables.

### 4.6.2.2 Intermediate Method: Modeling Process

The process of building a predictive model on the customers' reference data combined with the demographic data is approximately the same as the one you used for building the model on the reference data.

The only additional step you have to perform is to join the customers' reference data table (*Customers*) with the response data table (*Mailings1\_2*) with the demographic data table (*Demog*). To join these tables, you will use the *Data Manipulation feature*.

## Summary of the Modeling Settings to Use

The table below summarizes the modeling settings that you must use for the intermediate method. Except for the additional SQL query to be used on the screen *Data to be Modeled*, the other settings are similar to the ones used for the simple method.

For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

Task(s)	Screen	Settings
Creating the analytical dataset	► <a href="#">Data Manipulation</a> ► <a href="#">Define a New Analytical Dataset</a> ► <a href="#">Merge</a> ►	<ul style="list-style-type: none"> <li>source table: KEL_ADS_SimpleMethod</li> </ul> Merges <ul style="list-style-type: none"> <li>source field: <i>GEOID</i> / Target table: <i>Demog</i> / Target field: <i>GEO_ID</i></li> </ul> Save as: <ul style="list-style-type: none"> <li>KEL_ADS_IntermediateMethod</li> </ul>
<ul style="list-style-type: none"> <li>Selecting a partition strategy</li> <li>Specifying the data source</li> </ul>	<a href="#">Data to be Modeled</a>	<ul style="list-style-type: none"> <li>Partition strategy: <i>Random</i></li> <li>Select the option <i>ODBC Source</i> .</li> <li>In the <i>ODBC Source</i> field, specify the data source to be used</li> <li>In the <i>Training</i> field, select the analytical dataset KEL_ADS_IntermediateMethod, created in the previous step.</li> </ul>
Describing the data	<a href="#">Data Description</a>	Create the data description using the <i>Analyze</i> button.
Selecting the Target Variable and a weight variable	<a href="#">Selecting the Target Variable</a>	Select the variable <i>RESPONSE</i> as your target variable.
Selecting explanatory variables	<a href="#">Selecting Variables</a>	Exclude the variables <i>REFID</i> and <i>REF_DATE</i> from the list of variables to be used for modeling.

### 4.6.2.3 Intermediate Method: Results

The *Training the Model* screen shows the predictive power and the prediction confidence, which indicate the quality and robustness of the model generated on the customers' reference data combined with the demographic data.

The table below compares these results with the ones obtained for the simple method.

	Predictive Power	Prediction Confidence
Simple Method	0.198	0.976
Intermediate Method	0.199	0.971

Taking the geographic data into account has led you to obtain a model that has the same quality, which means that a regression/classification model does not find more information in the added data. However when adding the events data, the geographic data may add to the model quality. Before making strategic decisions and taking action, the last test you need to perform is to determine whether taking the events data you own into account can take you further into enhancing your model.

## 4.6.3 Overall Method: Adding Transaction Data

### 4.6.3.1 Overall Method: Description

Although the intermediate method resulted in a model that was both accurate and robust, you still have transaction data ([Sales](#) table) at your disposal. The last method you will use consists of taking all the data you own into account for building your model, that is:

- The reference data
- The demographic data
- The transaction data

The transaction data contains event data, where there may be zero, one, or many entries for each customer. Since the regression/classification model is not designed to process such data, you will need to use a data manipulation feature. That is where event logging comes into play!

Event logging is a data manipulation feature that combines and compresses event data in a manner that makes it available to Modeler. Event Logging adds no difficulty to the modeling process. All you have to do to is to configure specific application settings in SAP Predictive Analytics.

#### i Note

For a longer description of event logging, see Long Description.

As with the simple and intermediate methods, you will need to join the tables to be used - that is, the [Customers](#), [Mailings1\\_2](#) and [Demog](#) tables.

### 4.6.3.2 Overall Method: Modeling Process

Compared to using only the regression/classification model as you did for the first two methods, using event logging means performing the four additional steps below:

1. Selecting Events Data Source.



2. Describing Events Data.
3. Setting event logging Parameters.
4. Setting event logging Events Statistics

You will have to go through the same overall modeling process as the one you went through for the two other methods, with event logging steps taking place just:

- After the step Describing Data,
- Before the step Selecting the Target Variable.

## Summary of the Modeling Settings to Use

The table below summarizes the modeling settings you must use for the overall method. Except for the four steps specific to event logging, the other settings are similar to the ones used for the intermediate method.

Event logging steps are presented in details in the following sections.

For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

Task(s)	Screen	Settings
<ul style="list-style-type: none"> <li>• Selecting a partition strategy</li> <li>• Specifying the data source</li> </ul>	<i>Data to be Modeled</i>	<ul style="list-style-type: none"> <li>• Partition strategy: <i>Random</i></li> <li>• Select the option <i>ODBC Source</i> .</li> <li>• In the <i>ODBC Source</i> field, specify the data source to be used</li> <li>• In the <i>Training</i> field, select the analytical dataset <i>KEL_ADS_IntermediateMethod</i> .</li> </ul>
Describing the data	<i>Data Description</i>	<ul style="list-style-type: none"> <li>• Create the data description using the <i>Analyze</i> button.</li> </ul>
Selecting an Event Data Source	<i>Events Data</i>	<p>Select the option <i>Data Base</i> .</p> <p>In the <i>Folder</i> field, specify the data source to be used</p> <p>In the <i>Events</i> field, select the table <i>Sales</i> .</p>
Describing Event Data	<i>Events Data Description</i>	Use the description <i>sales_desc</i>
Setting Event Logging Parameters	<i>Event Logging Parameters Settings</i>	<p>Fill in the fields with the following values:</p> <p><i>Column for Join / Reference Dataset: REFID</i></p> <p><i>Column for Join / Log Dataset: REFID</i></p> <p><i>Log Date Column / Log Dataset: EVENT_DATE</i></p>

Task(s)	Screen	Settings
Setting Event Statistics	<a href="#">Event Logging Variables Selection for Functions</a>	Check <i>Selected Aggregate</i> on the line <i>AMOUNT / Sum</i> .  Reference Date: <i>Variable / REF_DATE</i>  Period Type: <i>Simple</i>  <i>Period Definition</i> : Define 3 successive period(s) of 1Month / Starting 3 Months before <i>REF_DATE</i>
Selecting the target variable and a weight variable	<a href="#">Selecting the Target Variable</a>	Select the variable <i>RESPONSE</i> as your target variable.
Selecting explanatory variables	<a href="#">Selecting Variables</a>	Exclude the variables <i>REFID</i> and <i>REF_DATE</i> from the list of variables to be used for modeling.

## Selecting the Type of Model to Create

In the Explorer area of the main menu, click the option [Perform anEvent Log Aggregation](#).

The screen [Add a Modeling Feature](#) is displayed.

Click the option [Add a Classification/Regression](#).

### **i** Note

When building a model you can either create aggregations or add extra transformations such as a classification/regression or a clustering/segmentation using Modeler.

### 4.6.3.2.1 Selecting Reference Data

The Events Data Source screen lets you specify the data source to be used as transaction data.

For this Scenario:

- The *Folder* field should already be filled in with the name of the data source that you specified on the Data to be Modeled screen.
- In the *Events* field, select the table Sales.

This analytical dataset contains the customers that were contacted in the two first waves of incentive mailings that your marketing department sent.

To Select Events Data

1. Select the type of your data source (Text Files, ODBC, ...).
2. In the *Folder* field, specify the folder where your data source is stored.

3. In the *Events* field, specify the access path and name of your data source.
4. If needed, press the *Enter* key to activate the *Next* button.
5. Click the *Next* button. The screen *Events Data Description* is displayed.

### 4.6.3.2.2 Describing Events Data

The screen *Events Data Description* lets you describe your events data, offering you the same options as the screen *Data Description*. For detailed procedures on how to set parameters on this screen, see *Describing the Data*.

For this Scenario

1. Create the data description by clicking the *Analyze* button.  
The description of your data appears.
2. Make sure the description is similar to the one on the screen below, that is:
  - The *EVENTID* variable must be set as integer and nominal and be specified as a key variable.
  - The *REFID* variable must be set as integer and nominal, be specified as a key variable and as an order variable.
  - The *EVENT\_DATE* variable must be set as date and continuous.
  - The *AMOUNT* variable must be set as number and continuous.
3. Click the *Next* button. The data description is taken into account.  
The screen *Event Logging Parameters Settings* appears.

### 4.6.3.2.3 Setting Event Logging Parameters

The screen *Event Logging Parameters Settings* lets you to set some parameters by performing the following tasks:

- Join your reference data with your transaction data.
- Select the variable to be used as the transaction variable.
- Set time parameters required for the generation of event logging statistics.
- Specify the storage type for internal computation.
- Specify a prefix for variables generated by the event logging.

For this scenario, the reference date will be the date when the mailing was sent, that is *REF\_DATE*.

You are interested in each customer purchases on the three months preceding the date on which they were sent the incentive mailing by your marketing department.

The panel *Event Logging Parameters Settings* looks like the one below.

	Events Data Set:	Reference Data Set:
Columns for Join:	<input type="text" value="OS"/>	<input type="text" value="workclass"/>
Events Date Column:	<input type="text" value="DAT"/>	
<hr/>		
Storage Type	<input type="radio"/> On Disk	<input checked="" type="radio"/> Memory
<hr/>		
Variables Prefix:	<input type="text" value="el"/>	

The table below summarizes the modeling settings to use for the screen [Event Logging Parameters Settings](#).

Field	Setting
<i>Column for Join / Reference Dataset</i>	<i>REFID</i>
<i>Column for Join / Log Dataset</i>	<i>REFID</i>
<i>Log Date Column / Log Dataset</i>	<i>EVENT_DATE</i>
<i>Reference Date</i>	<i>Variable</i>

## Columns for Join

Because event logging aggregates two sets of data, there must be a column that will be the pivot for this operation, and this column has to be present in both datasets. This parameter allows that: specifying for both datasets which column is holding the identifiers that will enable the aggregation to take place. Thus, it is not required that these columns have the same name as long as they are semantically equivalent.

## Log Date Column

Knowing the pivot column is not enough in order to aggregate the datasets. Events are aggregated for each identifier, but within specific time windows as well. Hence, events timestamps have to be present in the dataset; this parameter specifies which column holds this information.

## Storage type

Aggregating datasets may consume a lot of memory. Selecting the *On Disk* option rather than *Memory* causes the application to store its internal computations in a temporary file, thus lowering virtual memory consumption (but lowering speed as well).

### i Note

Due to their completely different structures, the two modes are not comparable in terms of internal storage size.

## Variables Prefix

Because event logging generates additional variables, it is possible to recognize them easily by specifying a prefix to their names.

### 4.6.3.2.4 Selecting Event Logging Statistics

The Variables Selection for Functions screen lets you specify:

- the aggregates you want to calculate on transaction data (or events),
- the periods over which you want to calculate these aggregates,
- and possibly filters on the data.

For this Scenario, you decide to calculate for every customer the amount of their purchases on the three months preceding the date on which they were sent the incentive mailing by your marketing department. That way, you should be able to determine and understand your customers' purchase behaviors.

#### 4.6.3.2.4.1 Specifying the Aggregates

The screen is split in two parts:

- the upper part of the panel allows you to select the meta operators to apply on the count operation. This operation, which is always done, counts the number of events in each defined period (see table below for details on the meta-operators).
- the table in the lower part of the panel displays the available aggregates.

For this Scenario:

- Do not select any meta-operators on the count operation.
- For the variable Amount, that gives for every customer the amount their individual purchases, select the function Sum, that calculate for every customer the sum of all their purchases amounts.
- Do not select any meta-operators.

To Create an Aggregate on Transaction Data:

1. Click the button *1- Click to Define the Aggregates* to display the corresponding section.

#### i Note

This section is displayed by default when opening the panel *Event Logging Variables Selection for Functions*.

2. You can choose to apply meta-operators on the *Count* operator. This operator, which is always computed, counts the number of events in each defined period (see table below for details on the meta-operators).
3. In the table, check the box *Selected Aggregate* corresponding to the variable and the operator to use to create the aggregate. Four types of operators are available for each variable: *Max, Sum, Average, and Min*.

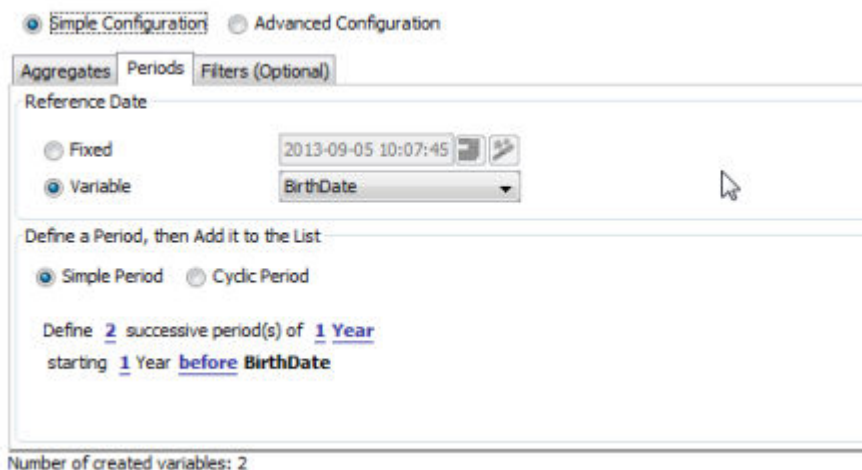
Depending on the results you want to obtain, you can apply meta-operators on the aggregates you are creating. The five types of meta-operators can be applied concurrently.

The meta-operator...	adds to the aggregate...	calculates...
<i>Variation</i>	one column for each couple of consecutive periods (that is n-1)	the difference between the values of two consecutive periods for all the periods of the aggregate
<i>Percentage Variation</i>	one column for each couple of consecutive periods	the difference in percentage between the values of two consecutive periods for all the periods of the aggregate
<i>Accumulation</i>	one column for each period, except the first one	the current total accumulation for each period of the aggregate
<i>Backward Accumulation</i>	one column for each period, except the last one	the current total accumulation for each period of the aggregate calculated backwards
<i>Global Sum</i>	one column	the sum of all periods values of the aggregate

4. By default all numerical variables are displayed in the table, however you can choose to display only the continuous variables by checking the box *Display only Continuous Variables*.
5. Click the button *2 - Click to Define the Periods* to select the periods over which the aggregates will be calculated.

## 4.6.3.2.4.2 Setting the Periods

The panel Event Logging Variables Selection for Functions looks like the one below.



For this scenario:

- The reference date will be the date when the mailing was sent, that is *REF\_DATE*.
- You are interested in each customer purchases each month on the three months preceding the date on which they were sent the incentive mailing by your marketing department.

The table below summarizes the modeling settings to use.

Field	Setting
<i>Reference Date</i>	<i>Variable / REF_DATE</i>
<i>Period</i>	<i>Simple</i>
<i>Period Definition</i>	Define 3 successive period(s) of 1Month Starting 3 Months before <i>REF_DATE</i>

To Define the Periods:

1. Select the type of the reference date by checking either *Fixed* or *Variable*. The reference date is the starting date used to define the periods to be analyzed. This date can be:
  - either *Fixed*, that is, a specific date, for example 05-24-2004,
  - or *Variable*, that is, a date defined by a variable from the dataset. For example, the first purchase date, the churn date, and so on.
2. Depending on the option selected above, either specify a fixed date in the corresponding text field, or select a date variable from the available date variables in the reference table.
3. Select the type of period you want to define. There are two types of periods:
  - *Simple Periods*, which are successive periods of a defined length
  - *Cyclic Periods*, which are recurring periods such as every day, every month, every year, and so on.
4. The period definition interface depends on the period type you have selected.

If you want to define a Simple Period:

1. In the displayed sentence, click the word or number corresponding to the parameter to modify. The periods are defined by:
  - a number of periods,
  - a duration,
  - a starting date, which can be before or after the reference date previously selected.
2. Repeat this step for all the parameters to define.

If you want to define a Cyclic Period:

1. Check the option *Cyclic Period* in the frame *Define a Period...*
2. In the drop-down list Create one period for each, select the cycle.

You can apply the periods to the entire dataset or select one or more successive periods:

- select the option *All Events* to get the cyclic periods in the whole dataset.
- select the option *Only over the following time frame* to limit the cyclic periods to a specific time frame. Defining this time frame amounts to creating simple periods.

### **i** Note

The number of variables created by event logging is indicated at the bottom of the panel. This number grows with the number of periods defined.

5. Once you have defined the periods you can set filters or pivots, by clicking the button *Optional - Click to Define the Filters*.

### 4.6.3.2.4.3 Setting Filters (optional)

The option *Filter* allows you to filter your data depending on the variables values.

The option *Pivot* allows you to create one event logging engine for each selected value. Creating a pivot amounts to creating a filter on one category for each selected categories.

For this Scenario, do not define any filters or pivots.

To Define a Filter or a Pivot:

1. Click the button *Optional - Click to Define the Filters* to display the filter interface.
2. Check the option *Filter or Pivot* depending on how you want to filter your data.
3. Select the *Filter* Type. Two types of filters are available:
  - by *Excluded Value(s)*, which means that the filter will exclude events containing the values listed in the table,
  - by *Kept Value(s)*, which means that the filter will keep only the events containing the values listed in the table.
4. Select the variable to filter by in the *Variables* drop-down list.
5. To add categories to the table, you can:
  - either automatically extract the selected variable categories by clicking the magnifier button located next to the list and then select the values to keep or exclude by checking the corresponding Selection box.
  - or enter a value in the field *New Category* and click the + button.

#### i Note

The number of variables created by event logging is indicated at the bottom of the panel. This number grows exponentially when filtering by pivot. The higher the number of variables, the longer the model learning.

### 4.6.3.2.4.4 Selecting the Variables

Once event logging has run on the data and computed the requested aggregates, new variables have been created. These variables appear in the list Explanatory Variables Selected of the panel Selecting Variables.

Different variables are created depending on the selected operators and meta operators. Different elements are used to build the variables names:

<prefix>	By default, the prefix is set to el , but can be modified.
<Engine>	<ul style="list-style-type: none"><li>• Name of the event logging engine, when several event logging engines are used (when using a pivot or the advanced configuration).</li><li>• Engine if only one event logging engine is used.</li></ul>



- `<Period, Pn>`

Number of the current period.

The periods are numbered from 0. If there are 4 periods from -2 years to +2 years, the periods are numbered 0, 1, 2, 3. With 0 being the oldest one and 3 the last one.

### i Note

Note that the numbering of output columns for cyclic and non-cyclic periods is different. For example, suppose that there is a fixed date of 8/18/2007, and you ask for the last 24 months (non-cyclic). Then output columns will have months numbered 0 through 23, with month 23 having entries between 7/19/2007 and 8/18/2007. On the other hand, if you specify cyclic by months, then output columns will be numbered 0 (Jan) through 11(Dec), regardless of the fixed date.

<code>&lt;n&gt;</code>	Total number of periods minus 1. For the example above n=3.
<code>&lt;Operator&gt;</code>	The operator applied
<code>&lt;Meta&gt;</code>	The meta operator applied
<code>&lt;Variable&gt;</code>	The variable on which the operator applies

The following table details the generated output variables.

Operator / Meta Operator Name	Syntax	Example
<i>Count (CNT)</i>	<code>&lt;prefix&gt;_&lt;engine&gt;_CNT_&lt;PeriodNumber&gt;</code>	el_Engine_CNT_0
<i>Sum (SUM)</i>	<code>&lt;prefix&gt;_&lt;engine&gt;_&lt;operator&gt;_&lt;period&gt;_&lt;variable&gt;</code>	el_Engine_SUM_1_OS
<i>Min (MIN)</i>		el_Engine_MIN_0_OS
<i>Max (MAX)</i>		el_Engine_AVG_3_OS
<i>Average (AVG)</i>		el_Engine_MAX_2_OS
<i>Variation (DIF)</i>	<code>&lt;prefix&gt;_&lt;engine&gt;_&lt;Meta&gt;_&lt;Operator&gt;_P0_P 1_&lt;variable&gt;</code>	el_Engine_DIFF_MIN_0_1_OS
<i>Percentage Variation (PER)</i>	<code>&lt;prefix&gt;_&lt;engine&gt;_&lt;Meta&gt;_&lt;Operator&gt;_P1_P 2_&lt;variable&gt;</code>	el_Engine_PER_SUM_1_2_OS
	...	
	<code>&lt;prefix&gt;_&lt;engine&gt;_&lt;Meta&gt;_&lt;Operator&gt;_Pn-1_P n_&lt;variable&gt;</code>	

<i>Accumulation (ACC)</i>	<prefix>_<engine>_<Meta>_<Operator>_P0_P 1_<variable>	eI_Engine_ACC_MIN_0_1_OS
	<prefix>_<engine>_<Meta>_<Operator>_P0_P 2_<variable>	eI_Engine_ACC_CNT_0_2
	...	
	<prefix>_<engine>_<Meta>_<Operator>_P0_P n_<variable>	
<i>Back Accumulation (BACK)</i>	<prefix>_<engine>_<Meta>_<Operator>_P0_P n_<variable>	eI_Engine_BACK_MAX_ 2_3_OS
	<prefix>_<engine>_<Meta>_<Operator>_P1_P n_<variable>	eI_Engine_BACK_CNT_0_3
	...	
	<prefix>_<engine>_<Meta>_<Operator>_Pn-1+P n_<variable>	
<i>Global Sum (FUL)</i>	<prefix>_<engine>_<Meta>_<Operator>_0_n_<variable>	eI_Engine_FUL_AVG_0<->3_OS

### 4.6.3.3 Overall Method: Results

The screen shows the predictive power and the prediction confidence, which indicate the quality and robustness of the model generated on the customers' reference data combined with the demographic and the transaction data.

The table below compares these results with the ones obtained for the simple and intermediate methods.

	Predictive Power	Prediction Confidence
<i>Simple Method</i>	0.198	0.976
<i>Intermediate Method</i>	0.199	0.971
<i>Overall Method</i>	0.399	0.986

Taking the transaction data into account has led you to obtaining a model that has a much better predictive power than with the two other methods. Thanks to event logging, you have been able to make the most out of your transaction data. On the basis of the model obtained, you can develop an extremely targeted marketing campaign that will help you in maximizing your profit.

## 4.7 Step 3 - Making a Decision and Taking Action

### 4.7.1 Identifying the Customers to Contact

You now need to clearly identify who these customers are among all the customers referenced in your database.

You will do this by:

- Applying the model to the customers' reference data table.
- Extracting the customers to contact from the application result file.

#### 4.7.1.1 Applying the Model to the Reference Data Table

For this Scenario, the table below summarizes the modeling settings to apply the model.

For a detailed procedure, see To Apply the Model to a New Dataset.

Task	Screen	Settings
<a href="#">Creating the Application Dataset</a>	<a href="#">Data Manipulation</a> > <a href="#">Create a New Analytical Dataset</a>	<p>source table: <i>Customers</i></p> <p>Fields</p> <ul style="list-style-type: none"><li>• Rename ID alias in <i>REFID</i></li></ul> <p>New Fields &gt;Function:</p> <ul style="list-style-type: none"><li>• <a href="#">New Function</a> &gt; <a href="#">Miscellaneous Operators</a> &gt; <a href="#">Constant</a></li><li>• Type: <i>DateTime</i></li></ul> <p>Value: <i>'2007-07-01 12:00:00'</i></p> <p>Name: <i>REF_DATE</i></p> <p>Merge:</p> <ul style="list-style-type: none"><li>• source field: <i>GEOID</i> / target table: <i>Demog</i> / target field: <i>GEO_ID</i></li></ul> <p>Filter:</p> <ul style="list-style-type: none"><li>• On: <i>REFID</i> / Operator: <i>Greater than</i> / Right Operand: <i>Constant(Integer)=59999</i></li></ul> <p>Save</p> <ul style="list-style-type: none"><li>• name: <i>KEL_ADS_ApplyDataSet</i></li></ul>

Task	Screen	Settings
Selecting the Events Dataset	Apply Events Data	<ul style="list-style-type: none"> <li>• Select the option Data Base .</li> <li>• in the Folder field, specify the data source to be used.</li> <li>• In the Events field, select the analytical dataset Sales .</li> </ul>
Setting the Reference Date	Apply Events Data	<ul style="list-style-type: none"> <li>• For the Reference Date , check the Variable option.</li> <li>• In the list Reference Column , select REF_DATE .</li> </ul>
Selecting the Application Dataset	▶▶ Applying the Mode ▶ Application Dataset ▶	<ul style="list-style-type: none"> <li>• Select the option <i>Data Base</i> .</li> <li>• In the Folder field, specify the data source where the application dataset is located.</li> <li>• In the Data field, select the analytical dataset <i>KEL_ADS_ApplyDataSet</i> .</li> </ul>
Selecting the Generation Options	▶▶ Applying the Model ▶ Generation Options ▶	<ul style="list-style-type: none"> <li>• In the <i>Generate</i> list, select <i>Decision</i> .</li> <li>• In the <i>Mode</i> list, select <i>Apply</i> .</li> </ul>
Setting where the Results will be Saved	▶▶ Applying the Model ▶ Results Generated by the Model ▶	<p>Select the option <i>Data Base</i> .</p> <p>This will lead into creating a table containing the application results in your database</p> <ul style="list-style-type: none"> <li>• In the <i>Folder</i> field, specify the data source to be used.</li> <li>• In the <i>Data</i> field, specify a name – for example, <i>KXSCORE</i> – for the application results table to be created in your database.</li> </ul>

The following panel is set with this information:

**Applying the Model**

Application Data Set

Data Type: Data Base

Folder: KEL Scenario

Data: KEL\_ADS\_ApplyDataSet

Apply Mapping

Generation Options

Generate: Decision

Mode: Apply

View Generated Outputs

Use direct apply in the database.

Results Generated by the Model

Data Type: Data Base

Folder: KEL Scenario

Data: kxscore

Define Mapping

Help Cancel Previous Apply

To Apply the Model to a New Dataset:

1. On the screen *Using the Model*, click the option *Apply the Model to a New Dataset*.  
The screen *Apply Events Data* appears.
2. In the section *Events Dataset*, select the format of the data source (Text files, Data Base, ...).
3. Click the *Browse* button to select:
  - In the *Folder* field, the folder or data base which contains your dataset,
  - In the *Events* field, the name of the file, table or analytical dataset corresponding to your dataset.
4. In the section *Fixed*, select the *Reference Date type*.
  - *Fixed* corresponds to a constant date you define.
  - *Variable* corresponds to a date variable existing in the events dataset.
5. Click the *Next* button.  
A dialog box appears asking to confirm the event dataset replacement.
6. Click the *Yes* button.  
The screen *Applying the Model* appears.
7. In the section *Application Dataset*, select the format of the data source (Text Files, Data Base, ...).
8. Click the *Browse* button to select:
  - In the *Folder* field, the folder or data base which contains your dataset,
  - In the *Data* field, the name of the file, table or analytical dataset corresponding to your dataset.
9. In the section *Generation Options*, select type of results you want to obtain in the *Generate* drop-down list.
10. In the *Mode drop-down list*, select the application mode of the model.

### i Note

If you select the Keep only outliers option, only the outlier observations will be presented in the results file obtained after applying the model.

11. In the section *Results Generated by the Model*, select the file format for the output file (Text Files, Data Base, ...).
12. Click the *Browse* button to select:
  - In the *Folder* field, the folder or data base in which you want to save the results.
  - In the *Data* field, the name of the file or table in which the results will be saved.
13. Click the *Apply* button.

If you have selected the *Decision* option in the *Generate* list, the *Classification Decision* screen appears.

## 4.7.1.1.1 Defining the Number of Customers to Contact

According to the income you have to generate – that is \$1,500,000 – you determined that you had to contact 1,667 people who would respond positively to your incentive mailing (Your Business Issue (see Your Objective)). These 1,667 responders represent 11.1% of the potential responders contained in the database. Taking into account that, though the KI is significantly higher with the last method, it is below 0.5, you should select a slightly higher percentage of detected target to be sure to reach your goal.

In this Scenario:

- Select *12% of Detected Target*.
- Set the *Total Population* to 300,000.

To Define the Number of Customers to Contact:

1. In the field *% of Detected Target*, enter *12*.

The cursor scale moves to 0.9% (see above), thus indicating that you need to send the incentive mailing to 0.9% of the customers referenced in your database – that is to 2,700 people – to contact 12% of responders.

2. To estimate the number of responders correctly identified by the model, you can use the Confusion Matrix.

In the field Total Population, enter the total number of customers in your apply dataset, that is, 300,000. By default, the Total Population is the number of records in the Validation dataset.

The following table details how to read the confusion matrix.

	<i>Predicted</i> [Target Category] Positive Observations Pre- dicted	<i>Predicted</i> [Non-target Cate- gory] Negative Observa- tions Predicted	<i>Total</i>
<i>True</i> [Target Category] Ac- tual Positive Observations	Number of correctly pre- dicted positive observations	Number of actual positive observations that have been predicted negative	Total number of actual posi- tive observations

<i>True</i> [Non-target Category] Actual Negative Observations	Number of actual negative observations that have been predicted positive	Number of correctly predicted negative observations	Total number of actual negative observations
<i>Total</i>	Total number of positive observations predicted	Total number of negative observations predicted	Total number of observations in the dataset

The Classification Rate, that is, the percentage of data accurately classified by the model when applied on the training dataset, is indicated below the confusion matrix.

You can also visualize the profit you will make by using the *Cost Matrix*.

1. Enter the average revenue generated from a person that responds positively in the cell *True true/Predicted True*.  
The Profit generated is displayed on the right of the *Cost Matrix*.
2. Click the Next button.  
The Applying the Model screen appears.  
Once application of the model has been completed, the results files of the application is automatically saved in the location that you had defined from the screen *Applying the Model*.

## 4.7.1.2 Extracting the Customers to Contact

At the end of the model application, you will find a table containing the application results in your database. This table has been created on the fly by the application. If you followed the recommended settings for this scenario, it is named *KXSCORE*.

The figure below shows the first lines of this table.

REFID	RESPONSE	rr_RESPONSE	decision_rr_RESPONSE	proba_decision_rr_RESPONSE
60000		-0.0235300860769003	false	0.971626223905274
60001		0.0207639974128645	false	0.966360266554785
60002		-0.00537067843090418	false	0.969496015933164
60003		-0.0195848182282352	false	0.970795303438372
60004		0.0160646123909129	false	0.968212725140976
60005		-0.00399167201770405	false	0.969439108391276
60006		0.0779777858150452	true	0.7098272727273347
60007		0.00535575044376382	false	0.969053367737587
60008		0.008045048157411	false	0.968942388324566
60009		0.00648700488736501	false	0.969006684191362
60010		-0.0478785451677473	false	0.97515117062903
60011		-0.0433511945737997	false	0.975150913312366
60012		-0.0268978829958147	false	0.972335522106471

The table below describes the application results table.

The column...	Contains...
<i>REFID</i>	The customers' unique identifiers
<i>RESPONSE</i>	No values. You will fill in this column once you know the actual responses of the customers to be contacted. That way, you can compare them with the responses predicted by the model.

The column...	Contains...
<i>rr_RESPONSE</i>	The <i>score</i> , or value predicted by the model for the target variable of each observation.
<i>decision_rr_RESPONSE</i>	The <i>decision</i> made by the model indicating whether the customer should be contacted or not.
<i>proba_decision_rr_RESPONSE</i>	The <i>probability</i> found by the model that the customer responds positively.

For this Scenario, you will extract the customers that you need to contact, that is who should respond positively to your campaign, from the rest of the table.

1. To Select the Customers to Contact:
  - a. Create a new analytical dataset, using the *KXSCORE* table as the data source main table.
  - b. Add a filter on the field *decision\_rr\_RESPONSE* to select only the customers for which the decision equals "*true*".
  - c. Click the *Next* button.
  - d. Save the dataset as *ADS\_CustomersToContact*.

The result of this selection is a table listing 2,583 customers to contact. You will now need to save this table in your database by using the Data Transfer feature.

2. To Save the Selected Data into a Table:
  - a. Open the *Data Transfer* feature.
  - b. Select the analytical dataset *ADS\_CustomersToContact* as data source.
  - c. Click the *Next* button.
  - d. Click the *Analyze* button to create the data description.
  - e. Set the *REFID* variable as a key.
  - f. Click the *Next* button.
  - g. Name the name table *CustomersToContact*.
  - h. Click the *Next* button.

According to the data mining model, all the customers contained in this table are the ones within your database who are the most likely to respond positively to the incentive mailing. With their *ID*, you can now use the *Reference Data* to extract their e-mail addresses and send them the incentive mailing.

## 4.7.2 Your Marketing and Sales Campaign: Wrap-Up

Two months after you sent the incentive mailings to your customers, you need to verify the efficiency of your campaign. To do this, you need to compare the actual responses of the customers with the ones predicted by the model that you used to target your incentive mailings.

The actual responses of the customers are contained in the table *Mailing3* and the list of customers that you contacted are contained in the table *CustomersToContact* (created using the Data Transfer feature). You will need to:

- First create an analytical dataset merging the *Mailing3* table and the *CustomersToContact* table.
- Use the Descriptive Statistics feature to see the results.



To Merge the Tables *Mailing3* and *CustomersToContact*:

1. In the *Data Transformation* section of the Start menu, click the option *Define Dataset*.
2. In your data base, select the *Mailing3* table as the data source.
3. Click the *Next* button.
4. In the *Merge* tab, select:
  - o *REFID* as the Source Field
  - o *CustomersToContact* as the Target Table
  - o *REFID* as the Target Field
5. Click the button *New Merge*.
6. In the *Fields* tab, check the Visibility option only for the field *RESPONSE* of the *Mailing3* table and the field *decision\_rr\_RESPONSE* of the *CustomersToContact* table.
7. Create a filter to select all customers having actually responded positively to the mailing, that is, the customers for which the field *RESPONSE* of the *Mailing3* table equals *true*.
8. Click the *Next* button.
9. Save the analytical dataset as *KEL\_ADS\_CheckResults*.

To Compare the Model to the Actual Results:

10. In the *Data Toolkit* section of the Start menu, click the option *Descriptive Statistics*.
11. Select *KEL\_ADS\_CheckResults* as the input dataset.
12. Click the *Next* button.
13. Click the *Analyze* button to create the data description.
14. Click the *Next* button.
15. Do not select a target variable.
16. Click the *Next* button.
17. There are no estimators to define, click the *Next* button.
18. Click the *Generate* button.

The dataset contains 1809 records. Using the Automated Analytics model, you contacted 2,583 customers intending to hit at least 1667 responders and generate an income of \$1,500,000.

Not only have you fulfilled your primary business objective, but you have also realized an unexpected profit margin. Knowing that the average income generated by each responder is \$900, the model enabled you to generate a profit margin of \$127,800 ( $\$900 * 142$  unexpected responders).

To finish up with this campaign, there is one last thing you need to do:

Let's celebrate all this with your boss!

# 5 Sequence Analysis Scenarios

## 5.1 Introduction to Application Scenarios

In these scenarios, you are the Marketing Director of an E-commerce company and you want to increase the profitability of your Web site. You have the budget to launch a major marketing initiative, but you're not sure what kind of campaign would be the most effective. Due to market pressures, you only have the time and money to test a few campaigns before launching a major initiative. The two key metrics that are being used to measure the performance of the Web site are the "conversion rate" and "stickiness". The conversion rate of a site is the percentage of visits that result in a purchase. At this time, your Web site has a conversion rate of 4%, meaning that 4 out of every 100 visitors purchase at least one item. The stickiness of a Web site is a measure of the number of pages viewed by each visitor. The more pages a visitor views, the more likely they are to purchase something. Your Web site is averaging about 10 pages per visit.

In order to achieve rapid insight into the different groups of visitors to your Web site, you have decided to use Modeler – Segmentation/Clustering to group the population with respect to their buying behavior and site abandonment. The goal of the analysis is to get descriptions of the groups of visitors who tend to purchase items frequently, and the indicators that a session is about to end. You already know the following basic facts about your Web site:

- An average of 50,000 visitors come to the Web site each day.
- For the 2000 sessions that result in a purchase each day, the average amount spent is \$181.
- The average profit margin for the Web site is 5%, so each purchase results in an average profit of \$9.05, resulting in \$18,100 of profit per day.
- There are four main entry points for the site – The home page, the members' home page, the sweepstakes page, and the specials page.
- The checkout process has five steps, all with the word "order" in the file name.
- Your site does not use "cookies" or require a login for your members, so each session is effectively anonymous unless a purchase is made.

The information that is available for analysis consists of the Web logs. Your DBA has pulled out a list of the sessions from a single day of traffic, along with a flag indicating if the session resulted in a purchase (the existence of "order5.tmp" in a session indicates a purchase). Along with the list of sessions, the parsed log from the day is also available. Since the information from the Web log is not aggregated for analysis, you will need to use the Data Manager – sequence coding prior to running the Modeler – Classification/Regression or Modeler – Segmentation/Clustering.

### Scenario 1

You will start by using sequence coding to create counts of each Web page that was viewed by each visitor, followed by a targeted segmentation with "purchase" as the target. This will give you a simple description of the different groups browsing your Web site, and the different conversion rates for each group.

## Scenario 2

In this scenario you want to predict when a visitor is going to leave your Web site. Your idea is to offer a \$5 coupon to visitors who are likely to leave in the hope of increasing the site stickiness. To achieve that, you will create a sequence coding model using intermediates sequences with the *FirstLast* option for the pages viewed. The intermediate sequence option will automatically create an appropriate target variable for determining which behaviors indicate the end of a session.

## 5.2 Introduction to Sample Files

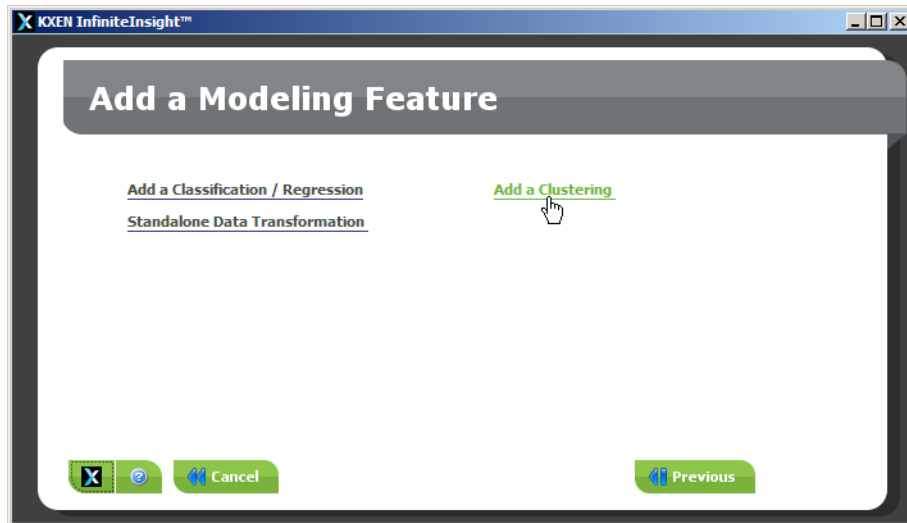
This dataset contains a single day of Web traffic from an E-commerce site in December 1999. The site content was served by a Broadvision server, but no "cookies" or login was required, making the sessions effectively anonymous.

File	Description
<code>session_purchase.csv</code>	list of sessions and binary purchase target (50581 rows)
<code>session_purchase_desc.csv</code>	description for <code>session_purchase.csv</code>
<code>file_view.csv</code>	log of files requested from Broadvision server (532860 rows)
<code>file_view_desc.csv</code>	description for <code>file_view.csv</code>
<code>session_purchase_skip.csv</code>	variable skip list for Scenario 1. These are the variables for which the value would not be known until the session had ended.
<code>session_continue_skip.csv</code>	variable skip list for Scenario 2

You can download the sample files from the SAP Help Portal at <http://help.sap.com/pa>.

## 5.3 Scenario 1: Segment Visitors to Understand Purchase Behavior Using File Counts

1. In *SAP Predictive Analytics* main menu, select the option *Perform a Sequence Analysis* in the *Data Manager* section.
2. The screen *Add a Modeling Feature* is displayed.



3. Click on the option *Add a Clustering*.

#### i Note

When building a model you can either simply analyze the sequences or add extra transformations such as a Classification/Regression (Modeler - Regression/Classification) or a Clustering/Segmentation (Modeler - Segmentation/Clustering).

## 5.3.1 Step 1 - Selecting the Data

### 5.3.1.1 Selecting a Data Source

For this Scenario, the file `session_purchase.csv` contains a list of session IDs and whether each session has led to a purchase or not. This will be referred to as the Reference dataset for Sequence Coding. A Sequence Coding Reference dataset must have a single variable unique primary key. If the primary key is non-unique or spread out over several variables, sequence coding will not function properly.

1. On the screen *Data to be Modeled*, select the data source format to be used (`Text files`, `ODBC`, ...).

Note that SAP HANA information views are not supported by Data Manager, only standard SAP HANA tables or views can be used as data source.

2. Use the *Browse* button on the right of the *Folder* field to select the folder where you have saved the sample files.
3. Click the *Browse* button next to the *Estimation* field and select the file `session_purchase.csv`.

The name of the file will appear in the *Estimation* field.

4. Click the *Next* button.

## 5.3.1.2 Describing the Data

### Why Describe the Data Selected?

In order for the application features to interpret and analyze your data, the data must be described. To put it another way, the description file must specify the nature of each variable, determining their:

- Storage format: number (*number*), character string (*string*), date and time (*datetime*) or date (*date*).

#### i Note

When a variable is declared as *date* or *datetime*, the Date Coder feature (KDC) automatically extracts date information from this variable such as the day of the month, the year, the quarter and so on. Additional variables containing this information are created during the model generation and are used as input variables for the model.

KDC is disabled for Time Series.

- Type: *continuous*, *nominal*, *ordinal* or *textual*.

### How to Describe Selected Variables

To describe your data, you can:

- Either use an existing description file, that is, taken from your information system or saved from a previous use of the application features,
- Or create a description file using the *Analyze* option, available to you in the application. In this case, it is important that you validate the description file obtained. You can save this file for later re-use. If you name the description file `KxDoc_<SourceFileName>`, it will be automatically loaded when clicking the *Analyze* button.

#### ⚠ Caution

The description file obtained using the *Analyze* option results from the analysis of the first 100 lines of the initial data file. In order to avoid all bias, we encourage you to mix up your dataset before performing this analysis.

Each variable is described by the fields detailed in the following table:

The Field...	Gives information on...
<i>Name</i>	the variable name (which cannot be modified)
<i>Storage</i>	the type of values stored in this variable: <ul style="list-style-type: none"><li>• <i>Number</i>: the variable contains only "computable" numbers (be careful a telephone number, or an account number should not be considered numbers)</li><li>• <i>String</i>: the variable contains character strings</li><li>• <i>Datetime</i>: the variable contains date and time stamps</li><li>• <i>Date</i>: the variable contains dates</li></ul>

The Field...	Gives information on...
<i>Value</i>	<p>the value type of the variable:</p> <ul style="list-style-type: none"> <li>• <i>Continuous</i> : a numeric variable from which mean, variance, etc. can be computed</li> <li>• <i>Nominal</i>: categorical variable which is the only possible value for a string</li> <li>• <i>Ordinal</i>: discrete numeric variable where the relative order is important</li> <li>• <i>Textual</i>: textual variable containing phrases, sentences or complete texts</li> </ul> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px;"> <p><b>⚠ Caution</b></p> <p>When creating a text coding model , if there is not at least one textual variable , you will not be able to go to the next panel.</p> </div>
<i>Key</i>	<p>whether this variable is the key variable or identifier for the record:</p> <ul style="list-style-type: none"> <li>• <i>0</i> the variable is not an identifier;</li> <li>• <i>1</i> primary identifier;</li> <li>• <i>2</i> secondary identifier...</li> </ul>
<i>Order</i>	<p>whether this variable represents a natural order. (0: the variable does not represent a natural order; 1:the variable represents a natural order). If the value is set at 1, the variable is used in SQL expressions in an "order by " condition.</p> <p>There must be at least one variable set as Order in the Event data source.</p> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px;"> <p><b>⚠ Caution</b></p> <p>If the data source is a file and the variable stated as a natural order is not actually ordered, an error message will be displayed before model checking or model generation.</p> </div>
<i>Missing</i>	<p>the string used in the data description file to represent missing values (e.g. "999" or "#Empty" - without the quotes)</p>
<i>Group</i>	<p>the name of the group to which the variable belongs. Variables of a same group convey a same information and thus are not crossed when the model has an order of complexity over 1 . This parameter will be usable in future version.</p>
<i>Description</i>	<p>an additional description label for the variable</p>
<i>Structure</i>	<p>this option allows you to define your own variable structure, which means to define the variables categories grouping.</p>

### 5.3.1.2.1 Viewing the Data

To help you validate the description when using the Analyze option, you can display the first hundred lines of your dataset.

1. Click the button [View Data](#). A new window opens displaying the dataset top lines:

Data Set: Census01.csv

	age	workclass	fnlwgt	education	education-n...	marital-status	occu
1	39	State-gov	77516	Bachelors	13	Never-married	Adm-c
2	50	Self-emp-no...	83311	Bachelors	13	Married-civ-...	Exec-r
3	38	Private	215646	HS-grad	9	Divorced	Handle
4	53	Private	234721	11th	7	Married-civ-...	Handle
5	28	Private	338409	Bachelors	13	Married-civ-...	Prof-sj
6	37	Private	284582	Masters	14	Married-civ-...	Exec-r
7	49	Private	160187	9th	5	Married-spo...	Other:
8	52	Self-emp-no...	209642	HS-grad	9	Married-civ-...	Exec-r
9	41	Private	45781	Masters	14	Never-married	Prof-sj
10	42	Private	159449	Bachelors	13	Married-civ-...	Exec-r
11	37	Private	280464	Some-college	10	Married-civ-...	Exec-r
12	30	State-gov	141297	Bachelors	13	Married-civ-...	Prof-sj
13	23	Private	122272	Bachelors	13	Never-married	Adm-c
14	32	Private	205019	Assoc-acdm	12	Never-married	Sales
15	40	Private	121772	Assoc-voc	11	Married-civ-...	Craft-t
16	34	Private	245487	7th-8th	4	Married-civ-...	Transp
17	25	Self-emp-no...	176756	HS-grad	9	Never-married	Farmir
18	32	Private	186824	HS-grad	9	Never-married	Machir
19	38	Private	28887	11th	7	Married-civ-...	Sales
20	43	Self-emp-no...	292175	Masters	14	Divorced	Exec-r
21	40	Private	193524	Doctorate	16	Married-civ-...	Prof-sj
22	54	Private	302146	HS-grad	9	Separated	Other:
23	35	Federal-gov	76845	9th	5	Married-civ-...	Farmir
24	43	Private	117037	11th	7	Married-civ-...	Transp
25	59	Private	109015	HS-grad	9	Divorced	Tech-s

First Row Index: 1 Last Row Index: 100

2. In the field *First Row Index*, enter the number of the first row you want to display.
3. In the field *Last Row Index*, enter the number of the last row you want to display.
4. Click the *Refresh* button to see the selected rows.

### 5.3.1.2.2 Describing the Data

For Sequence Coding to be able to join the Reference and Transaction datasets, the Reference dataset to be analyzed must contain a single variable that serves as a unique key variable.

To Specify that a Variable is a Key:

1. In the *Key* column, click the box corresponding to the row of the key variable.
2. Type in the value "1" to define this as a key variable.

**Description: Desc\_Census01.csv**

Index	Name	Storage	Value	Key	Order	Missing	Group	Descrip...	Structure
1	age	number	continuous	0	0				
2	workclass	string	nominal	0	0	?			
3	fnlwgt	number	continuous	0	0				
4	education	string	nominal	0	0				
5	education-num	number	ordinal	0	0				
6	marital-status	string	nominal	0	0				
7	occupation	string	nominal	0	0	?			
8	relationship	string	nominal	0	0				
9	race	string	nominal	0	0				
10	sex	string	nominal	0	0				
11	capital-gain	number	continuous	0	0	99999			
12	capital-loss	number	continuous	0	0				
13	hours-per-week	number	continuous	0	0				
14	native-country	string	nominal	0	0	?			
15	class	number	nominal	0	0				
16	KxIndex	integer	continuous	1	0			Automa...	

Add Filter in Data Set

Analyze Open Description Save Description View Data

For this Scenario, use the file `session_purchase_desc.csv` as the description file.

To Describe the Data:

1. On the screen *Data Description*, click the button *Open Description*.

The following window opens:

**Guessed Description**

Index	Name	Storage	Value	Key	Order	Missing	Group	Description	Structure
1	SessionID	integer	continuous	0	0				
2	Purchase	integer	nominal	0	0				
3	KxIndex								

Add Filter in Data Set

**Load a Description for session\_purchase.csv**

Data Type: Text Files

Folder: ../../Samples/KSC Browse

Description: session\_purchase\_desc.csv Browse

OK Cancel

2. In the window *Load a Description*, select the type of your description file.
3. In the *Folder* field, select the folder where the description file is located with the *Browse* button.

### i Note

The folder selected by default is the same as the one you selected on the screen *Data to be Modeled*.

4. In the *Description* field, select the file containing the dataset description with the *Browse* button.
5. Click the *OK* button. The window *Load a Description* closes and the description is displayed on the screen *Data Description*.
6. Click the *Next* button.



### 5.3.1.3 Selecting Events Data

The screen Events Data lets you specify the data source to be used as the Transaction dataset.

For this Scenario:

- The *Folder* field should already be filled in with the name of the data source that you specified on the *Data to be Modeled* screen.
  - Select the file `file_view.csv`.
1. Select the format of your data source (Text Files, ODBC, ...).
  2. In the *Folder* field, specify the folder where your data source is stored.
  3. In the *Events* field, specify the name of your data source.
  4. Click the *Next* button.

### 5.3.1.4 Describing Events Data

The screen *Events Data Description* lets you describe your Transaction data, offering you the same options as the screen *Data Description*.

For sequence coding to function properly, there must be a variable in the Transaction dataset that is the same as the primary key declared for the Reference dataset, referred to as a “Join Column”. The name of the variable can be different, but the storage and value must be the same. The values of this variable need not be unique, since each Reference key can have 0, 1, or several associated transactions.

In addition to a suitable join column, the Transaction dataset must have at least one datetime variable. The datetime variable will be used by sequence coding to order the transactions.

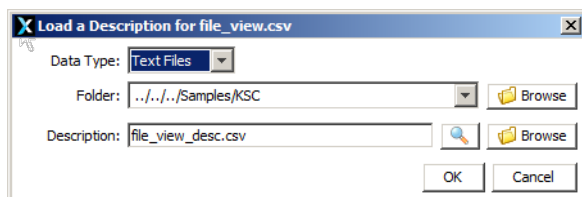
One of the datetime variables must absolutely be ordered and declared as such by setting to 1 the Order column for this variable in the description file.

When the data source comes from a database, Automated Analytics uses a query with an order by on the variable set as *Order* to retrieve the data. But when the data source is a file (.txt, .csv, ...), Automated Analytics verifies if the variable set as Order is actually ordered in the file, if not an error message is displayed.

For detailed procedures on how to set parameters on this screen, see [Describing the Data \[page 127\]](#).

For this Scenario, use the description file `file_view_desc.csv`.

1. On the screen *Event Data Description*, click the button *Open Description*.



The following window opens:

2. In the window *Load a Description*, select the file `file_view_desc.csv`.

3. Click the *OK* button. The window *Load a Description* closes and the description is displayed on the screen *Event Data Description*.

#### i Note

Note that the *Order* column is set at *1* for the *Time* variable, thus indicating that this variable is used as a natural order.

4. Click the *Next* button.

## 5.3.2 Step 2 - Defining the Modeling Parameters

### 5.3.2.1 Setting Sequence Coding Parameters

The screen *Sequence Analysis Parameters Settings* enables you to set some sequence coding parameters by performing the following tasks:

- Join your reference data with your transaction data
- Calculate the intermediate sequences
- Filter your events by period

For this Scenario:

- Select the *SessionID* column as the join column for both the log and reference datasets.
  - Select *Time* as the *Log Date Column*.
  - In the advanced parameters, keep *75%* of the hits.
  - Select *Infinite* as the *Time Window*.
1. On the screen *Sequence Analysis Parameters Settings*, select the join column for both the log and reference datasets.
  2. Select the *Log Date Column*.
  3. Click the *Advanced* button to set the advanced parameters.
  4. In the *Advanced* panel, slide the filter to *75%*.

#### 5.3.2.1.1 Understanding Data Manager - Sequence Coding Parameters

### Joining Your Data

To aggregate the reference data with the events data, you have to join both tables and indicate which column of each table corresponds to the reference ID.

In the fields *Columns for Join*, select the variables corresponding to the customer ID in both datasets. The information contained in both selected variables must be the same.

In the field *Log Date Column*, select the variable corresponding to the date and/or time of the log data.

## Calculating the Intermediate Sequences

The mode *Intermediate Sequences* provides you with additional information about the transitions and sequences existing in your datasets:

- order of the steps
- details of the steps
- continuity of the session for each step

## Filtering the Events by Time Window

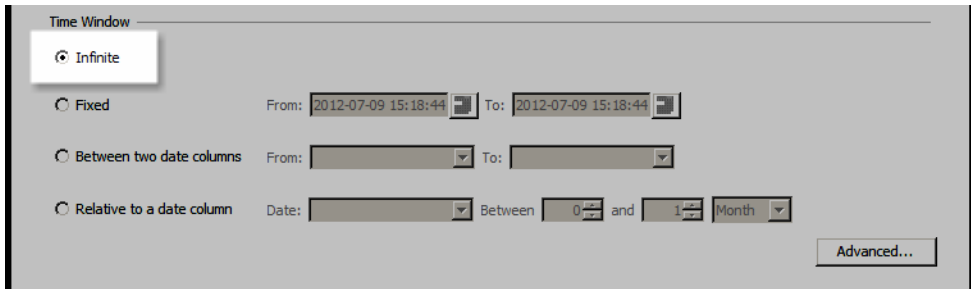
The section *Time Window* allows you to filter the events on which the model will be built by setting a period defined either by fixed dates or by values existing in the dataset. The following options are available to filter the events dataset:

Option	Description
<i>Infinite</i>	No time window is defined: all the events will be used.
<i>Fixed</i>	Only the events for which the <i>Log Date Column</i> value is between the two selected dates will be used.
<i>Between two date columns</i>	<p>Only the events for which the <i>Log Date Column</i> value is between the values of the two selected date columns will be used.</p> <p>For example, you can select the date columns corresponding to the beginning and the end of a trial period, dates that can be different for each customer.</p>
<i>Relative to a date column</i>	<p>Only the events for which the <i>Log Date Column</i> value fits in the range defined with respect to the selected date column will be used.</p> <p>For example, you can use the purchase date of a credit card as the reference and select all events that occurred in the three months leading to this date.</p>

### Caution

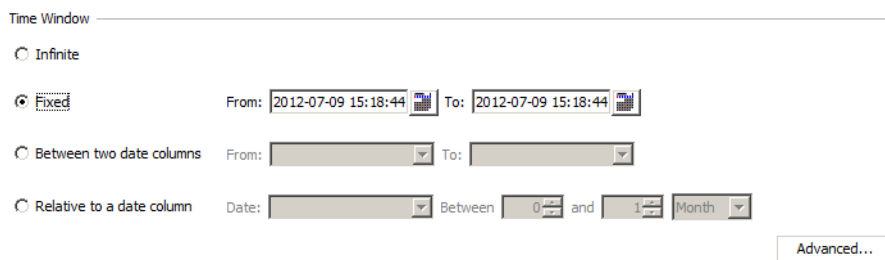
Be careful when choosing a period, the selected period must contain events existing in the dataset, or else you will obtain aberrant results for your model (negative KI, KR equal to 1, ...).

To Use All the Events, keep the Infinite option.



To Use Only the Events Occurring in a Fixed Time Window:

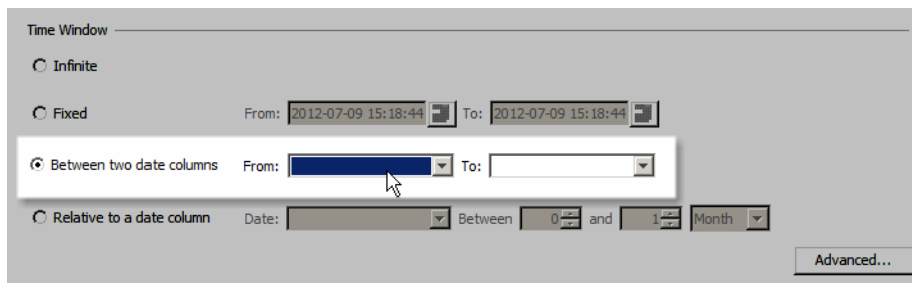
1. Check the *Fixed* option.



2. In the *From* field, select the date before which no events should be used.
3. In the *To* field, select the date after which no events should be used.

To Use Only the Events Occurring Between Two Date Columns:

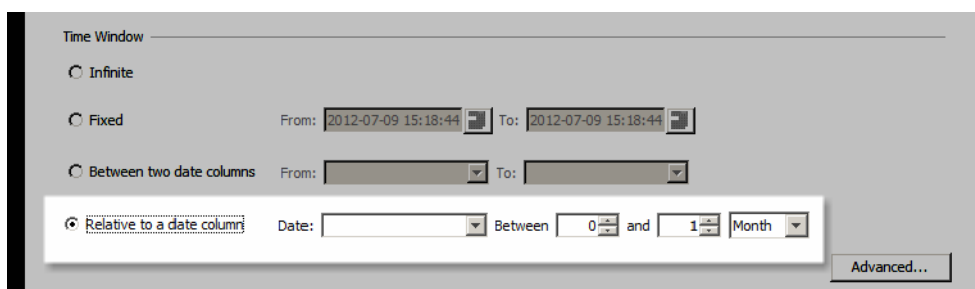
1. Check the option *Between two date columns*.



2. In the *From* field, select the date column containing the date before which no events should be used.
3. In the *To* field, select the date column containing the date after which no events should be used.

To Use Only the Events Occurring in a Range Relative to a Date Column:

1. Check the option *Relative to a date column*.



2. In the *Date* list, select the column that contains the date to use as a reference for the time window.

- In the *Between* field, enter the number of units that will indicate the start of the time window. The following table sums up the values you can use to define the beginning of the time window.

Value	Significance
<i>negative integer</i>	the time window begins before the reference date
0	the time window begins at the reference date
<i>positive integer</i>	the time window begins after the reference date

- In the *and* field, enter the number of units that will indicate the end of the time window.
- In the last drop-down list, enter the unit to be used to define the time window. For example, if you have set the parameters *Date CardPurchaseDate* Between -3 and 0 Month, only events occurring in the three months leading to the date of purchase will be kept for each customer.

### 5.3.2.1.2 Understanding Advanced Parameters

The advanced parameters allow you to configure the following elements:

- the prefix to be added to sequence coding generated variables,
- the location where the temporary files generated by the modeling are stored,
- the amount of information that will be kept for the modeling.

#### Sequence Coding Generated Variable Prefix

You can define a specific prefix that will be used to identify variables created by Data Manager. By default, this prefix is set to `ksc`.

#### Storage Type

When creating a model, Sequence Coding generates large quantities of temporary columns, you can select whether the data generated will be stored in a memory space or on a disk.

The option *In memory* is selected by default.

#### Filtering the Events

The *Filtering* option allows you to group rare categories into a single category labeled `KxOther`. It is very common for transaction logs to have many infrequently occurring categories that by themselves will not make reliable predictors. A predictive benefit can often be achieved by combining these rare categories into a single group. The Filtering slide allows you to select the categories to keep as separate columns based on percentage

of the overall transaction log. The categories corresponding to the remaining percentage of transactions are grouped in the KxOther column, which is automatically generated by Data Manager – Sequence Coding .

For example, if you set the Filtering slider at 90%, it means that the total number of transactions when adding all the categories assigned to separate columns must not exceed 90% of the total number of transactions. The categories that make up the remaining 10% of the transactions will be grouped under KxOther.

You can also define a threshold so that transitions which duration between two events is higher than the defined threshold will be ignored in the transition count.

### 5.3.2.1.2.1 Setting a Threshold

For the sample data, each row of the transaction log represents an HTML file requested by the visitor's browser. There are 10184 different files that are requested during the day. However, by positioning the Filtering slide at 75%, only 99 files are retained for separate count columns, and the rows with the remaining 10085 files are grouped into the KxOther count. This means that the 99 most common files make up 75% of the log and the remaining 10085 files make up only 25% of the log.

1. Check the box *Filter Transitions* greater than.
2. In the number field, enter the number of units defining the threshold.
3. In the drop-down list, select the unit to be used to define the threshold.

### 5.3.2.2 Selecting Sequence Coding Statistics

The screen *Sequence Analysis Variables Selection for Functions* lets you specify the type of statistics you want to calculate on transaction or event data.

For this Scenario, you decide to calculate for each session which pages have been visited on the web site. That way, you should be able to determine and understand which pages led the visitors to make a purchase.

You must use the following settings:

- For the variable Page, select the function Count, which will create a state column for each page visited.
1. The *Sequence Analysis Variables Selection for Functions* screen lists all the variables for which statistics can be calculated. For each variable listed, select the functions to use. You can choose among the three functions Count, CountTransition and FirstLast.
  2. Click the *Next* button.

#### 5.3.2.2.1 Operations Definition

Several standard sequence coding columns are created for each reference ID. For reference IDs that have no transactions associated with them, the standard sequence coding columns will have null values.

*KSC\_Start\_Date*: The timestamp of the first transaction in the log for each reference ID.

*KSC\_End\_Date*: The timestamp of the last transaction in the log for each reference ID.

*KSC\_TotalTime*: The seconds between the *KSC\_Start\_Date* and *KSC\_End\_Date*.

*KSC\_Number\_Events*: The number of transactions in the log associated with each reference ID.

In addition to the standard Data Manager – Sequence Coding columns, three types of operations are available:

- Count,
- Count the transitions,
- First and last.

## Count

When you select the *Count* option, sequence coding creates a new column for each value of the inserted variables.

*Count* encodes the sequences using one column per valid category in the specified nominal column. Each valid category is referred to as a “state”. Categories that are seen only once for the transactions associated with the reference id present in the Estimation dataset are discarded.

## CountTransition

When you select the *CountTransition* option, sequence coding creates a new column for each transition of categories in the selected dataset.

*CountTransition* encodes the sequences using one column per valid pair wise category transition in the specified column. Each valid category transition is referred to as a “state transition”. State transitions that are seen only once for the transactions associated with the reference id present in the Estimation dataset are discarded. A separate *KxOther* column will be created for rare transitions, using the threshold set by the Filter slider bar in the same way a *KxOther* column is created for the counts.

## FirstLast

The *FirstLast* option creates two columns, the categories of the selected variable from the first and last transactions in the log for each reference ID, called *FirstState* and *LastState* respectively. The *FirstState* and *LastState* columns are created automatically when either the *Count* or *CountTransition* options are selected.

### 5.3.2.3 Checking the Transactions

At this stage, the application analyses the datasets and creates a number of new variables, or columns. Depending on which operations you chose during the previous step, sequence coding creates:

- four standard columns - *ksc\_Start\_Date*, *ksc\_End\_Date*, *ksc\_TotalTime*, and *ksc\_Number\_Events*.

- one column for each state (if you have selected *Count*).
- one column for each transition (if you have selected *CountTransitions*).
- Two columns, *FirstState* and *FinalState* (if you have selected, *Count*, *CountTransitions*, or *FirstLast*).
- Six columns, *LastStepNumber*, *Last\_date-time*, *Last\_duration*, *Session\_Continue*, *LastState*, and *NextState* (if you have selected Intermediate Sequences).

For this Scenario, after the transactions are checked, sequence coding should have kept 99 state columns for the *Page* variable, plus the four standard columns and the *FirstState* and *LastState* columns.

1. During the model checking a progress bar is displayed.
2. When the process is over, click the button *Show Detailed Log*. The number of columns created by sequence coding is indicated.
3. Click the *Next* button.

### 5.3.2.4 Selecting Variables

Once the reference dataset, the events dataset and their descriptions have been entered, select the variables:

- one or more Targets Variables,
- possibly a Weight Variable,
- and the Explanatory Variables.

For this Scenario:

- Keep Purchase as the target.
- Use the `session_continue_skip.csv` file to select the variables to exclude. This list of variables includes the information that is not known about a session until a purchase has occurred or is very likely to occur. For this Web site, the checkout process included five order pages. The presence of any of the five order pages in the log indicates that they have already started the checkout process. The presence of `order5.tpl` indicates that a purchase has occurred. Since the goal of the analysis is to gain new insights into what behaviors lead to a purchase, these order pages and other similar information must be excluded from the analysis.

To select a Target Variable, on the screen *Selecting Variables*, in the section *Explanatory Variables Selected* (left hand side), select the variables you want to use as target variables.

To Exclude Explanatory Variables:

1. On the screen *Selecting Variables*, click the button *Open a Saved List* located under the section *Excluded Variables*.  
The window *Load Excluded Variables List* opens.
2. In the *Variables* field, select the file containing the variables to skip.
3. Click the *OK* button, the window closes. The list of excluded variables has been populated.

### 5.3.2.5 Setting the Number of Clusters

Before generating the model, you need to set the number of clusters you want to create.



For this Scenario, set the number of clusters to *10*, which is the default number.

In the panel *Summary of Modelling Parameters*, type the number of clusters you want to generate in the field *Find the best number of clusters in this range*.

## 5.3.3 Step 3 - Generating and Validating the Model

### 5.3.3.1 Generating the Model

Once the modeling parameters are defined, you can generate the model. Then you must validate its performance using the quality indicator *KI* and the robustness indicator *KR*:

- If the model is sufficiently powerful, you can analyze the responses that it provides in relation to your business issue.
- Otherwise, you can modify the modeling parameters in such a way that they are better suited to your dataset and your business issue, and then generate new, more powerful models.

On the screen *Summary of Modelling Parameters*, click the *Generate* button.

The screen *Training the Model* will appear.

The model is being generated.

A progress bar will allow you to follow the process.

### 5.3.3.2 Validating the Model

Once the model has been generated, you must verify its validity by examining the performance indicators:

- The quality indicator *KI* allows you to evaluate the explanatory power of the model, that is, its capacity to explain the target variable when applied to the training dataset. A perfect model would possess a *KI* equal to 1 and a completely random model would possess a *KI* equal to 0.
- The robustness indicator *KR* defines the degree of robustness of the model, that is, its capacity to achieve the same explanatory power when applied to a new dataset. In other words, the degree of robustness corresponds to the predictive power of the model applied to an application dataset.

For this Scenario, the model generated possesses:

- A quality indicator *KI* equal to 0.98,
- A robustness indicator *KR* equal to 0.99.

This means that Clustering found a reliable grouping (*KR* is greater than 0.90) that does a reasonable job of partitioning the purchasing visitors and the non-purchasing visitors (*KI* of 0.98). It is safe to look at the descriptive results of the segmentation to gain insight.

## 5.3.4 Step 4 - Analyzing and Understanding the Model

### 5.3.4.1 Segment Descriptions

On the screen [Cross Statistics](#), you can look at the logical definition and/or the cross statistics of each variable to gain an understanding of what kind of visitors belong to each cluster. Three clusters are particularly informative for your business problem, which is to determine which kind of population you should try to attract to increase your profit:

- the two clusters that have the highest conversion rates,
- the cluster that has the lowest conversion rate.

The chart below summarizes these clusters, and gives them each a label based on the cluster definition:

Freq.	Conv.	Definition	Label
1.9%	31.4%	<code>/shop/shipChart.html ]0;5]</code>	Shippers
3.5%	25.4%	<code>/welcome.html [1;20]</code>	Members
11.8%	0.1%	<code>/holiday/holiday-Sweeps.tmpl [1]</code>	Sweepstakers

The cluster Shippers is defined by sessions in which the shipping chart ([/shop/shipChart.htm](#)) has been seen between 1 and 5 times. Actually, this cluster does not give you much information. It just tells you that visitors that go to the shipping chart will probably make a purchase, which is rather logical. If you don't intend to buy, why would you look at the shipping information?

The cluster Members is more informative. It shows that people visiting the member home page ([welcome.html](#)) are more likely to buy. This is an interesting piece of information. It means that members are more likely to make a purchase than other visitors. So increasing the number of members should increase your profit.

The cluster Sweepstakers gives you information on a previous attempt at increasing the number of purchase through a sweepstake. You can see that only 0.1% of the people visiting the sweepstake page actually make a purchase. You can infer from this that your previous campaign had the effect opposite to the one expected.

## 5.4 Scenario 2: Predict End of Session Using Intermediate Sequences

1. In the main menu, select the option [Perform a Sequence Analysis](#) in the [Data Manager](#) section.

The screen [Add a Modeling Feature](#) is displayed.

2. Click on the option [Add a Classification / Regression](#).

### i Note

When building a model, you can either simply analyze the sequences or add extra transformations such as a Classification/Regression (Modeler - Regression/Classification) or a Clustering/Segmentation (Modeler - Segmentation/Clustering).

## 5.4.1 Step 1 - Selecting the Data

To know how to select and describe the data go to section [Selecting the Data and Describing the Data in Scenario 1](#).

1. Select the *Random partition strategy*.
2. Use the file `session_purchase.csv` as the reference file and use the file `session_purchase_desc.csv` as its description file.
3. Select the file `file_view.csv` and use the description file `file_view_desc.csv`.

## 5.4.2 Step 2 - Defining the Modeling Parameters

### Setting Sequence Coding Parameters

For this Scenario:

- Select the SessionID column as the join column for both the log and reference datasets.
- Select Time as the Log Date Column.
- Check the option Intermediate Sequences.
- In the advanced parameters, keep 75% of the hits.

#### i Note

To know how to set the parameters go to section [To Set the Parameters in scenario 1](#).

### Selecting Sequence Coding Statistics

In this scenario, you decide to calculate for each session which pages have been visited on the web site and what page led the net surfer to another. By adding page transactions count to the model, more information on the net surfers' behavior will appear.

You decide to calculate for each session which pages have been visited first and last on the web site and what pages had been visited in between. That way, you should be able to determine when a visitor is going to leave your web site and decide on which pages to make a \$5 reduction offer to keep the visitor and encourage him to make a purchase.

You must use the following settings.

For the variable *Page*, select the function *FirstLast*, which will create two states columns for each session, one containing the first page visited, the other the last page visited.

#### i Note

To know more about Sequence Coding Statistics, go to section [Selecting Date Manager - Sequence Coding Statistics](#) (see "Selecting Sequence Coding Statistics") in scenario 1.

## Checking the Transactions

For this scenario, after the transactions are checked, sequence coding should have kept 98 state columns for the Page variable.

## Selecting Variables

For this Scenario:

- Use the `session_continue_skip.csv` file to select the variables to exclude.
- Use `KSC_Session_continue` as the target and remove Purchase from the targets.

### i Note

To know how to select variables, go to section Selecting Variables (see "For this Scenario") in scenario 1.

## 5.4.3 Step 3 - Generating and Validating the Model

### Generating the Model

Once the modeling parameters are defined, you can generate the model. Then you must validate its performance using the quality indicator KI and the robustness indicator KR:

- If the model is sufficiently powerful, you can analyze the responses that it provides in relation to your business issue.
- Otherwise, you can modify the modeling parameters in such a way that they are better suited to your dataset and your business issue, and then generate new, more powerful models.

To generate the model, on the screen *Summary of Modelling Parameters*, click the *Generate* button. The screen *Training the Model* will appear. The model is being generated. A progress bar will allow you to follow the process.

### Validating the Model

Once the model has been generated, you must verify its validity by examining the performance indicators:

- The quality indicator KI allows you to evaluate the explanatory power of the model, that is, its capacity to explain the target variable when applied to the training dataset. A perfect model would possess a KI equal to 1 and a completely random model would possess a KI equal to 0.
- The robustness indicator KR defines the degree of robustness of the model, that is, its capacity to achieve the same explanatory power when applied to a new dataset. In other words, the degree of robustness corresponds to the predictive power of the model applied to an application dataset.

For this scenario, the model generated possesses:

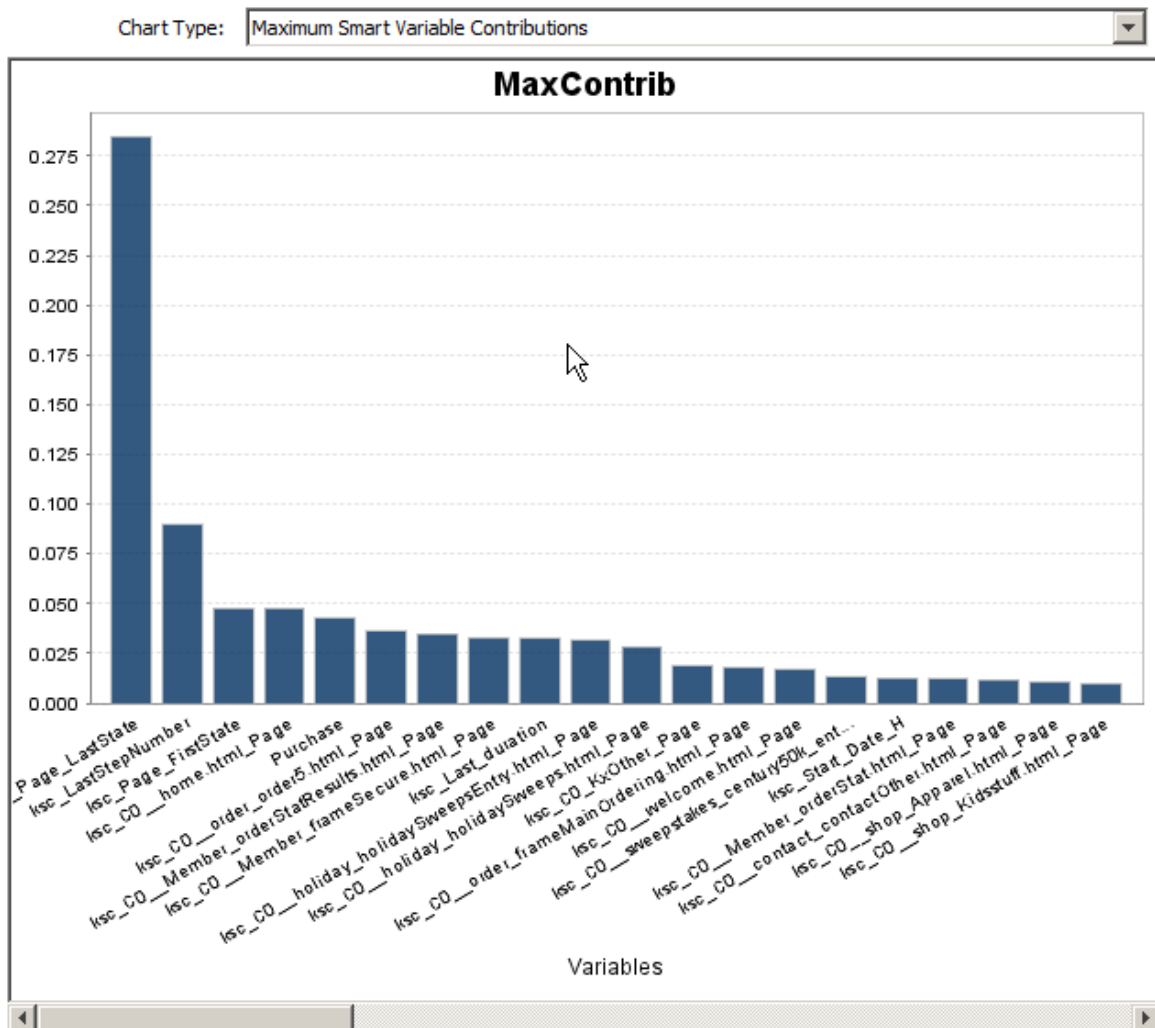
- A quality indicator *KI* equal to 0.70,
- A robustness indicator *KR* equal to 0.98.

This means that Classification/Regression found a robust model (*KR* is greater than 0.90) that does a reasonable job of predicting the end of a session (*KI* of 0.70). It is safe to look at the variables contributions to gain insight.

## 5.4.4 Step 4 - Analyzing and Understanding the Model

### Contributions by Variables

The following graph presents the variables contributions.



The pages having the more impact (positive or negative) on the buying act are listed in the following table.

Page viewed	This variable indicates...
<a href="#">KSC_Page_LastState</a>	the last page the internaut has viewed before ending his session
<a href="#">KSC_Last_duration</a>	duration of the session from the first page viewed to the previous state
<a href="#">KSC_LastStepNumber</a>	the number of pages the internaut has viewed before ending his session
<a href="#">Count_holidaySweepsEntry.html</a>	the number of time the page holidaySweepsEntry (access to holiday promotions) has been viewed

The impact of each page on the purchase is detailed in section Significance of Categories .

## Significance of Categories

### [KSC\\_Page\\_LastState](#)

This is by far the strongest predictor. This is similar to a low order Hidden Markov Model, where the current state is used to predict the next one.

### [Last\\_duration and LastStepNumber](#)

The length of the session and the number of pages viewed are also important. If the net surfer has viewed only one page, he has not yet entered the site and may end his session because the site may not seem of interest to him, but if he has viewed more than 12 pages, he has probably found what he was looking for and will end his session. If he has seen between 2 and 11 pages, he is probably shopping and thus should continue his session.

### [Count\\_holidaySweepsEntry.html](#)

If the page has been viewed it is a good indicator that the session will continue. Since this page is the entry point of a holiday promotion, the net surfer will at least go to the promotion page.

# 6 Text Analysis Scenario

## 6.1 About Text Analysis Scenario

### Who Should Read this section

This section is addressed to the business users who wish to perform tasks using predictive information about their customers or prospects through Automated Analytics powerful engine. There is no prerequisite for technical data mining knowledge.

### Prerequisites

Before reading this section, you should read the sections *Classification/Regression*, *Segmentation/Clustering* from the guide *Automated Analytics User Guides and Scenarios* that present respectively:

- An introduction to the Automated Analytics
- The essential concepts related to use of the Automated Analytics features

When following the scenario described in this user section, you will have to use data manipulation feature. No prior knowledge of SQL is required to use data manipulation -only knowledge about how to work with tables and columns accessed through ODBC sources. Furthermore, users must have “read” access on these ODBC sources. To use the Java graphical interface, users need write access on the tables KxAdmin and ConnectorsTable , which are used to store representations of data manipulations.

Please note that SAP HANA information views are not supported by Data Manager, only standard tables or views can be used as data source.

For more technical details regarding the Automated Analytics, please contact us. We will be happy to provide you with more technical information and documentation.

### What this section Covers

This section introduces you to the main functionalities of the Data Manager – text coding feature. Using the application scenario you can create your first models with confidence.

Data Manager – text coding lets you build predictive models from data containing textual fields. Thanks to text coding models, you can:

- Improve your models with textual processing.
- Handle some text mining problems such as text categorization or mail rerouting.
- Do automatic language recognition.

To know more about the basic concepts underpinning the Automated Analytics, read the *Automated Analytics User Guides and Scenarios*.

## Organization of this section

This document is subdivided into three chapters.

This chapter, About this document, serves as an introduction. This is where you will find information pertaining to the reading of this section, and information that will allow you to contact us.

The Chapter 2, General Introduction to Scenario, provides a summary to the text coding application scenario. It also introduces the user interface and the data files used in this scenario.

The Chapter 3, Standard Modeling with Text Coding, presents the Data Manager – text coding feature. It describes how to create five different predictive models, by adding data to the original dataset, and by using only Classification/Regression for the first two models, and then text coding combined with Classification/Regression for the last three models. You will then be able to compare the results obtained with each model.

If you want more information on Automated Analytics and on the essential concepts of modeling data, read the *Automated Analytics User Guides and Scenarios* guide available on the SAP Help Portal at <http://help.sap.com/pa>

## 6.1.1 Before Beginning

Files and Documentation Provided with this Guide.

### Sample Data Files

Both the evaluation version and the registered version of SAP Predictive Analytics are supplied with sample data files. These files allow you to take your first steps using various features of the application, and evaluate them.

During installation of SAP Predictive Analytics, the following sample files for text coding are saved under the folder `Samples\KTC`:

- `dmc2006.txt`
- `desc_dmc2006_without_textual.txt`
- `dmc2006_enriched.txt`
- `desc_dmc2006_enriched_no_textual.txt`
- `desc_dmc2006_enriched_textual.txt`

To obtain a detailed description of these files, see "Introduction to Sample Files".

The folder `Samples\KTC` is located:

- for Windows , in the folder `\Samples\KTC`, located in the installation directory.



for UNIX, in the folder `Samples\KTC` located in the folder where you have decompressed the installation archive file (that is `.tar.Z` or `.tar.gz`).

## Supported Languages Files

The text coding feature comes packaged with rules for several languages and can be easily extended to other languages.

The pre-packaged that comes with the installation includes:

- Dutch (du),
- English (en),
- French (fr),
- German (de),
- Japanese (jp)
- Spanish (sp),
- and Italian (it).

The folder `Resources\KTCData` is located:

- for Windows, in the folder `Desktop\Automated\Resources\KTCData`, located in the installation directory.

for UNIX, in the folder `Resources\KTCData` located in the folder where you have decompressed the installation archive file (that is `.tar.Z` or `.tar.gz`).

## Documentation

### Full Documentation

Complete documentation is available on the <http://help.sap.com/pa/>

This documentation covers:

- The operational use of Automated Analytics features,
- The architecture and integration of Automated Analytics API,
- The Java graphical user interface: SAP Predictive Analytics.

### Contextual Help

Each screen in SAP Predictive Analytics is accompanied by contextual help that describes the options presented to you, and the concepts required for their application.

To display the contextual help, in the *Help* menu, select *Help* or press *F1* on your keyboard.

## 6.2 General Introduction to Scenario

### Scenario

This scenario demonstrates how to use the text coding feature for creating a standard model.

The file `dmc2006.txt` is the sample data file that you will use to follow the scenario described in this user guide. It is the contest file from the Data Mining Cup 2006 (<http://www.data-mining-cup.com/2006/wettbewerb/aufgabe/1165919250/>), which is a German eBay file containing auctions with full conformance with protection of data privacy. The data used in this scenario are online auctions from the category "Audio&Hi-Fi:MP3-Player:Apple iPod".

The purpose of this scenario is to predict for new auctions if the actual sales revenue is higher than the average sales revenue of the product category.

### Introduction to Sample Files

The application is provided with sample data files allowing you to evaluate the text coding feature and take your first steps in using it. The data, or variables, contained in the sample file `dmc2006.txt` are described in the following table.

Variable	Description	Example of Values
<i>auct_id</i>	ID number of auction	An index value
<i>Item_leaf_category_name</i>	Product category	A numerical value with two decimals
<i>Listing_title</i>	Title of auction	
<i>Listing_subtitle</i>	Subtitle of auction	
<i>Listing_start_date</i>	Start date of auction	A date in the format such as
<i>Listing_end_date</i>	End date of auction	
<i>Listing_durtn_days</i>	Duration of auction	Specification in days
<i>Listing_type_code</i>	Type of auction (normal auction, multi-auction, ...)	
<i>Feedback_score_at_listing_time</i>	Feedback score by the seller at listing time of auction	An integer value
<i>Start_price</i>	Start price (in EUR)	A numerical value with n decimals
<i>Buy_it_now_price</i>	Buy-it-now price (In EUR, for buy	A numerical value with n decimals
<i>Buy_it_now_listed_flag</i>	Auction listing with buy-it-now option	1 if the information is true
<i>Bold_fee_flag</i>	Auction listing with boldface	1 if the information is true
<i>Featured_fee_flag</i>	Auction listing as homepage top offer	1 if the information is true
<i>Category_featured_fee_flag</i>	Auction listing as category top offer	1 if the information is true

Variable	Description	Example of Values
<i>Gallery_fee_flag</i>	Auction listing with gallery image	1 if the information is true
<i>Gallery_featured_fee_flag</i>	Auction listing with gallery (just in gallery view)	1 if the information is true
<i>Ipix_featured_fee_flag</i>	Auction listing with ipix (Additional, xxl, pic.show, pack)	1 if the information is true
<i>Reserve_fee_flag</i>	Auction listing with reserve price	1 if the information is true
<i>Highlight_fee_flag</i>	Auction listing with background color (in list view)	1 if the information is true
<i>Schedule_fee_flag</i>	Auction listing with determination of start time	1 if the information is true
<i>Border_fee_flag</i>	Auction listing with border	1 if the information is true
<i>Qty_available_per_listing</i>	Quantity of offered articles for multi-auctions	An integer value
<i>Gms</i>	Achieved sales revenue (In EUR)	A numerical value with n decimals (for multi-auctions average price of sold articles)
<i>Category_avg_gms</i>	Average sales revenue (In EUR) of product category (item_leaf_category_name)	A numerical value with n decimals
<i>Gms_greater_avg</i>	0 if $gms \leq category\_avg\_gms$ 1 if $gms > category\_avg\_gms$	Target

- The `filedmc2006_enriched.txt` is an enriched version of the `dmc2006.txt` dataset. The Data Manipulation feature has been used to create new variables from the ones already existing in the original dataset.
- The file `desc_dmc2006_enriched_no_textual` is the description file corresponding to the data file `dmc2006_enriched.txt` with no variable declared as string textual.
- The file `desc_dmc2006_enriched_textual.txt` is the description file corresponding to the data file `dmc2006_enriched.txt` with the *listing\_title* variable declared as string textual.

## Introduction to SAP Predictive Analytics

To accomplish the scenario, use SAP Predictive Analytics Desktop or Client. Then, select the feature you want to work with and let yourself be guided through all stages of the modeling process.

To start SAP Predictive Analytics:

1. Select **Start** > **Programs** > **SAP Business Intelligence** > **SAP Predictive Analytics Desktop** > **SAP Predictive Analytics**.  
The SAP Predictive Analytics welcome page appears.
2. Select the feature related to the type of model you want to create in the *Modeler* section

## 6.3 Extracting Information from Textual Data

### 6.3.1 Simple Method: Using a Classification Model on the Data

#### Description

Using the Modeler - Regression/Classification feature, you will generate a predictive model in order to determine if the auction sales revenue is higher than the sales revenue of its category.

This model will be generated by using as is the data provided in your data base.

To start a Classification/Regression Model, on the start panel, select *Classification / Regression* in the *Modeler* Section.

#### Modeling Process

The Modeler - Regression/Classification feature allows you to create explanatory and predictive models.

The first step in the modeling process consists of defining the modeling parameters:

#	Step	Section
1	Select a data source to be used as training dataset	Selecting a Data Source"
2	Describe the dataset selected.	Describing the Data
3	Select the target variable, and possibly a weight variable.	Selecting the Target Variable and a Weight Variable
4	Select the explanatory variables.	Selecting Explanatory Variables

#### Summary of the Modeling Settings to Use

The table below summarizes the modeling settings that you must use for the simple method. It should be sufficient enough for users who are already familiar with SAP Predictive Analytics.

For detailed procedures and more information, see the following sections.

Task(s)	Screen	Settings
<ul style="list-style-type: none"><li>Specifying the Data Source</li><li>Selecting a Partition Strategy</li></ul>	<i>Data to be Modeled</i>	<ul style="list-style-type: none"><li>Select the option <i>Use a File or a Database</i> .</li><li>In the <i>Folder</i> field, select the folder <i>Samples/KTC/</i></li><li>In the <i>Dataset</i> field, select the file <i>dmc2006.txt</i> .</li><li>Partition strategy: <i>Random Without Test</i></li></ul>

Task(s)	Screen	Settings
Describing the Data	<i>Data Description</i>	<ul style="list-style-type: none"> <li>Use the <i>Analyze</i> button to obtain the data description.</li> </ul>
Selecting the Target Variable and a Weight Variable	<i>Selecting the Target Variable</i>	<ul style="list-style-type: none"> <li>Select <i>gms_greater_avg</i> as the target variable</li> <li>Do not select a weight variable</li> </ul>
Selecting Explanatory Variables	<i>Selecting Variables</i>	<ul style="list-style-type: none"> <li>Exclude the variable <i>gms</i> from the list of variables to be used for modeling</li> </ul>

### 6.3.1.1 Selecting a Data Source

After selecting the type of model that you want to generate, you must select the data source that you want to use as the training dataset.

For this Scenario, in the panel *Select a Data Source*, set the options to the values listed in the following table.

Option	Value
<i>Use a File or a Database Table / Use Data Manager</i>	<i>Use a File or a Database Table</i>
<i>Data Type</i>	<i>Text File</i>
<i>Folder</i>	<i>Samples/KTC/</i>
<i>Dataset</i>	<i>dmc2006.txt</i>

1. On the screen *Select a Data Source*, after selecting option *Use a File or a Database Table*, select the option *Text files* in *Data Type* to select the data source format to be used.
2. Click the *Browse* button. The *Data Source Selection* dialog opens. Double-click the *Samples* folder, then the *KTC* folder.

#### **i** Note

Depending on your environment, the *Samples* folder may or may not appear directly at the root of the list of folders. If you selected the default settings during the installation process, you will find the *Samples* folder located in the installation directory.

3. Select the file *dmc2006.txt*, then click OK.

The name of the file will appear in the *Dataset* field.

4. Click *Next*.

## 6.3.1.2 Describing the Data

### Why Describe the Data Selected?

In order for the application features to interpret and analyze your data, the data must be described. To put it another way, the description file must specify the nature of each variable, determining their:

- Storage format: number (*number*), integer (*integer*), character string (*string*), date and time (*datetime*) or date (*date*).

Type: *continuous*, *nominal*, *ordinal* or *textual* .

#### ⚠ Caution

When creating a text coding model , you need to define at least one variable as textual to be able to go to the next panel.

For more information about data description, see the Classification, Regression, Segmentation and Clustering Scenarios – Automated Analytics User Guide.

### How to Describe Selected Variables

To describe your data, you can:

- Either use an existing description file, that is, taken from your information system or saved from a previous use of Automated Analytics features,

Or create a description file using the *Analyze* option, available to you in SAP Predictive Analytics. In this case, it is important that you validate the description file obtained. You can save this file for later re-use. If you name the description file `KxDesc_<SourceFileName>`, it will be automatically loaded when clicking the *Analyze* button.

#### ⚠ Caution

The description file obtained using the Analyze option results from the analysis of the first 100 lines of the initial data file. In order to avoid all bias, we encourage you to mix up your dataset before performing this analysis.

Each variable is described by the fields detailed in the following table:

The Field...	Gives information on...
<i>Name</i>	<i>the</i> variable name (which cannot be modified)

The Field...	Gives information on...
<i>Storage</i>	<p>the type of values stored in this variable:</p> <ul style="list-style-type: none"> <li>• <i>Number</i>: the variable contains only "computable" numbers (be careful a telephone number, or an account number should not be considered numbers)</li> <li>• <i>String</i>: the variable contains character strings</li> <li>• <i>Datetime</i>: the variable contains date and time stamps</li> <li>• <i>Date</i>: the variable contains dates</li> </ul>
<i>Value</i>	<p>the value type of the variable:</p> <ul style="list-style-type: none"> <li>• <i>Continuous</i>: a numeric variable from which mean, variance, etc. can be computed</li> <li>• <i>Nominal</i>: categorical variable which is the only possible value for a string</li> <li>• <i>Ordinal</i>: discrete numeric variable where the relative order is important</li> <li>• <i>Textual</i>: textual variable containing phrases, sentences or complete texts</li> </ul>
<i>Key</i>	<p>whether this variable is the key variable or identifier for the record:</p> <ul style="list-style-type: none"> <li>• <i>0</i> the variable is not an identifier;</li> <li>• <i>1</i> primary identifier;</li> <li>• <i>2</i> secondary identifier...</li> </ul>
<i>Order</i>	<p>whether this variable represents a natural order.</p> <p>There must be at least one variable set as Order in the Event data source.</p> <p>Warning - If the data source is a file and the variable stated as a natural order is not actually ordered, an error message will be displayed before model checking or model generation.</p>
<i>Missing</i>	the string used in the data description file to represent missing values (e.g. "999" or "#Empty" - without the quotes)
<i>Group</i>	the name of the group to which the variable belongs
<i>Description</i>	an additional description label for the variable

### 6.3.1.2.1 Creating a Description File

For this Scenario, create the data description by clicking the *Analyze* button.

1. On the screen Data Description, click the Analyze button.

The data description will appear.

2. Check that the description obtained is correct.

3. Once the data description has been validated, you can:
  - Save it by clicking the [Save](#) button.
  - Click the [Next](#) button to go to the following step.

The screen [Selecting the Target Variable](#) will appear.

4. Go to the section [Selecting a Target Variable](#).






### 6.3.1.2.2 A Comment about Database Keys

For data and performance management purposes, the dataset to be analyzed must contain a variable that serves as a key variable. Two cases should be considered:

- If the initial dataset does not contain a key variable, a variable index KxIndex is automatically generated by text coding. This will correspond to the row number of the processed data.
- If the file contains one or more key variables, they are not recognized automatically. You must specify them manually in the data description. See the procedure [To Specify that a Variable is a Key](#). On the other hand, if your data is stored in a database, the key will be automatically recognized.

To Specify that a Variable is a Key:

1. In the [Key](#) column, click the box corresponding to the row of the key variable.
2. Type in the value "1" to define this as a key variable.

Index	Name	Storage	Value	Key	Order	Missing	Group	Description	Structure
1	auct_id	string	nominal	1	0				
2	item_leaf_category_name	string	nominal	0	0				
3	listing_title	string	textual	0	0				
4	listing_subtitle	string	nominal	0	0				
5	listing_start_date	date	continuous	0	0				

### 6.3.1.2.3 Selecting the Target Variable and a Weight Variable

For this Scenario:

- Select the variable gms\_greater\_avg as your target variable.
  - Do not select any weight variable.
1. On the screen [Selecting Variables](#), in the [Explanatory variables selected](#) section, located on the left hand side, select the variable you want to use as Target Variable.

#### **i** Note

On the screen [Selecting Variables](#), variables are presented in the same order as that in which they appear in the table of data. To sort them alphabetically, select the option [Alphabetic sort](#), presented beneath each of the variable list.

2. Click the button > located on the left of the Target Variables list on the upper right hand side. The variable moves to the screen section [Target Variables](#).



To remove variables from the *Target Variables list*, select the variables you want to remove and click the button <.

### 6.3.1.2.4 Selecting Explanatory Variables

By default, and with the exception of key variables such as KxIndex, all variables contained in your dataset are taken into consideration for generation of the model. You may exclude some of these variables.

For this Scenario:

- Exclude gms from the list of variables to be used for modeling the variables since this variable contains the actual amount the auction reached, it answers the question and so would provide a perfect model if used.
  - Retain all the other variables.
1. On the screen *Selecting Variables*, in the section *Explanatory Variables Selected* on the left hand side, select the variable to be excluded.

#### i Note

On the screen *Selecting Variables*, variables are presented in the same order as that in which they appear in the table of data. To sort them alphabetically, select the option *Alphabetic Sort*, presented beneath each of the two parts of the screen.

2. Click the button > located in the center of the screen. The variable moves to the *Excluded Variables list*. To remove variables from the *Excluded Variables list*, select the variables you want to remove and click the button <.
3. Click *Next*. The screen *Summary of the Modeling Parameters* is displayed.

## 6.3.1.3 Results

### Model Performance Indicators

Once the model has been generated, you must verify its validity by examining the performance indicators:

	Predictive power	Prediction confidence
<i>Simple Method</i>	0.4738	0.9775

The predictive power (KI) is a quality indicator that allows you to evaluate the explanatory power of the model, that is, its capacity to explain the target variable when applied to the training dataset. A perfect model would possess a predictive power equal to 1 and a completely random model would possess a predictive power equal to 0.

The prediction confidence (KR) defines the degree of robustness of the model, that is, its capacity to achieve the same explanatory power when applied to a new dataset. In other words, the degree of robustness corresponds to the predictive power of the model applied to an application dataset.

To see how the predictive power and the prediction confidence are calculated, see Predictive Power, Prediction Confidence, and Profit Curves in the document Classification, Regression, Segmentation, and Clustering Scenarios - Automated Analytics User Guide.

### **i** Note

Validation of the model is a critically important phase in the overall process of Data Mining. Always be sure to assign significant importance to the values obtained for the predictive power and the prediction confidence of a model.

## Presentation of the Automated Analytics User Menu

Once the model has been generated, click *Next*. The screen *Using the Model* is displayed.

The screen *Using the Model* presents the various options for using a model.

Section	Possible Actions
<i>Display</i>	Display the information relating to the model just generated or opened, referring to the model curve plots, contributions by variables, the various variables themselves, HTML statistical reports, table debriefing, as well as the model parameters.
<i>Run</i>	Apply the model just generated or opened to new data, to run simulations, and to refine the model by performing automatic selection of the explanatory variables to be taken into consideration.
<i>Save/Export</i>	Save the model, or generate the source code.

## Taking a Closer Look at the Model

From the screen *Using the Model*, you can display a suite of plotting tools that allow you to analyze and understand the model generated in details. The three most useful tools are described in the table below.

On the screen...	You can observe and analyze...
<i>Profit Curves</i>	The performance of the model with respect to a hypothetical perfect model and a random type of model
<i>Contributions by Variables</i>	The contribution of each of the explanatory variables with respect to the target variable
<i>Significance of Categories</i>	The significance of the various categories of each variable with respect to the target variable

On the screen *Contributions by Variables* (see below), you notice that among the variables that contribute the most to the explanation of the target variable is *listing\_end\_date*. From this result and the knowledge of how auctions work, you can infer that calendar time has an impact on the auctions results and so you may want to

detail this variable content into more informative elements such as the day of the week, the month, and so on. This leads you to the intermediate method.

## 6.3.2 Intermediate Method: Adding Information with the Data Manipulations

### Description

The result of the simple method has highlighted the fact that dates have an important role in the modeling. It seems logical for time information to have an impact on auctions such as the day of the week, the day of the month, the month of the year. You can assume that the results of auctions are better on week-ends or at the beginning of the months or better some months than others, etc...

To make the most of the date variables, you will create new variables, for example by separating the days of the week so that they can be used as input in the modeling.

Additionally, to make use of the two other most important variables, you will extract more information from [Start\\_price](#) and [Buy\\_it\\_now\\_price](#) by calculating the ratio between the starting price and the sales mean for the category and the ratio between the Buy-it-now price and the sales mean for the category.

### Modeling Process

The process of building a predictive model on a dataset containing added time data is approximately the same as the one you used for building the model on the original data.

The only additional step you have to perform is to create new columns for both variables [listing\\_start\\_date](#) and [listing\\_end\\_date](#) : one for each day of the week, one for the day of the month and one for the month of the year.

The modified dataset contains the following added columns:

- extracted from the original variable `listing_start_date` :
  - `listing_start_monday`
  - `listing_start_tuesday`
  - `listing_start_wednesday`
  - `listing_start_thursday`
  - `listing_start_friday`
  - `listing_start_saturday`
  - `listing_start_sunday`
  - `listing_start_dayofmonth`
  - `listing_start_monthofyear`
- extracted from the original variable `listing_end_date` :
  - `listing_end_monday`
  - `listing_end_tuesday`
  - `listing_end_wednesday`
  - `listing_end_thursday`

- listing\_end\_friday
- listing\_end\_saturday
- listing\_end\_sunday
- listing\_end\_dayofmonth
- listing\_end\_monthofyear

You will also create two new columns in which the ratios described in the previous section will be stored:

- *Start\_price\_div\_mean\_category*, which is the result of the division of *start\_price* by *category\_avg\_gms*.
- *Buy\_it\_now\_price\_div\_mean\_category*, which is the result of the division of *buy\_it\_now\_price* by *category\_avg\_gms*.

To create these columns, you can use the data manipulation feature. However to speed the process for this demonstration, the modified dataset is provided in the folder `Samples/KTC`. The data file that correspond to the original file with the data manipulation creation is `dmc2006_enriched.txt`.

## Summary of the Modeling Settings to Use

The table below summarizes the modeling settings that you must use for the intermediate method. Except for the additional columns created in the dataset, the other settings are similar to the ones used for the simple method.

For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

Task(s)	Screen	Settings
Specifying the Data Source	<a href="#">Data to be Modeled</a>	<ul style="list-style-type: none"> <li>● Select the option <i>Text Files</i> in <i>Data Type</i>.</li> <li>● In the <i>Folder</i> field, select the folder <code>Samples/KTC/</code></li> <li>● In the <i>Dataset</i> field, select the file <code>dmc2006_enriched.txt</code>.</li> </ul>
Describing the Data	<a href="#">Data Description</a>	<ul style="list-style-type: none"> <li>● Select <code>desc_dmc2006_enriched_no_textual.txt</code> as the description file.</li> </ul>
Selecting the Target Variable and a Weight Variable	<a href="#">Selecting the Target Variable</a>	<ul style="list-style-type: none"> <li>● Select <i>gms_greater_avg</i> as the target variable</li> <li>● Do not select a weight variable</li> </ul>
Selecting Explanatory Variables	<a href="#">Selecting Variables</a>	<ul style="list-style-type: none"> <li>● Exclude the variables <i>KxIndex</i> and <i>gms</i> from the list of variables to be used for modeling</li> </ul>

## Results

The Training the Model screen shows the predictive power and the prediction confidence obtained for the model generated with the additional columns added to the original dataset.

The table below compares these results with the ones obtained for the simple method.

	Predictive power	Prediction confidence
<a href="#">Simple Method</a>	0.4738	0.9775
<a href="#">Intermediate Method</a>	0.5472	0.9757

The created variables give a better model. Indeed the predictive power has increased from 0.46 to 0.54. You can notice that the prediction confidence has slightly decreased but it's still high enough to guarantee a robust model.

Adding data from already existing variables has led you to obtaining a model that has a better quality and a good robustness.

### Taking a Closer Look at the Model

On the screen [Statistical Reports](#) > [Model Performance](#) > [KI & KR](#), you will notice that the added variables have made a difference in the model since some appear among the variables with the higher individual predictive power (KI). The individual predictive power (KI) represents the capacity of a variable to predict the target if only this variable was available.

You can see that both variables [listing\\_start\\_monthofyear](#) and [listing\\_end\\_monthofyear](#) appear in the top ten variables. When looking at their categories importance, you will notice that the auctions happening in December, indicated as [12](#) on the [Category Significance graph](#), have a better chance to sale higher than the average. This can be explained by the fact that people buy more around Christmas than any other period of the year.

Another of the top variables that already appeared in the previous model is [listing\\_title](#). When looking at the variable categories, you can see that each category contains many varied textual elements.

You can infer that this variable contains information that has yet to be exploited. Since this variable is a string, the best way to extract this hidden information is to use text coding, which leads you to the advanced method.

## 6.3.3 Advanced Method: Using Text Coding to Extract Information from the Textual Variables

### Description

Although the intermediate method resulted in a model that was both accurate and robust, you still have textual data not yet exploited. Since Modeler - Regression/Classification is not designed to process such data, you will need to use a data encoding feature. That is where text coding comes into play! Text coding is a data encoding feature that allows building a representative vector of the textual entries; it splits texts in words unit and extracts roots from the dataset..

Text coding is automatically included when 'textual' attributes are declared.

## Modeling Process

Compared to using only Modeler - Regression/Classification as you did for the first two methods, using text coding means performing the two additional steps below:

- Setting the language parameters.
- Setting the dictionary and encoding parameters.

### Selecting the Type of Model to Create

1. On SAP Predictive Analytics starting page, select **Data Manager** > **Perform a Text Analysis** . The screen *Add a Modeling Feature* is displayed.
  - *Add a Classification/Regression* analyzes the textual data, generates the corresponding variables and builds a Classification/Regression model on it.
  - *Add a Clustering* analyzes the textual data, generates the corresponding variables and builds a Clustering model on it.
  - *Standalone Data Transformation* analyzes the textual data and generates the corresponding variables.

For this scenario, select Add a Classification/Regression.

### Summary of the Modeling Settings to Use

The table below summarizes the modeling settings you must use for the advanced method. Except for the Text Coding specific steps - which are grayed in the table below, the other settings are similar to the ones used for the intermediate method.

Text Coding steps are presented in details in the following sections.

For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

Task(s)	Screen	Settings
Specifying the Data Source	<i>Reference Dataset</i>	<ul style="list-style-type: none"> <li>• Select <i>Text Files</i> in the <i>Data Type</i> list.</li> <li>• In the <i>Folder</i> field, select the folder <code>Samples/KTC/</code></li> <li>• In the <i>Dataset</i> field, select the file <code>dmc2006_enriched.txt</code>.</li> </ul>
Describing the Data	<i>Data Description</i>	<ul style="list-style-type: none"> <li>• Select <code>desc_dmc2006_enriched_textual.txt</code> as the description file.</li> <li>• Check that <i>listing-title</i> is set as <i>textual</i> .</li> </ul>
Text Coding - Setting the Language Definition	<i>Text Coding Parameters Settings</i>	<ul style="list-style-type: none"> <li>• Keep the <i>Default Local Text Coding Repository</i> .</li> <li>• In the <i>Language Recognition Mode</i> section, select <i>User Defined Language</i>.</li> <li>• Select <i>ge</i> (German) in the combo box as the <i>User Defined Language</i></li> </ul>

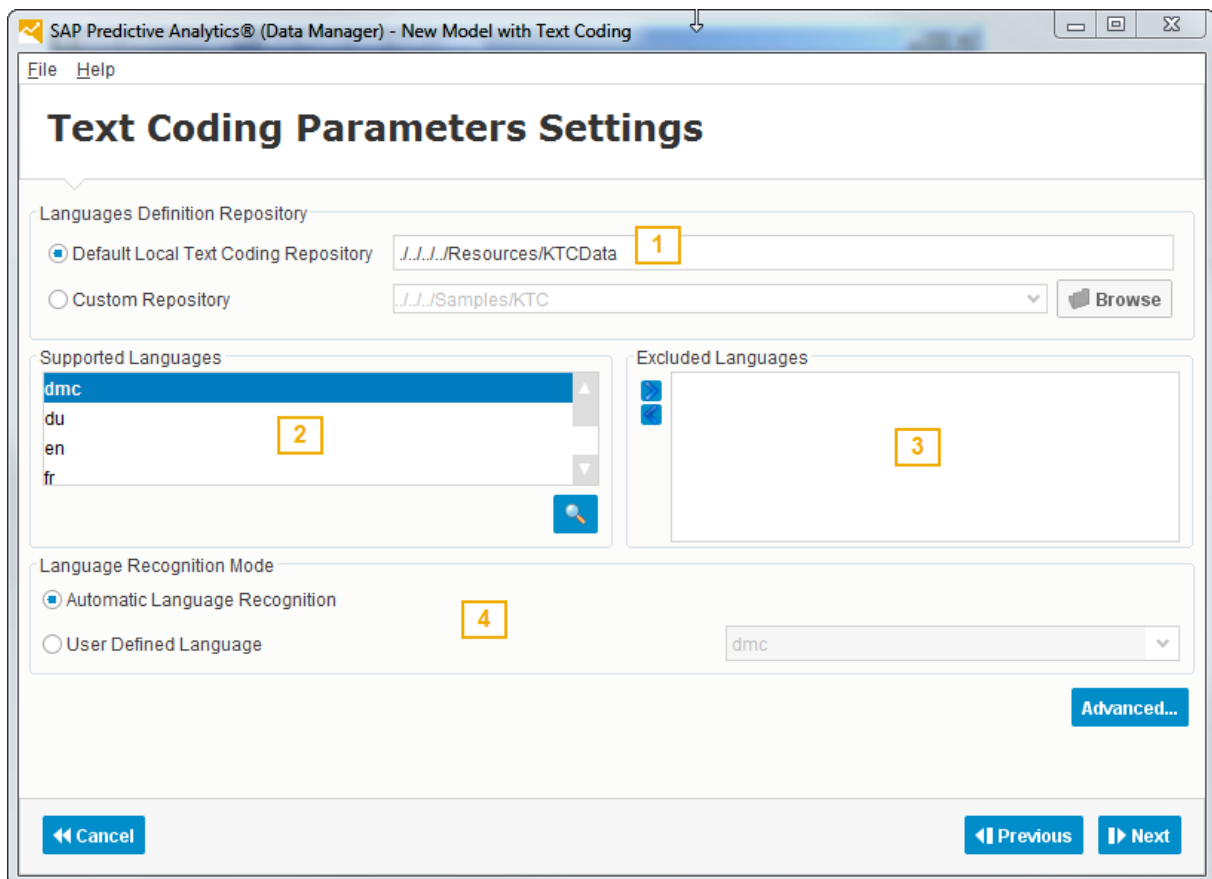
Task(s)	Screen	Settings
Text Coding - Setting the Dictionary and Encoding Parameters	<a href="#">Text Coding Parameters Settings (2)</a>	<ul style="list-style-type: none"> <li>Keep the <i>default</i> settings</li> </ul>
Selecting the Target Variable and a Weight Variable	<a href="#">Selecting Variables</a>	<ul style="list-style-type: none"> <li>Select <i>gms_greater_avg</i> as the target variable</li> <li>Do not select a weight variable</li> </ul>
Selecting Explanatory Variables	<a href="#">Selecting Variables</a>	<ul style="list-style-type: none"> <li>Exclude the variables <i>KxIndex</i> and <i>gms</i> from the list of variables to be used for modeling</li> </ul>

### 6.3.3.1 Setting Text Coding Parameters

#### Text Coding Languages Parameters

The first panel Text Coding Parameters Settings allows you to choose the language settings:

- Define the location of the Language Definition Repository (1),
- Select the list of Supported Languages (2),
- select the list of Excluded Languages (3),
- Select the Language Recognition Mode (4).



For this Scenario:

- Keep the default *Text Coding Language Definition Repository*.
- You can exclude the language named en (for English).
- Select the *User Defined Language* option for the *Language Recognition Mode*.
- If you did not exclude the English language, select *ge* (German) in the combo box as the *User Defined Language*.

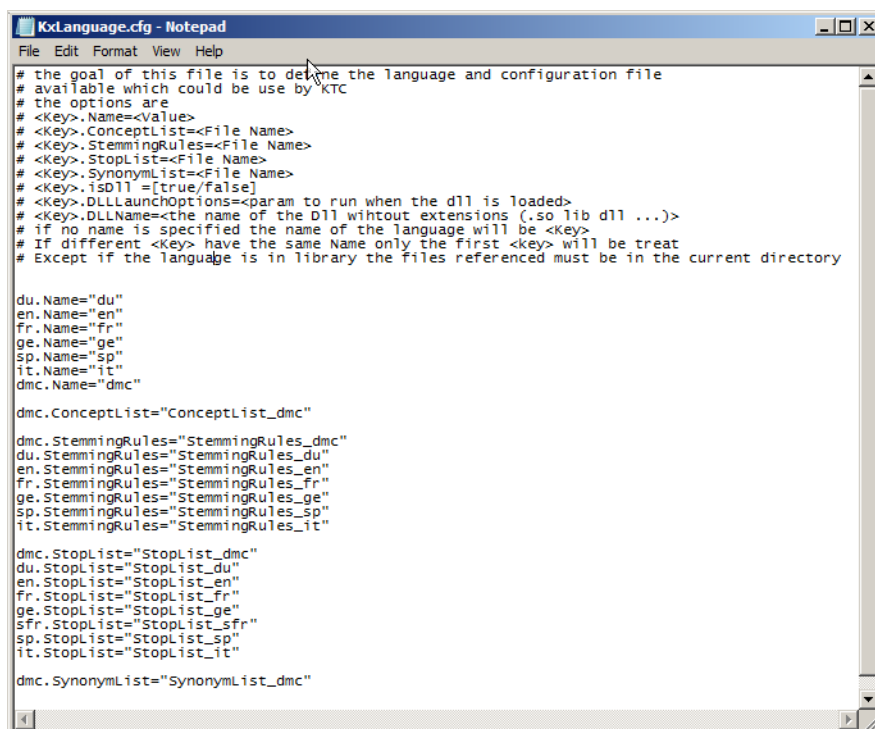
For Advanced Users:

You can create your own file to indicate the parameters on this panel. This file has to be named *KxLanguage.cfg* and needs to be structured as follows:

```
<Key>.Name="<Value>"
<Key>.ConceptList="<File Name>"
<Key>.StemmingRules="<File Name>"
<Key>.StopList="<File Name>"
<Key>.SynonymList="<File Name>"
```

To add comments, begin these lines with *#*. *<Key>* refers to the defined language.

The configuration file *KxLanguage.cfg* should look like the following one:



```
KxLanguage.cfg - Notepad
File Edit Format View Help
# the goal of this file is to define the language and configuration file
# available which could be use by KTC
# the options are
# <Key>.Name=<Value>
# <Key>.ConceptList=<File Name>
# <Key>.StemmingRules=<File Name>
# <Key>.StopList=<File Name>
# <Key>.SynonymList=<File Name>
# <Key>.isDll=[true/false]
# <Key>.DLLLaunchOptions=<param to run when the dll is loaded>
# <Key>.DLLName=<the name of the Dll wihtout extensions (.so lib dll ...)
# if no name is specified the name of the language will be <Key>
# If different <key> have the same Name only the first <key> will be treat
# Except if the language is in library the files referenced must be in the current directory

du.Name="du"
en.Name="en"
fr.Name="fr"
ge.Name="ge"
sp.Name="sp"
it.Name="it"
dmc.Name="dmc"

dmc.ConceptList="ConceptList_dmc"

dmc.StemmingRules="StemmingRules_dmc"
du.StemmingRules="StemmingRules_du"
en.StemmingRules="StemmingRules_en"
fr.StemmingRules="StemmingRules_fr"
ge.StemmingRules="StemmingRules_ge"
sp.StemmingRules="StemmingRules_sp"
it.StemmingRules="StemmingRules_it"

dmc.StopList="StopList_dmc"
du.StopList="StopList_du"
en.StopList="StopList_en"
fr.StopList="StopList_fr"
ge.StopList="StopList_ge"
sfr.StopList="StopList_sfr"
sp.StopList="StopList_sp"
it.StopList="StopList_it"

dmc.SynonymList="SynonymList_dmc"
```

### i Note

- The *<Key>* or language name has to be entered in the configuration file *KxLanguage.cfg* to be taken into account in the interface. If not set up there, the language will not appear in the interface.
- If no *<Key>* or language name is specified, the name of the language will be *<Key>*. If different *<Key>* have the same name, only the first *<Key>* will be treated. The referenced files have to be in the current directory.



## Text Coding Dictionary and Encoding Parameters

The second panel *Text Coding Parameters Settings* allows you set the construction parameters for the dictionary and the encoding parameters.

### Dictionary Construction Parameters

The dictionary is made of roots, that is, meaningful words or terms. You can set the following parameters of the dictionary construction:

- **Stop Words Removing:** when this option is checked, the stop words are removed from the list of roots
- **Stemming Reduction:** when this option is checked, the affixes are removed to limit the number of roots.
- **Concept Merging:** this option allows you to use an external file associating terms (that is groups of words designating a single concept, such as “the White House” or “credit card”) with concepts. Because it treats groups of words, this option is applied before the removal of the stop words and the stemming. You can create your own concepts dictionary by creating a text file named `ConceptList_<LanguageCode>` (without extension), which contains on each line a group of words associated with the corresponding concept. For example, you can create a concept list for an airline company:

```
word=concept
business-class=BusinessClass
first-class=FirstClass
flying-blue=FlyingBlue
```

Or you can apply the concept of “creditcard” to any credit card (such as “American Express”, “Visa Card”, ...):

```
credit-card=creditcard
american-express=creditcard
visa-card=creditcard
mastercard=creditcard
```

#### i Note

- you have to put a “=” sign of equality between the words and the concepts, to replace the blanks (or every other separator) by dashes and to write the words in lower-case letters (since the concept merging is applied after the removal of all upper-case letters).
- you have to do the concept merging for the singular and plural forms of the words to cover all the occurrences.

The use of the concept list being language dependent, the appropriate list is automatically selected once the language has been either automatically identified, or selected by the user.

- **Synonyms Replacement:** this option allows you to use an external file defining synonymic roots. It will be used to replace some roots by a root selected by the user. This option is applied after the stop words have been removed and the stemming rules have been applied. You can create your own synonyms dictionary by creating a text file named `SynonymList_<LanguageCode>`, which contains on each line a root found by text coding associated with the synonym root as shown below:

#### ≡, Sample Code

```
<found_root>=<replacement_root>
```

The use of the synonyms list being language dependent, the appropriate list is automatically selected once the language has been either automatically identified, or selected by the user.

- *Maximum Generated Root Number*: this option allows you to select how many roots you want to keep in the dictionary. By default the roots with the highest frequencies are kept, but you can select a percentage of the most frequent roots to exclude by clicking the Advanced button.

### Encoding Parameters

Each root is converted into a variable and, when the root appears in a text, its presence can be encoded in three ways:

- *Boolean*: the presence of the word is encoded 1 and its absence is encoded 0.
- *Term Frequency*: the number of apparitions of the root in the current text.
- *TF-Inverse Document Frequency*: a measure of the general importance of a root in the current document relative to the whole set of documents based on Term Frequency.

$$TFIDF = TF * \log_{10} (\text{TotalNumberOfDocuments} / \text{NumberOfDocumentsContainingTheRoot})$$

- *Term Count*: the number of times the root appears in the current text.
- *TC-Inverse Document Frequency*: a measure of the general importance of a root in the current document relative to the whole set of documents based on Term Count.

$$TCIDF = TC * \log_{10} (\text{TotalNumberOfDocuments} / \text{NumberOfDocumentsContainingTheRoot})$$

For this Scenario:

- Keep the default parameters.
- Click *Next*, the panel *Text Coding Model Learning* is displayed.

This panel lists the roots identified by text coding in the analyzed textual variable, listing\_title in this scenario, with their respective frequency of apparition in the dataset. It allows you to identify the most frequent roots and to decide if these roots are really meaningful for your problem or not.

## 6.3.3.2 Results

The table below compares the results obtained with the model generated with text coding with the ones obtained for the first two methods.

	<i>Predictive power (KI)</i>	<i>Prediction confidence (KR)</i>
<i>Simple Method</i>	0.4738	0.9775
<i>Intermediate Method</i>	0.5472	0.9757
<i>Advanced Method</i>	0.6813	0.9618

The analysis of textual variables gives a better model. Indeed the predictive power has increased from 0.55 to 0.68.

Using text coding has led you to obtaining a model with a better quality and a high robustness.

### Taking a Closer Look at the Model

On the screen *Contributions by Variables*, you'll notice that the variables that have been created by the text coding engine are important in the final model. For example <tc\_listing\_title\_2gb> is the best maximum

smart variable contribution. From this debriefing, you can see that 25 variables are displayed, 14 of which have been generated by the text coding engine.

However after studying the roots listed in the panel text *Coding Model Learning*, you can see that some of them are similar and should probably be merged. For example, both variables `<tc_listing_title_2gb>` and `<tc_listing_title_2 >` exist and yet they contain the same information.

When building a model, the text coding engine automatically generates two variables:

- `<tc_<variable name>_EffectiveRoot>`: this variable counts the final number of roots in the *csReferer textual* field.
- `<tc_<variable name>_CountInformation>`: this variable counts the number of roots before filtering.

## 6.3.4 Advanced Method without Stop Words and Stemming Rules

### Description

In the results of the model using text coding, you can see that the variables created by text coding have brought information in the final model. For example `tc_listing_title_2gb` is the most contributive variable. You have seen that some of these variables contain the same information and should be grouped. However before grouping similar terms, you have to measure the impact of the German processing on the dataset. To that effect, you will build a text coding model without specific German processing in order to see what its impact on the model quality is.

### Modeling Process

The process of using text coding without German specific processing is approximately the same as the one you used for building the previous model. You will only need to change the dictionary and encoding parameters.

### Summary of the Modeling Settings to Use

The table below summarizes the modeling settings you must use for the advanced method.

The text coding specific steps are grayed in the table below and the step different from the previous model is indicated in blue. The other settings are similar to the ones used for the advanced method. Text coding steps are presented in details in the following sections.

For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

Task(s)	Screen	Settings
Specifying the Data Source	<a href="#">Data to be Modeled</a>	<ul style="list-style-type: none"> <li>Select the option <i>Text Files</i> in <i>Data Type</i>.</li> <li>In the Folder field, select the folder <code>Samples/KTC/</code>.</li> <li>In the <i>Dataset</i> field, select the file <code>dmc2006_enriched.txt</code>.</li> </ul>
Describing the Data	<a href="#">Data Description</a>	<ul style="list-style-type: none"> <li>Select <code>desc_dmc2006_enriched_textual.txt</code> as the description file.</li> <li>Check that listing-title is set as <i>textual</i>.</li> </ul>
Text Coding - Setting the Language Definition	<a href="#">Text Coding Parameters Settings</a>	<ul style="list-style-type: none"> <li>Keep <i>default Language Definition Repository</i>.</li> <li>Select the <i>User Defined Language</i> option as the <i>Language Recognition Mode</i>.</li> <li>Select <i>ge</i> (German) in the combo box as the <i>User Defined Language</i>.</li> </ul>
Text Coding - Setting the Dictionary and Encoding Parameters	<a href="#">Text Coding Parameters Settings (2)</a>	<ul style="list-style-type: none"> <li>Uncheck <i>Stop Word Removing</i>.</li> <li>Uncheck <i>Stemming Reduction</i>.</li> </ul>
Selecting the Target Variable and a Weight Variable	<a href="#">Selecting Variables</a>	<ul style="list-style-type: none"> <li>Select <code>&lt;gms_greater_avg&gt;</code> as the target variable</li> <li>Do not select a weight variable</li> </ul>
Selecting Explanatory Variables	<a href="#">Selecting Variables</a>	<ul style="list-style-type: none"> <li>Exclude the variables <code>&lt;KxIndex&gt;</code> and <code>&lt;gms&gt;</code> from the list of variables to be used for modeling</li> </ul>

## Results

The table below compares the results obtained without using stop words identification and stemming rules with the ones obtained for the other methods.

	<i>Predictive power (KI)</i>	<i>Prediction confidence (KR)</i>
<i>Simple Method</i>	0.4738	0.9775
<i>Intermediate Method</i>	0.5472	0.9757
<i>Advanced Method</i>	0.6813	0.9618
<i>Advanced Method without Stop Words and Stemming</i>	0.6788	0.9640

There is no significant evolution of the predictive power and the prediction confidence. So you can conclude that using German stop words and stemming rules does not really add anything to the model.

## 6.3.5 Adapted Method: Defining a Specific Language for the Domain

### Description

The results of disabling the German stop words and stemming rules show that they have no real impact on the model quality. Actually, after viewing the data, that makes sense. Indeed, the content of the `<listing title>` variable cannot be considered exactly as natural language but more as a language specific to a smaller domain.

So in this last method, you will define the stop words and stemming rules based on German but relevant to this domain only. This comes down to creating a specific language, which you will name `<dmc>`.

### Modeling Process

For this method you will have to create a list of stop words specific to the current domain and the stemming rules also adapted to this domain. The process is the same as the one you used for the advanced method; you will only need to set the language to the one you will create in the following steps. The two sections below describe how to create a stop words list and stemming rules. However since the process of creating the stop words list and the stemming rules can be lengthy, both are provided as an example in SAP Predictive Analytics. Thus the new language `dmc` will appear in the list of languages.

### How to Detect Stop Words

Stop words are words that bring no information because they are too frequent or on the contrary that are less frequent.

Typically stop words are link words such as `<aber, ob, ich, so, am, auf>`... in German.

However other words can also be defined as stop words. The panel *Text Coding Model Learning* obtained by the advanced method without stop words and stemming rules can give you insight on which words can be considered as stop words.

This panel lists for each textual variable:

- the identified roots in the Root column.
- the number of occurrences of each root in the Frequency column.

Look at the roots *ipod* and *apple* for example. When you compare their number of occurrences with the total number of records in the dataset, it appears that *ipod* is present 7500 times and *apple* 5228 times in a dataset that counts 8000 lines. It is evident that they are much too frequent to contain information.

Another way to detect stop words is to use individual variable contribution after the classification/regression process in order to see the variables that have no predictive power (KI).

The stop words list is stored in a text file named `StopList_<language_code>`. For example, the stop words list created for the specific language you are working on will be named `StopList_dmc`.

## How to Build Simple Stemming Rules

According to the words displayed in the panel [Text Coding Model Learning](#), you can build some simple stemming rules. Indeed the first thing that appears is that some words such as *20gb*, *20g* and *20* can be merged into a single words *20-GB*.

It can be defined by these stemming rules:

```
7 3 ^20gb$ nocond nocond ^20gb$ 20-GB 4
8 3 ^20$ nocond nocond ^20$ 20-GB 4
```

### i Note

See Regular Expression Reminder in the Annex.

The syntax of a stemming rule is the following:

Rule	Step	CondWord		
	CondR1	CondR2Match	Replace	StepAfter

The columns represent:

- *Rule* : the number of the rule
- *Step* : the step the rule belongs to
- *CondWord* : the condition applied to the word
- *CondR1* : the condition applied to the first region
- *CondR2* : the condition applied to the second region
- *Match* : the parts of the word to select for replacement
- *Replace* : the string to replace the matched part
- *StepAfter* : the step to go if the rule has been applied

```
So the stemming rule 7 3 ^20gb$ nocond nocond ^20gb$ 20-GB 4 says: "if the word is 20gb then replace 20gb by 20-GB and go to stemming rules of step 4 if they exist"
```

Another way to create stemming rules is to use the copy button in the panel [Text Coding Model Learning](#), then paste the information on excel and sort the data by alphabetical order. Then you can identify different forms of the same words. For example, three different occurrences of a simple word can be identified:

- eingeschweisst
- eingeschweist
- eingeschweißt

So you can create two stemming rules to manage this word:

```
85 3 ^eingeschweist$ nocond nocond ^eingeschweist$ eingeschweisst 4
86 3 ^ eingeschweißt$ nocond nocond ^eingeschweißt$ eingeschweisst 4
```

These rules will replace two of the identified forms by the third one, so that only one form remains.

Moreover in the file you can find color names in different languages, for example blau in German and blue in English. So you can create the associated stemming rules such as:

```
65 3 ^blue$ nocond nocond ^blue$ blau 4
```

You can also create stemming rules that merge words that often appear together in the dataset such as "original and packaging" which translate in German to:

- original
- verpackt

This can be managed by the following the stemming rules:

41	3	^original\$	nocond	nocond	^original\$	original_verpackt	4
42	3	^verpackt\$	nocond	nocond	^verpackt\$	original_verpackt	4

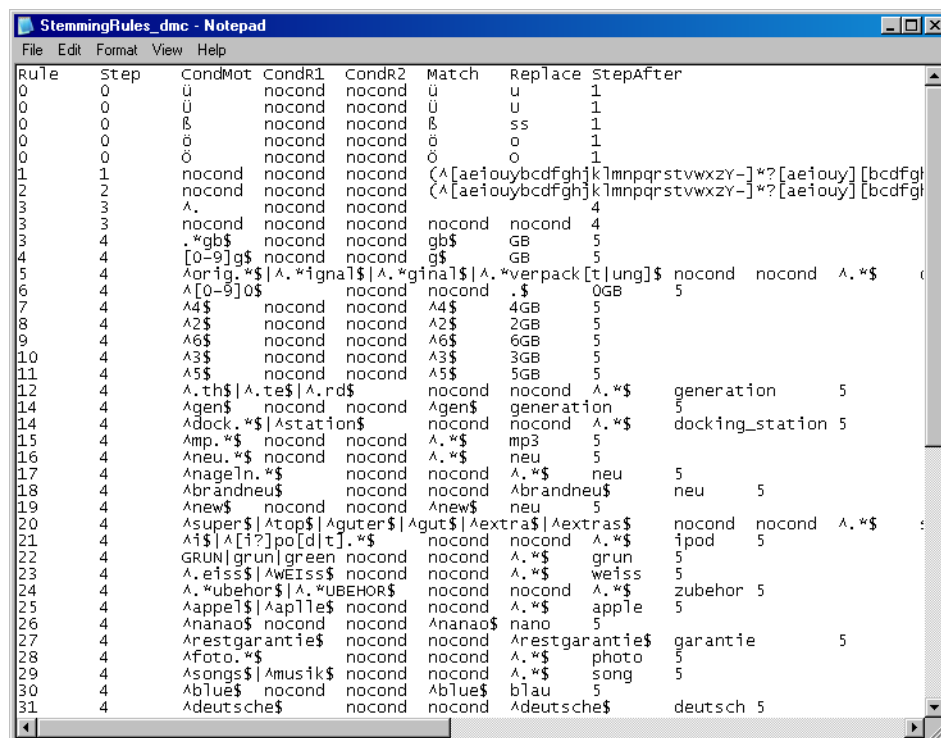
Lastly, you can merge correlated roots. In the previous model, look the [Variables Correlations](#) in [Statistical Report > Descriptive Statistics](#).

Index	First Variable	Second Variable	Coefficient
11	listing_end_date	listing_start_date	0.995
3	buy_it_now_price	buy_it_now_price_div_mean_c...	0.994
18	start_price	start_price_div_mean_category	0.995
14	listing_end_monthofyear	listing_start_monthofyear	0.868
12	listing_end_date	listing_start_monthofyear	0.863
15	listing_start_date	listing_start_monthofyear	0.852
20	tc_listing_title_mp3	tc_listing_title_player	0.843
13	listing_end_monthofyear	listing_start_date	0.808
10	listing_end_date	listing_end_monthofyear	0.805
17	listing_type_code	start_price_div_mean_category	0.778
16	listing_type_code	start_price	0.771
4	category_avg_gms	item_leaf_category_name	0.621
7	item_leaf_category_name	tc_listing_title_30	0.595
5	category_avg_gms	tc_listing_title_30	0.577
9	item_leaf_category_name	tc_listing_title_video	0.565
6	category_avg_gms	tc_listing_title_video	0.545
19	tc_listing_title_guter	tc_listing_title_zustand	0.530
8	item_leaf_category_name	tc_listing_title_30gb	0.525
1	buy_it_now_listed_flag	buy_it_now_price	-0.962
2	buy_it_now_listed_flag	buy_it_now_price_div_mean_c...	-0.98

You can see that the roots mp3 and player are highly correlated so you can create a stemming rule that will merge those roots into a single one.

43	3	^mp3\$	nocond	nocond	^mp3\$	mp3_player	4
44	3	^player\$	nocond	nocond	^player\$	mp3_player	4

The stemming rules are listed in a text file named `StemmingRules_<language_code>`. For example, the stemming rules created for the specific language you are working on will be sorted in the file `StemmingRules_dmc`. The stemming rules list for the dmc language should look like the following file:



## Summary of the Modeling Settings to Use

The table below summarizes the modeling settings you must use for the final method. The text coding specific steps are grayed in the table below and the steps different from the previous model are indicated in green. The other settings are similar to the ones used for the advanced method.

For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

Task(s)	Screen	Settings
Specifying the Data Source	<i>Data to be Modeled</i>	<ul style="list-style-type: none"> <li>Select the option <i>Text Files</i> in <i>Data Type</i>.</li> <li>In the <i>Folder</i> field, select the folder <code>Samples/KTC/</code></li> <li>In the <i>Dataset</i> field, select the file <code>dmc2006_enriched.txt</code>.</li> </ul>
Describing the Data	<i>Data Description</i>	<ul style="list-style-type: none"> <li>Select <code>desc_dmc2006_enriched_textual.txt</code> as the description file.</li> <li>Check that listing-title is set as <code>textual</code>.</li> </ul>



Task(s)	Screen	Settings
Text Coding - Setting the Language Definition	<a href="#">Text Coding Parameters Settings</a>	<ul style="list-style-type: none"> <li>Keep <i>default Language Definition Repository</i>.</li> <li>Select the <i>User Defined Language</i> option as the <i>Language Recognition Mode</i>.</li> <li>Select <i>dmc</i> in the combo box as the <i>User Defined Language</i>.</li> </ul>
Text Coding - Setting the Dictionary and Encoding Parameters	<a href="#">Text Coding Parameters Settings (2)</a>	<ul style="list-style-type: none"> <li>Check <i>Stop Word Removing</i>.</li> <li>Check <i>Stemming Reduction</i>.</li> <li>Check <i>Concept Merging</i>.</li> <li>Check <i>Synonym Replacement</i>.</li> </ul>
Selecting the Target Variable and a Weight Variable	<a href="#">Selecting Variables</a>	<ul style="list-style-type: none"> <li>Select <code>&lt;gms_greater_avg&gt;</code> as the target variable.</li> <li>Do not select a weight variable.</li> </ul>
Selecting Explanatory Variables	<a href="#">Selecting Variables</a>	<ul style="list-style-type: none"> <li>Exclude the variables <code>&lt;KxIndex&gt;</code> and <code>&lt;gms&gt;</code> from the list of variables to be used for modeling.</li> </ul>

## Results

The screen below shows the quality (KI) and robustness (KR) indicators obtained for the model generated with text coding adapted to the specific domain of application.

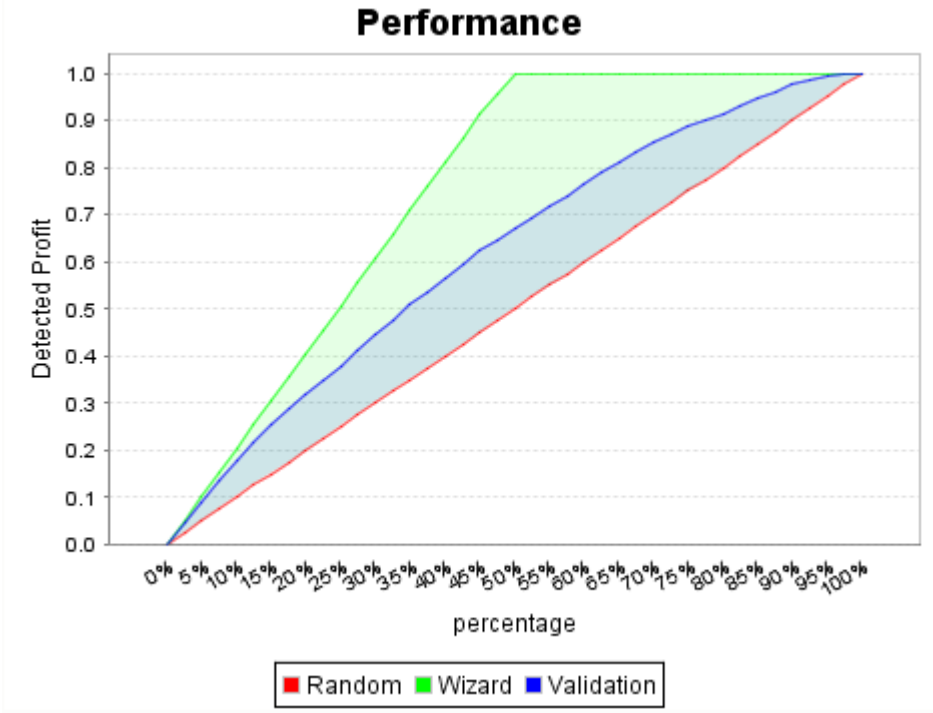
The table below compares these results with the ones obtained for the first two methods.

	Predictive power (KI)	Prediction confidence (KR)
<a href="#">Simple Method</a>	0.4738	0.9775
<a href="#">Intermediate Method</a>	0.5472	0.9757
<a href="#">Advanced Method</a>	0.6813	0.9618
<a href="#">Advanced Method without Stop Words and Stemming</a>	0.6788	0.9640
<a href="#">Adapted Method</a>	0.7133	0.9679

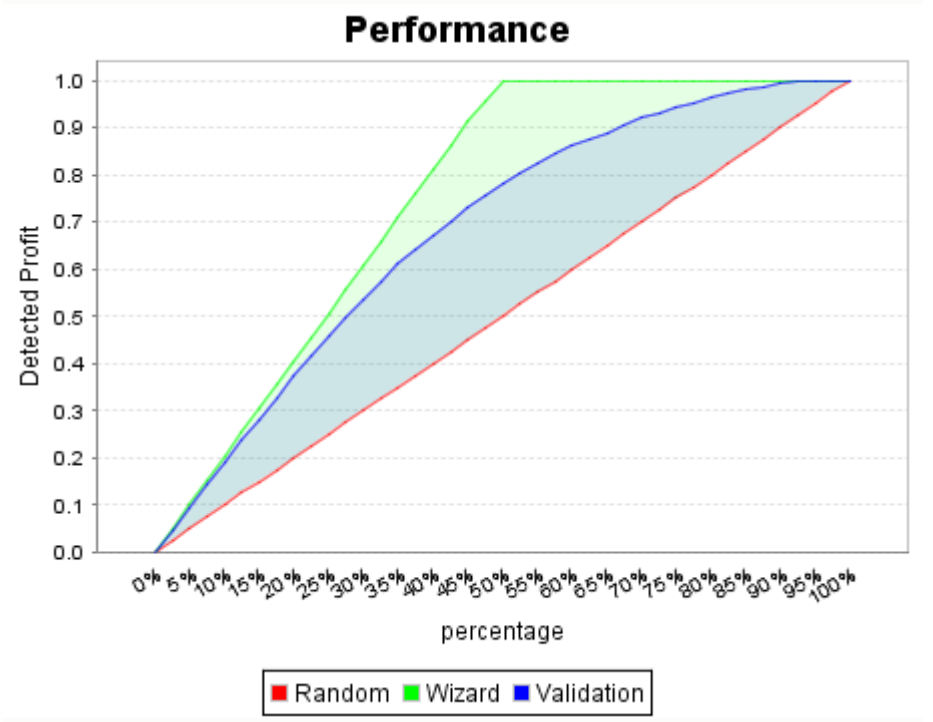
The predictive power has largely improved and the prediction confidence stays very high.

So you can see that from a simple classification/regression model performed on the original dataset to a text coding + classification/regression model on improved data with a specialty language defined you have gained a lot in model quality (+23% in predictive power) without losing model robustness.

The increased quality of the model is clearly apparent on the model graphs below when you look at the area between the validation curve and the random curve.



Model graph for the simple method



Model graph for the adapted method

With each method you have been able to uncover more and more information from your data. When looking at the Maximum Smart Variable Contributions below, you can see that the majority of the most contributive variables come from the textual analysis of the data. The variable that contributes the most to the target is `<tc_listing_title_capacity_2gb>`.

## 6.4 Annex - Regular Expression Reminder

The regular expressions engine used for the stemming rules is a PCRE engine (Perl Compatible Regular Expression). The following table summarizes the main elements that can be used in the regular expressions:



\	general escape character with several uses
^	assert start of subject (or line, in multiline mode)
\$	assert end of subject (or line, in multiline mode)
.	match any character except newline (by default)
[	start character class definition
]	end character class definition
	start of alternative branch
(	start sub pattern
)	end sub pattern
?	extends the meaning of ( , also 0 or 1 quantifier, also quantifier minimizer
*	0 or more quantifier
+	1 or more quantifier
{	start min/max quantifier
}	end min/max quantifier

# Important Disclaimers and Legal Information

## Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

## Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

## Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

## Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

## Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.



© 2020 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 <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.