Hands-On Tutorial SAP Predictive Analytics, Automated Mode: Data Manager

Sharpen your data to produce stronger predictive models. This guide shows how to use the Data Manager in SAP Predictive Analytics to derive additional variables for your analysis, leading to a more detailed picture and better predictions.

The data used in this guide is publicly available so that the readers can follow hands-on and carry out the same analysis themselves.

June 2016

Andreas Forster
Predictive Presales Expert
SAP Switzerland
andreas.forster@sap.com

TABLE OF CONTENTS

INTRODUCTION

You may know, or guess, that the “Automated Analytics” in SAP Predictive Analytics is all about automating the process of creating predictive models.

Part of that process is to help the user create additional variables that help the model predict more accurately. This is what the Data Manager is primarily for. Without having to create new columns in the database, which might or might not be used in a model, the Data Manager creates a semantic view on top of the physical layer of a database. This is done in a graphical interface, which can produce hundreds or thousands of variables.

Think of creating such variables like a digital camera that zooms in on a situation. Creating these variables is very similar for the predictive model. They help to have a closer look and to better understand the situation. The model can take this information into account to get a clearer picture. Also similar to a digital camera, that picture is taken in “Automatic” mode. There is no need to fiddle with the detailed settings. Just keep producing models with little effort.

A very comprehensive framework underpins this creation, ensuring that powerful models are created without putting any constraints on the data. If you like to know more about the overall concept, you may want to read this article.[1]

This document explains how to create these detailed variables with the Data Manager, laying the foundations for strong predictive models. You can simply read this document or even implement your first models based on logic from Data Manager by following the steps hands-on. As example, we will predict which customers of a bank are likely to be interested in signup up for a credit card. Whilst this tutorial uses Data Manager in combination with a classification, Data Manager can also enhance other models, such as Regressions, Clustering or Time Series.

You may benefit most from this guide if you have already got some first experience with SAP Predictive Analytics. In case you have not used SAP Predictive Analytics before, I suggest to have a look at another tutorial first, in which you create predictions based on existing information/columns.[2] You can also get an overview with the official tutorials.[3]

To learn more about Data Manager please see the “Getting Started with Data Manipulation” help file.[4]

I would like to thank Abdel Dadouche, Gaëtan Saulnier and Orla Cullen, whose content of a hands-on session at a TechEd && d-code event inspired the creation of this guide.

[1] “How does Automated Analytics do it? The magic behind creating predictive models automatically” http://scn.sap.com/docs/DOC-65046

[2] “Hands-On Tutorial SAP Predictive Analytics, Automated Mode: Classification” http://scn.sap.com/docs/DOC-68110

[3] “Official Product Tutorials” http://scn.sap.com/docs/DOC-32651

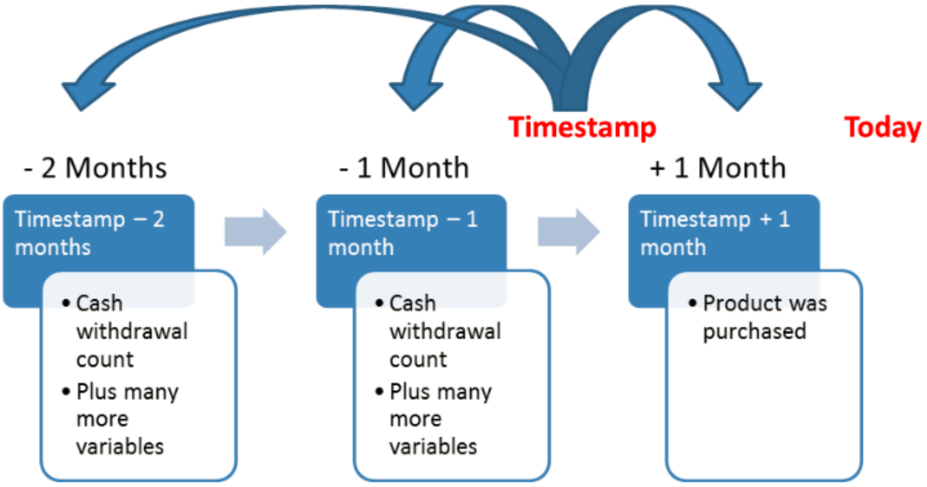
[4] Getting Started with Data Manipulation: http://help.sap.com/pa

COMPONENTS

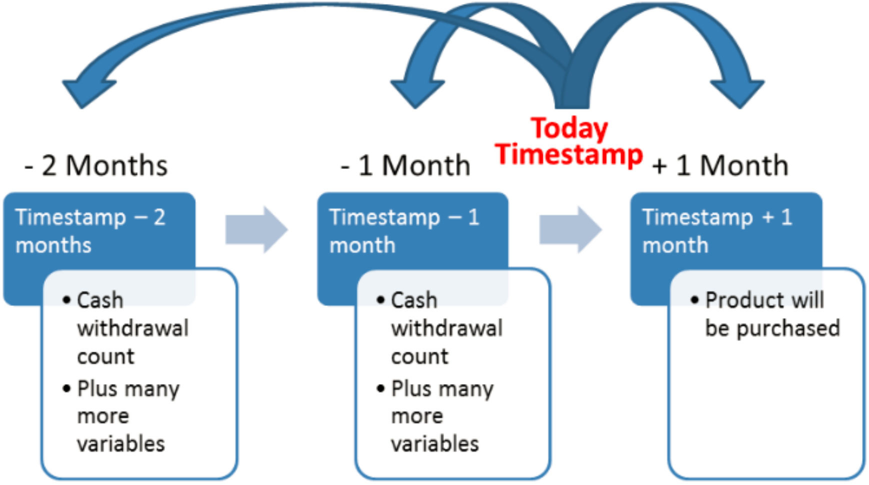
The Data Manager creates a semantic view called **"Analytical Data"**, or "Analytical Dataset". It is a graphical interface to join existing database tables and to create additional columns that can help the predictions. Note that the columns to not have to be created one-by-one. You will see that many detailed variables can be created at once.

The view has an understanding of time built in. It is able to relate to a certain point in time (called time-stamp). Information from before the time-stamp describes past behaviour and is used to train a predictive model. The view also contains a target variable describing a behaviour after the time-stamp, on which a model will be trained. The trained model can then be applied with a more recent timestamp (or even today's datetime) to predict future behaviour.

Example for training a model on past data with a time-stamped population.



Example for applying a model on a time-stamped population to predict the target variable.



The Analytical Dataset created by the Data Manager consists of the following three components. Only once all these components have been created, the dataset can be used for predictions.

Component	Description
Entity	Whose behaviour to predict
Time-stamped Population	What behaviour to predict
Analytical Record	Historic information the predictions are based on

Entity
Everything starts with the entity. The entity is the person or item whose behaviour we want to forecast. Based on the business requirement, it is immediately clear what the entity is going to be. In our example we want to forecast the behaviour of customers. So in that case the entity is a customer.

Note that Data Manager requires that you have a table with unique entries for all relevant customers. Duplicates are not allowed in this table.

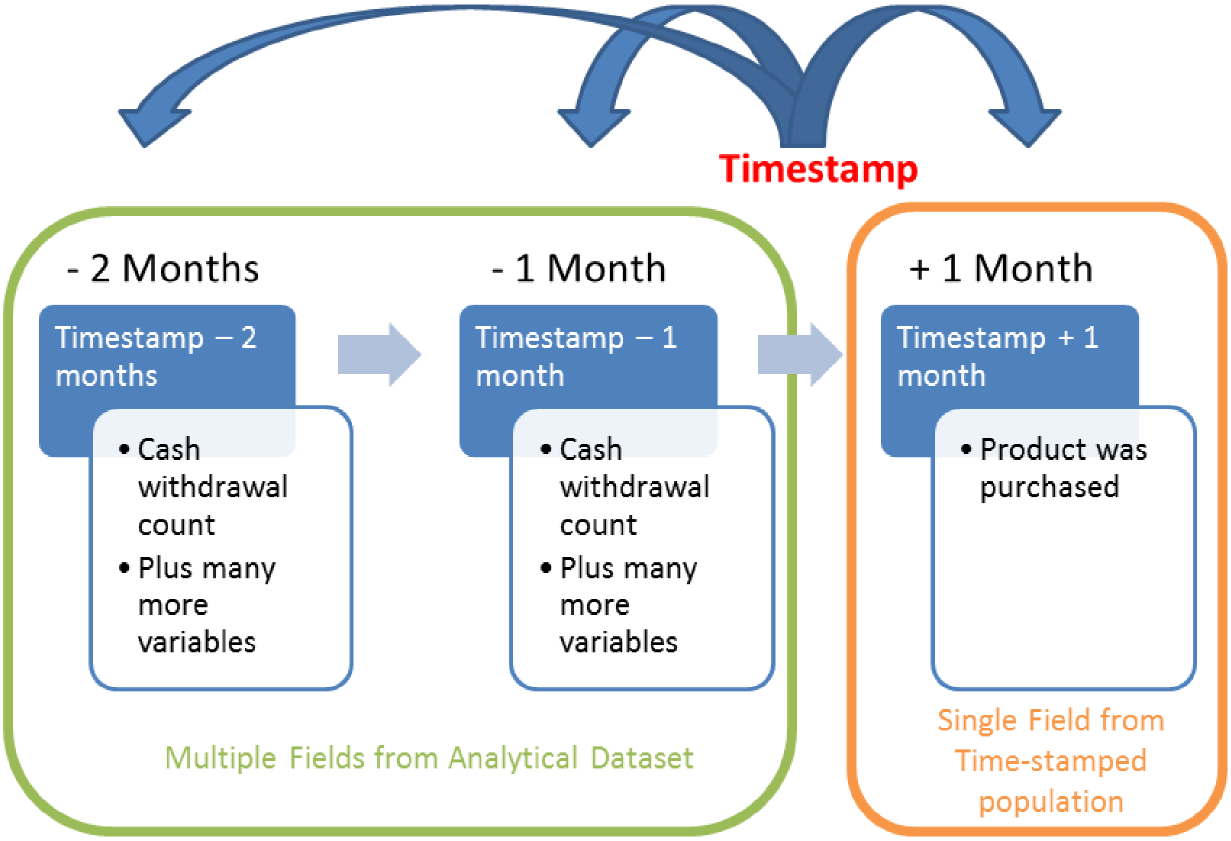
Time-stamped Population
The time-stamped population enriches the entity with the business context of what behaviour is to be forecasted. This contains two pieces of information

- A filter on the entities to select only the ones relevant for the prediction
- A target variable, which describes during training what behaviour the entity showed. When applying the model, it describes the probability of showing the behaviour in future. For a classification, as shown in this tutorial, this indicator has to be binary, ie yes/no, 1/0 or any other combination of two values. To carry out a regression or time-series forecast the variable must be numeric. The target variable is the only variable that is based on information after the time-stamp.

Analytical Record
The analytical record describes the entity's behaviour (so the customer's in our case) before the time stamp. Such descriptions can be rather static (ie "Gender", "Natonality") or dynamic (ie "Age", "Income last month", Cash withdrawal count last month").

Typically you will spend most of the time trying to improve a model by providing additional variables in the Analytical Record. If the relevant data is spread over multiple columns, you can join these and dynamically derive new columns, ie through aggregation or pivoting.

As a quick recap: Columns from the Analytical Dataset describe the past. The target column from the time-stamped population is the only variable from after the time-stamp describing the future behaviour to predict.



Examples
Here are some examples for such Analytical Datasets:

Purchasing Affinity: Bank's customers' interest in credit card

Component	Description
Entity	Customer
Time-stamped Population	As seen from the time stamp: Filter: Existing customers who have not yet signed up for a credit card Target Variable: Indicator whether the customer did or will sign up in the future time-window
Analytical Record	Past behaviour of the customer, as seen from the timestamp, ie - Age - Number of years the customer has been a client of the bank - Average amount on current account previous quarter - Average amount on current account the quarter before that - Change in amount between the quarters as absolute values and in percent - …

Customer Churn: Risk of losing an insurance customer

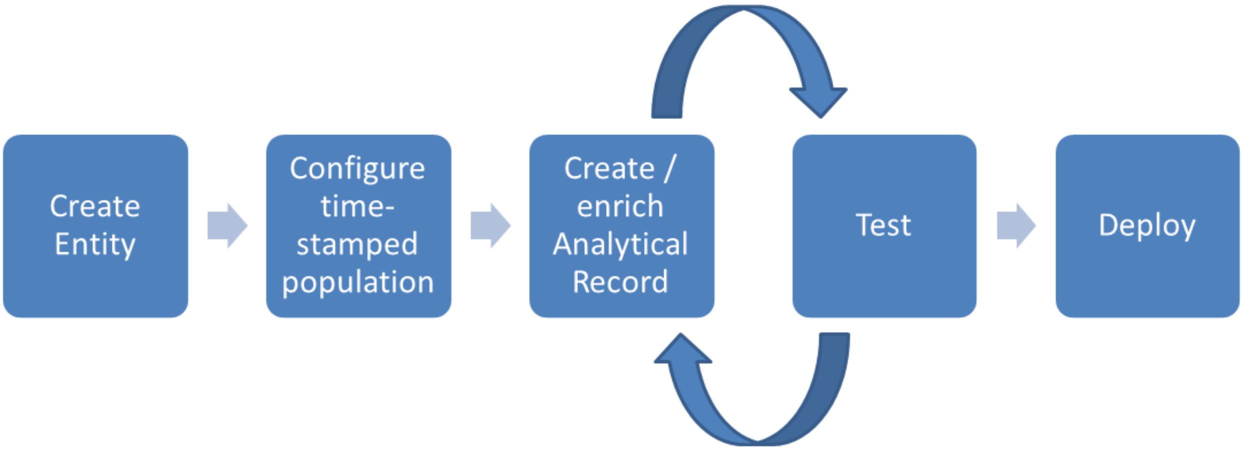
Component	Description
Entity	Customer
Time-stamped Population	As seen from the time stamp: Filter: Existing insurance customer Target Variable: Indicator whether the customer did or will cancel the existing contract in the future time-window
Analytical Record	Past behaviour of the customer, as seen from the timestamp, ie - Age - Number of years the customer has been a client of the insurer - Number of claims last year - Number of claims last year that were not reimbursed in full - …

Preventive Maintenance: Risk of a manufacturing machine/device showing unexpected behaviour

Component	Description
Entity	Manufacturing machine
Time-stamped Population	As seen from the time stamp: Filter: Machines that have been running without anomalies after the last maintenance interval Target Variable: Indicator whether the machine did or will show an unusual pattern in the future time-window (ie unexpectedly high vibrations)
Analytical Record	Past behaviour of the machine, as seen from the timestamp, ie - Duration since last maintenance - Count of items produced since last maintenance - Maximum temperature previous day - Maximum temperature two days ago - Change in temperature between the two days in degrees Celsius or Fahrenheit - Change in temperature between the two days in percent - …

MODELING STEPS

The modeling of the Analytical Dataset is closely linked with creating / testing the predictive model. You have to create the entity, time-stamped population and a first Analytical record before testing the logic with a predictive model. If this step is successful, you enrich the Analytical Record to better describe the time-stamped population. You keep testing the prediction and enriching the Analytical Record until you are happy to deploy the model.


Create Entity
Configure time-stamped population
Create / enrich Analytical Record
Test
Deploy

Create Entity

The entity (ie Customer) is defined with a single table with unique entries for each entity. The table can have multiple columns, you just need to specify an identifier column for the entity (ie an ID column). Automated Analytics will rename the ID to “KxId”, clearly indicating the importance of this column.

Configure time-stamped population

Now link the entity table with further tables to filter the dataset according to the use case and to calculate the target variable. One of the tables must contain a date column so that the time-stamp concept can be applied. The time-stamp is used to

- Create the target variable to indicate the future behaviour. Note that the variable must return only two values, ie “Yes” and “No or 1 and 0 or any other value pair. So to predict whether a customer will sign up for a credit card, this variable will indicate whether the customer did sign up in the time-frame defined after the time-stamp.
- Apply a filter so that the dataset is restricted according to the use case, ie exclude any customers that had already signed up for a credit card before the time-stamp. As they have already purchased the product, their profile in the dataset would distort the model.

Create Analytical Record

Next, you need to create the Analytical Record to describe the entity’s past behaviour. In the Analytical Record you link further tables to the entity. Columns from these tables are immediately added. Further columns can be calculated. If you need to use tables that have already been used in the time-stamped population, these tables also need to be added again to the Analytical Record.

At this point you just need to create a very basic Analytical Record. Once the Analytical Record is saved, you can test the model by creating a classification in the Automated Modeler.

Test Analytical Record

Open the Automated Modeler to test the Analytical Record. Have a look how the data is described. If the model does not give any warnings or error you know you are on a good track and you can further enhance the Analytical Record.

Enrich Analytical Record & Test

Back to the Analytical Record, keep adding additional columns to describe the entity and its behaviour in more detail. Keep testing the model every so often. Models can be saved under different versions, so it is easy to go back to an earlier model if needed.

Some examples of the types of variables that can be created are

- **Aggregates**: Your data might hold detailed transactional data about activities in your customers' accounts. Out of this history, Data Manager can create aggregates such as the count of the transactions or the average amount per transaction.
- **Pivots**: Pivoting builds on the above aggregation and makes it easy to graphically create a large number of more detailed aggregates. Sticking to the same example of activities in a customer account, pivoting can create individual counts of transaction types. So you can have transaction counts by cash withdrawals, standing orders, and so on.
- **Understanding of time**: Typically it is crucial to put the historic data into a context of time. Therefore the Data Manager has an in-built concept to relate the various measures to moments or ranges in time. Again, without the need for any coding, Data Manager can create detailed variables such as
  - Count of cash withdrawals in the previous quarter.
  - Count of cash withdrawals in the same quarter the year before.
  - Change in cash withdrawal counts in absolute values.
  - Change in cash withdrawal counts in percent.

The more variables you have available the better. Automated Analytics will find out which variables are needed for the model and which ones can be eliminated.

Aim to have the “Prediction Confidence” equal to or greater than 0.95 and ty to get the “Predictive Power” as high as possible (close to 1). In order to increase "Predictive Power" you can try adding additional variables. To increase "Predictive Confidence" you can try adding additional rows of data.

Deploy

Once you are happy with the model you put it into action. This could be scoring your current customer base and to write the probabilities in the database. It could also mean to add a new scoring column to an existing database view for real-time scoring or you could embed the model in various programming languages such as C++ or JavaScript into your application.

In productive environment you will use the “Model Manager” to monitor the model’s accuracy over time and to retrain if needed. This step is not covered in this guide.

Reuse Analytical Record

Often a large part of the first Analytical Record you have created can be reused for further predictions. The input columns used for additional predictions, ie probability to upgrade from a single product to a bundle, will be very similar. Just make sure that the target variable is the only column using data from after the time-stamp.

HANDS-ON IMPLEMENTATION

Background

We want to help a bank optimize its marketing for credit cards and we will predict which customers are most likely interested in signing up for a credit card next quarter. We will use a dataset that was shared as part of the “3rd European Conference on Principles and Practice of Knowledge Discovery in Databases” held in Prague in the year 1999, or abbreviated PKDD ’99.[5]

This dataset contains information about individual (anonymous) customers, some general data about the person and some activity on the accounts. For a more detailed description you can see the Appendix.

Pre-Requisites

You need the following elements to be able to follow the steps hands-on

- It is assumed you have a fair understanding of SAP HANA with some hands-on experience of loading data.
- It is also assumed you have had some first experience creating a classification with SAP Predictive Analytics, Automated Mode In case you are new to SAP Predictive Analytics, see the introduction of this guide for some initial tutorials to get you started.
- You need to have access to a SAP HANA system in which you can load the PKDD ’99 data. See the chapter “Data Load” in the appendix. You are given the database content, which can be loaded easily into SAP HANA. Just note, that some of the data has been transformed from its original structure. For instance the client’s birthdate and gender are stored in a single column in the original data. This has been separated in two columns.
  In case you don’t have a SAP HANA system available, you might be able to use subscription-based system at relatively low cost, for instance on Amazon Web Services.[6]
- Most obviously, you need an installation of SAP Predictive Analytics, which includes the Data Modeler and Automated Mode. This guide has been written with SAP Predictive Analytics 2.1. Evaluation copies are currently available on the SAP Community Network.[7]
- Finally, you need to set up an ODBC Source from the computer with SAP Predictive Analytics to the SAP HANA system. In this document the ODBC source is called “My HANA” and the schema name is “I056450”. In your environment those details will be different.

[5] PKDD ’99, http://lisp.vse.cz/pkdd99/

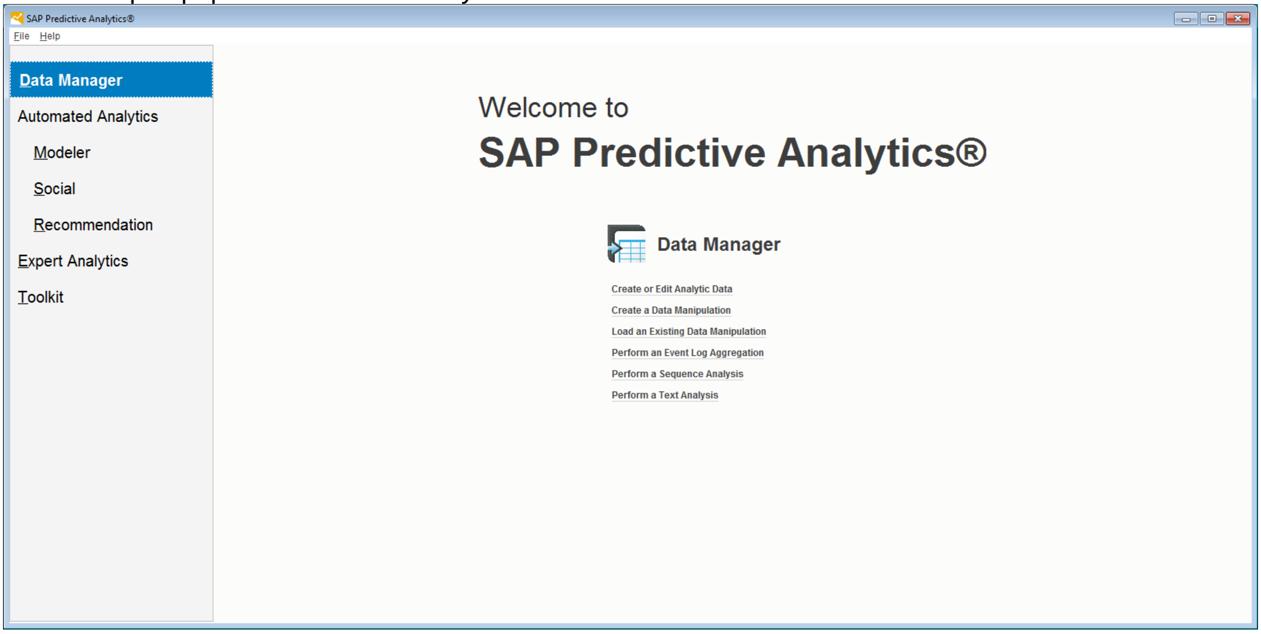
[6] Amazon Web Services, http://aws.amazon.com/sap/

[7] SCN, http://scn.sap.com/community/predictive-analytics

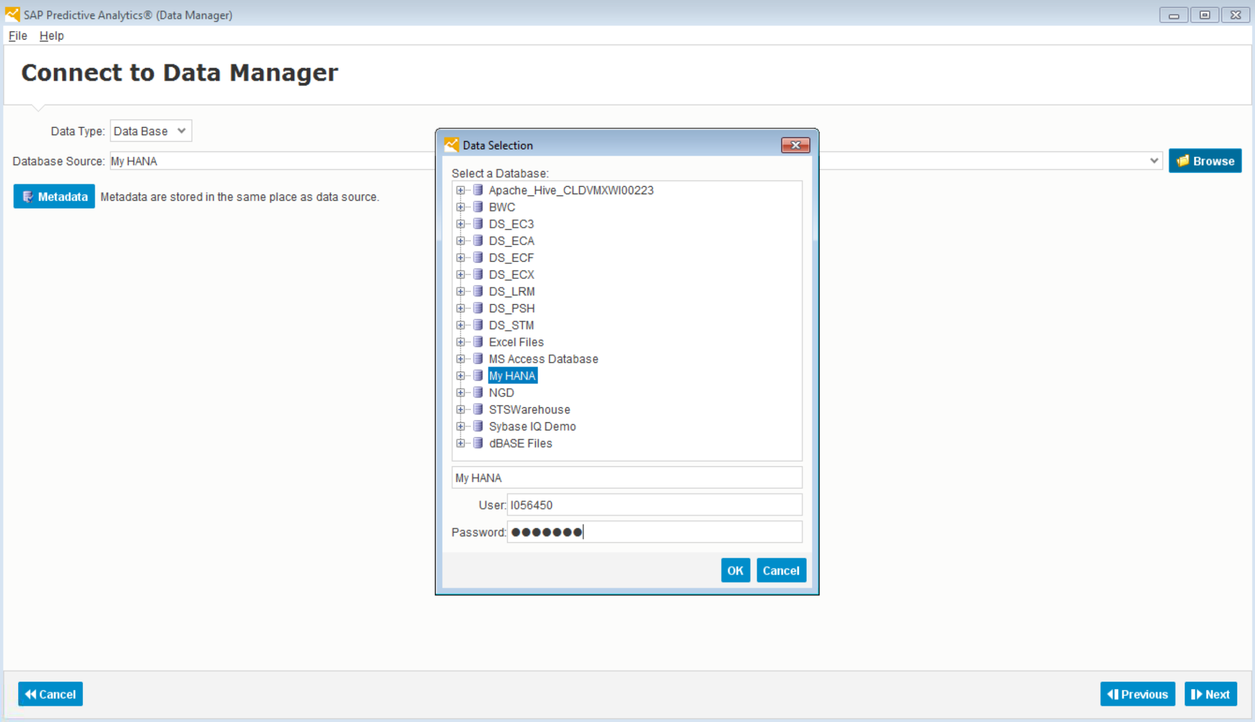
Step 1 – Create Entity

Start by opening up SAP Predictive Analytics. The version used in this document is SAP Predictive Analytics 2.1.

Click on “Data Manager” → “Create or Edit Analytical Data”. This Analytical Data will contain out Entity, the Time-stamped population and the Analytical Record.


SAP Predictive Analytics®
File Help
Data Manager
Automated Analytics
Modeler
Social
Recommendation
Expert Analytics
Toolkit
Welcome to
SAP Predictive Analytics®
Data Manager
Create or Edit Analytic Data
Create a Data Manipulation
Load an Existing Data Manipulation
Perform an Event Log Aggregation
Perform a Sequence Analysis
Perform a Text Analysis

Ensure the “Data Type” is set to “Data Base”. Click “Browse” and select the ODBC source to the SAP HANA system. Remember to enter the SAP HANA user name and password in this window. Hit “OK”.



SAP Predictive Analytics® (Data Manager)

Connect to Data Manager

Data Type: Data Base

Database Source: My HANA

Metadata are stored in the same place as data source.

Data Selection

Select a Database:

- Apache_Hive_CLDVMXWI00223
- BWC
- DS_EC3
- DS_ECA
- DS_ECF
- DS_ECX
- DS_LRM
- DS_PSH
- DS_STM
- Excel Files
- MS Access Database
- My HANA
- NGD
- STSWarehouse
- Sybase IQ Demo
- dBASE Files

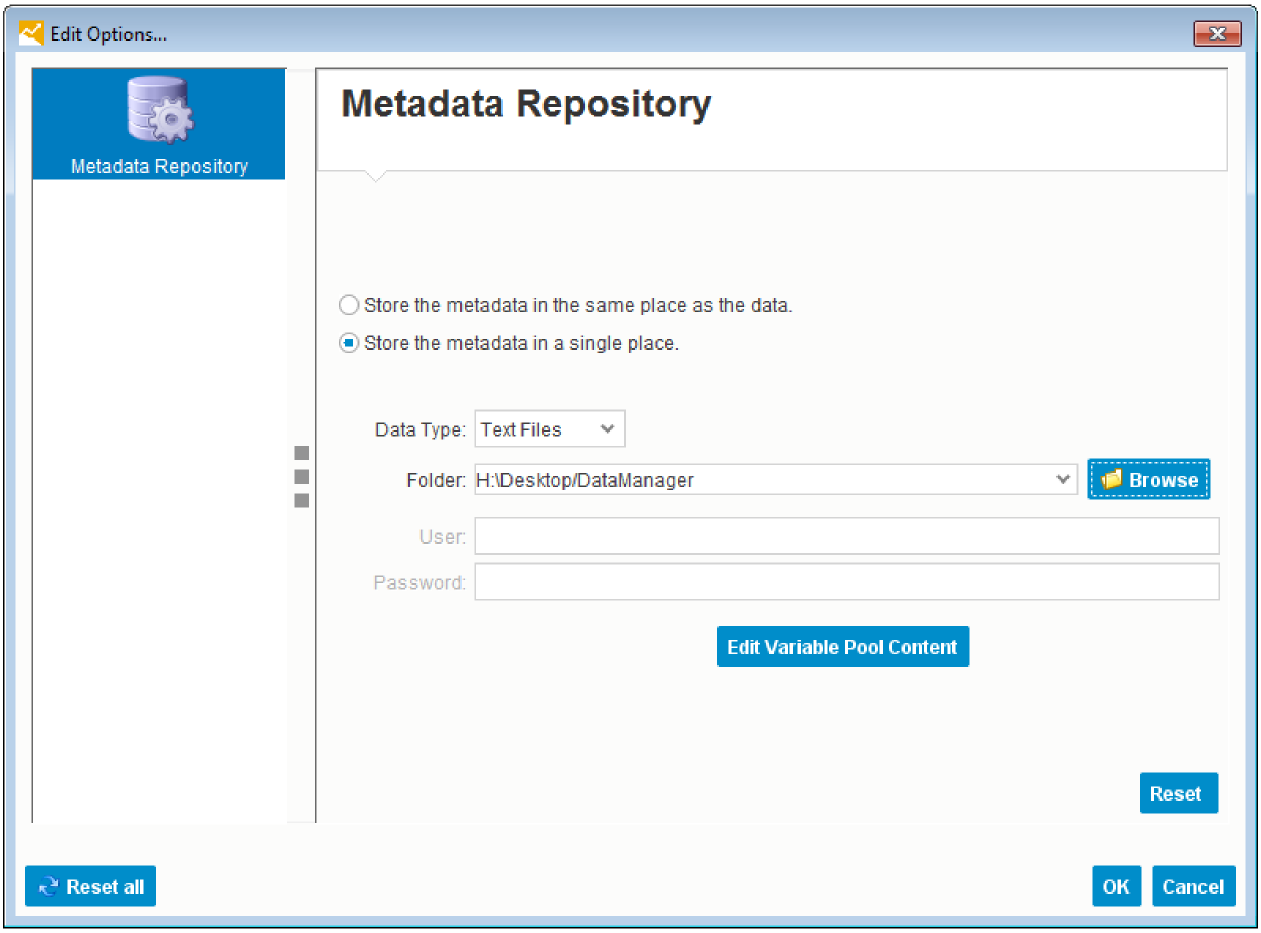
My HANA

User: I056450

Password: ●●●●●●●

Now specify where to save the metadata of the “Analytical Data” model. By default it is saved in the same database, that we use for the prediction. In that case SAP Predictive Analytics creates additional tables in the database which will contain the metadata. However, we save the model locally as files on our local computer.

Click “Metadata”. Select “Store the metadata in a single place”. Set “Data Type” to “Text Files” and select a folder.



○ Store the metadata in the same place as the data.

● Store the metadata in a single place.

Data Type: Text Files

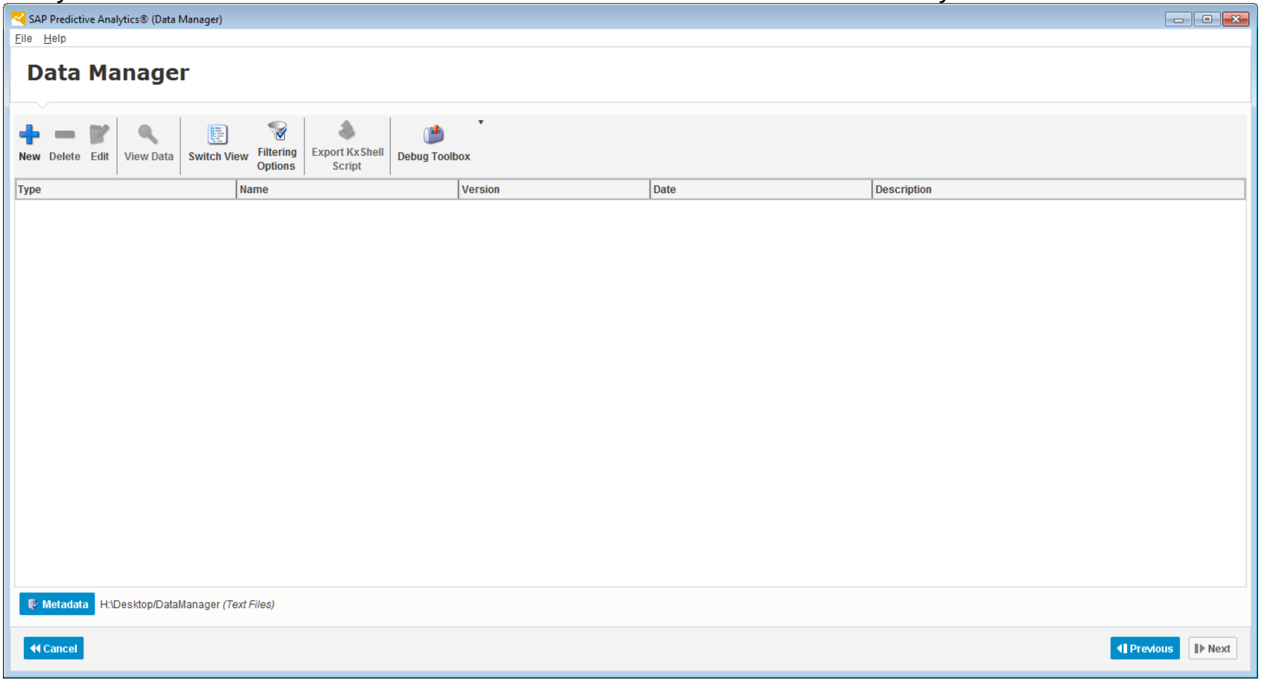
Folder: H:\Desktop/DataManager

User:

Password:

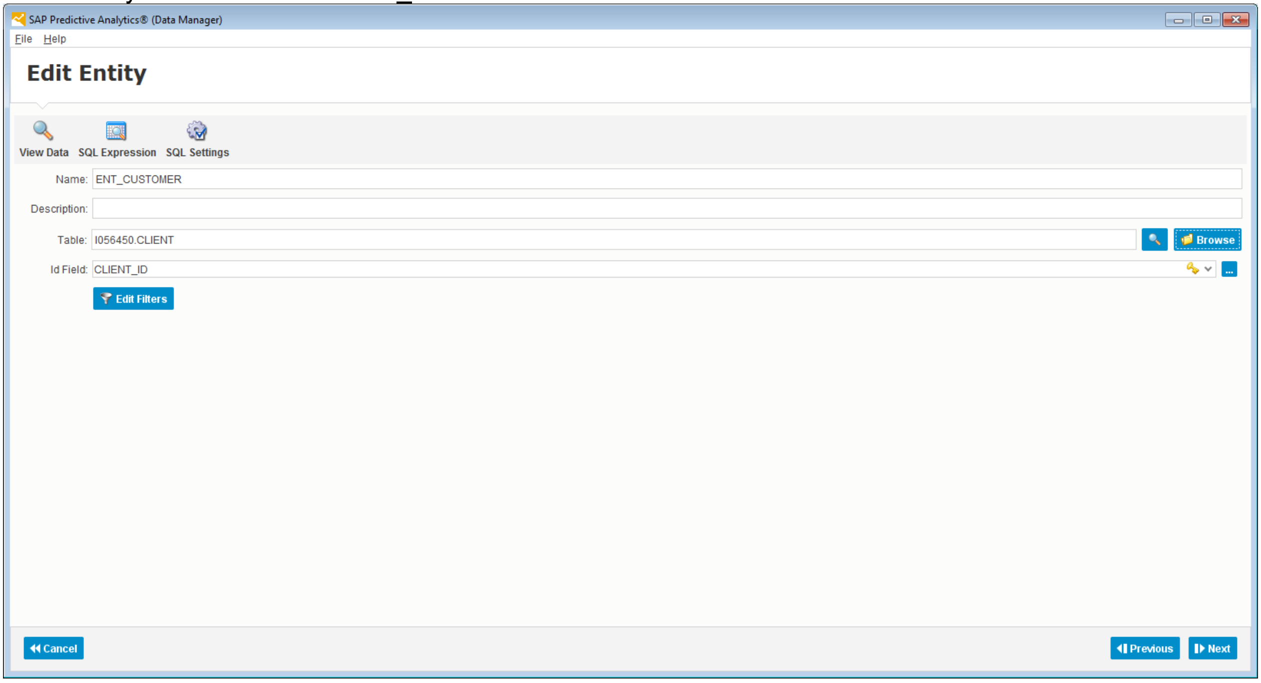
Click “OK” → “Next”.

Now we are ready to start creating our data model. The Entity, the Time-stamped Population and the Analytical Record are all created from this window. We will see this screen many times.



H:\Desktop\DataManager *(Text Files)*

You have to start with the entity whose behaviour we want to predict. Click “New” → “Entity”. Name it “ENT_CUSTOMER”. Select the “CLIENT” table and set the “Id Field” to the column that uniquely identifies each entitiy. Here this is “CLIENT_ID”.



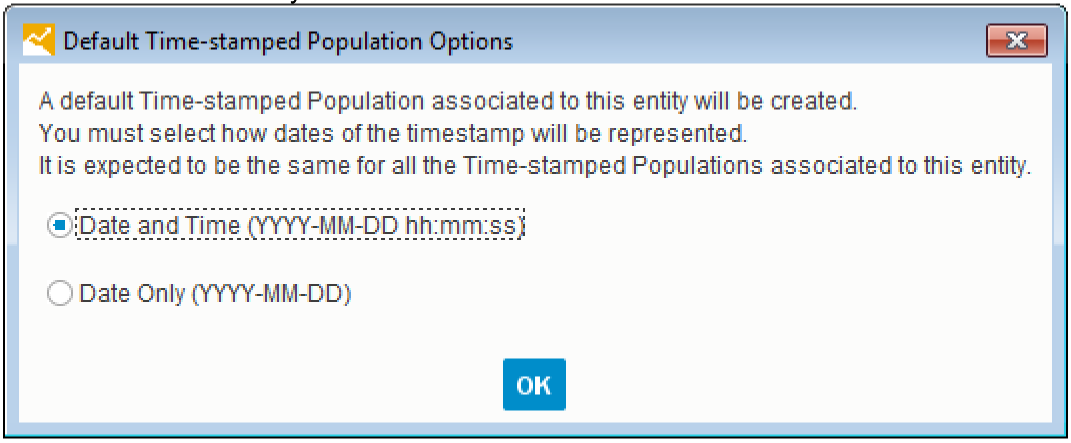
Name: ENT_CUSTOMER

Description:

Table: I056450.CLIENT

Id Field: CLIENT_ID

No further changes are needed. Hit “Next”. You will be asked for the Date type. Select “Date and Time” and “OK”. Most databases support joins on DateTime-stamps. If your database does not support such joins, you have to select “Date Only”.



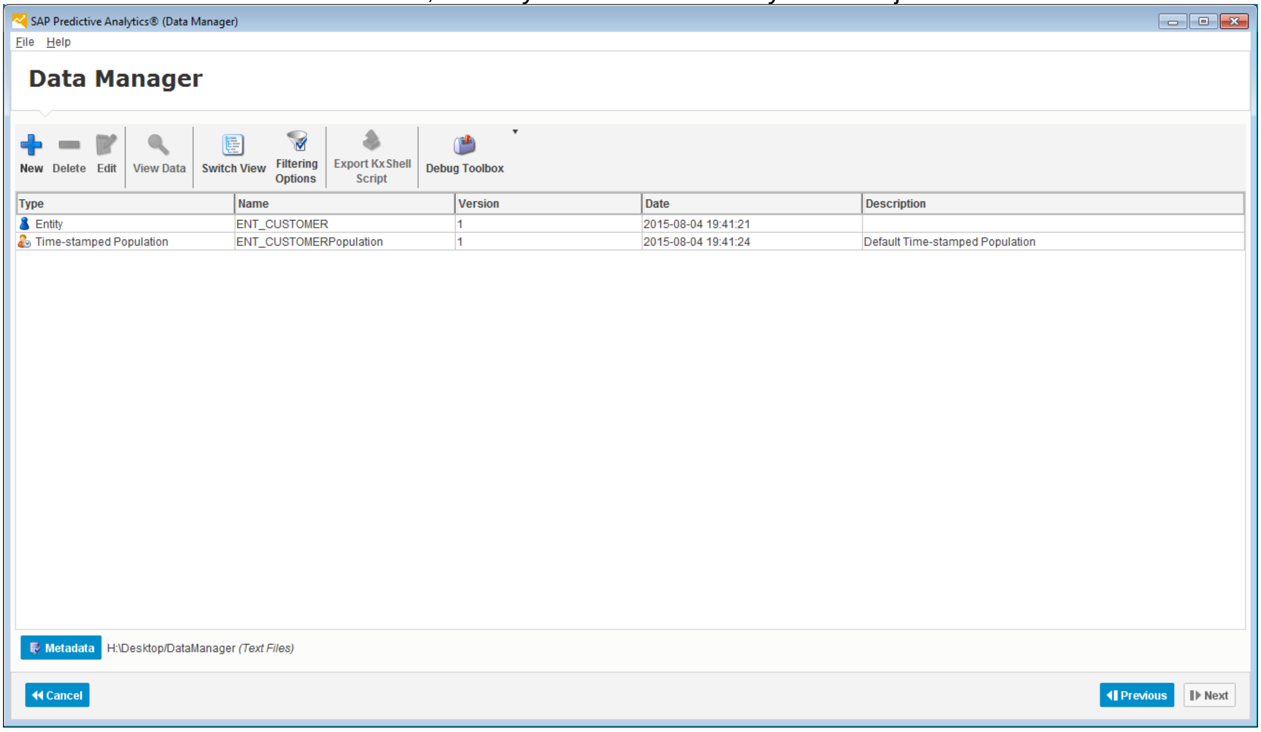
Default Time-stamped Population Options

A default Time-stamped Population associated to this entity will be created.
You must select how dates of the timestamp will be represented.
It is expected to be the same for all the Time-stamped Populations associated to this entity.

● Date and Time (YYYY-MM-DD hh:mm:ss)

○ Date Only (YYYY-MM-DD)

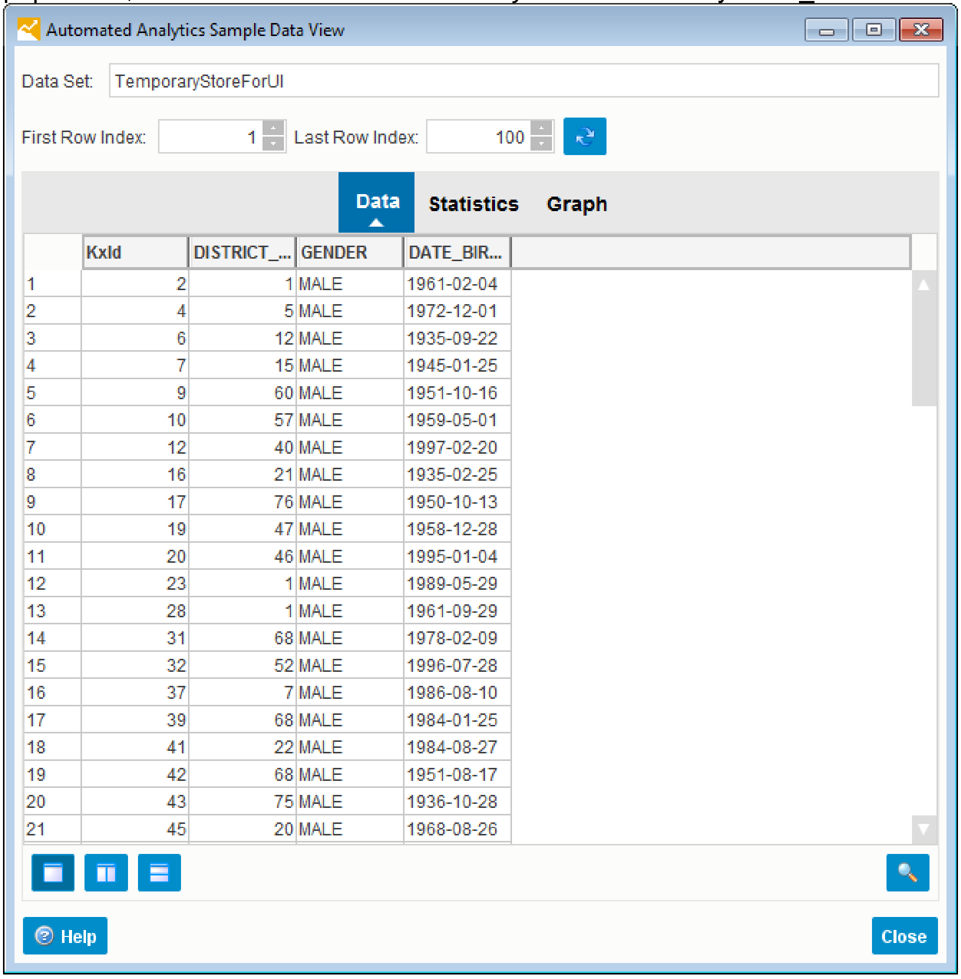
And we are back in the main screen, where you can see the entity we have just created.



Type	Name	Version	Date	Description
Entity	ENT_CUSTOMER	1	2015-08-04 19:41:21	
Time-stamped Population	ENT_CUSTOMERPopulation	1	2015-08-04 19:41:24	Default Time-stamped Population

H:\Desktop/DataManager *(Text Files)*

You can also see that a time-stamped population has been created. Before configuring this time-stamp population, look into the content of the entity. Select the entity “ENT_CUSTOMER” and click “View Data”.



Data Set: TemporaryStoreForUI

First Row Index: 1 Last Row Index: 100

	KxId	DISTRICT_...	GENDER	DATE_BIR...
1	2	1	MALE	1961-02-04
2	4	5	MALE	1972-12-01
3	6	12	MALE	1935-09-22
4	7	15	MALE	1945-01-25
5	9	60	MALE	1951-10-16
6	10	57	MALE	1959-05-01
7	12	40	MALE	1997-02-20
8	16	21	MALE	1935-02-25
9	17	76	MALE	1950-10-13
10	19	47	MALE	1958-12-28
11	20	46	MALE	1995-01-04
12	23	1	MALE	1989-05-29
13	28	1	MALE	1961-09-29
14	31	68	MALE	1978-02-09
15	32	52	MALE	1996-07-28
16	37	7	MALE	1986-08-10
17	39	68	MALE	1984-01-25
18	41	22	MALE	1984-08-27
19	42	68	MALE	1951-08-17
20	43	75	MALE	1936-10-28
21	45	20	MALE	1968-08-26

You see a preview of the data. Notice that the entity’s ID got renamed from “CLIENT_ID” to “KxId”. In the database the column is still called “CLIENT_ID”,but within Data Manager this very important column is now referred to as “KxId”.

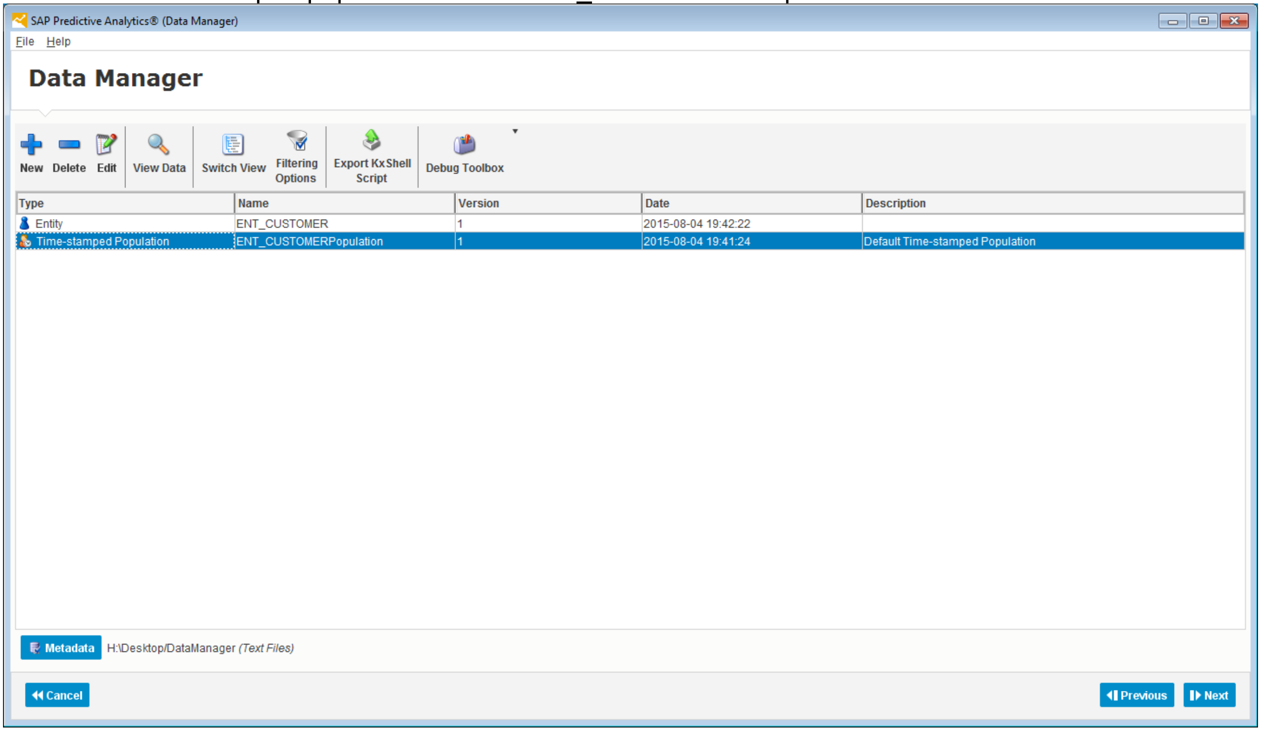
Click “Close”.

Step 2 – Configure Time-stamped Population

You will remember that the time-stamp population filters the dataset and that it provides the target variable. It does not matter in which sequence you create these elements, we start with the filtering.

In our case, we need to reduce the dataset to customers that have not taken out a credit card before the timestamp.

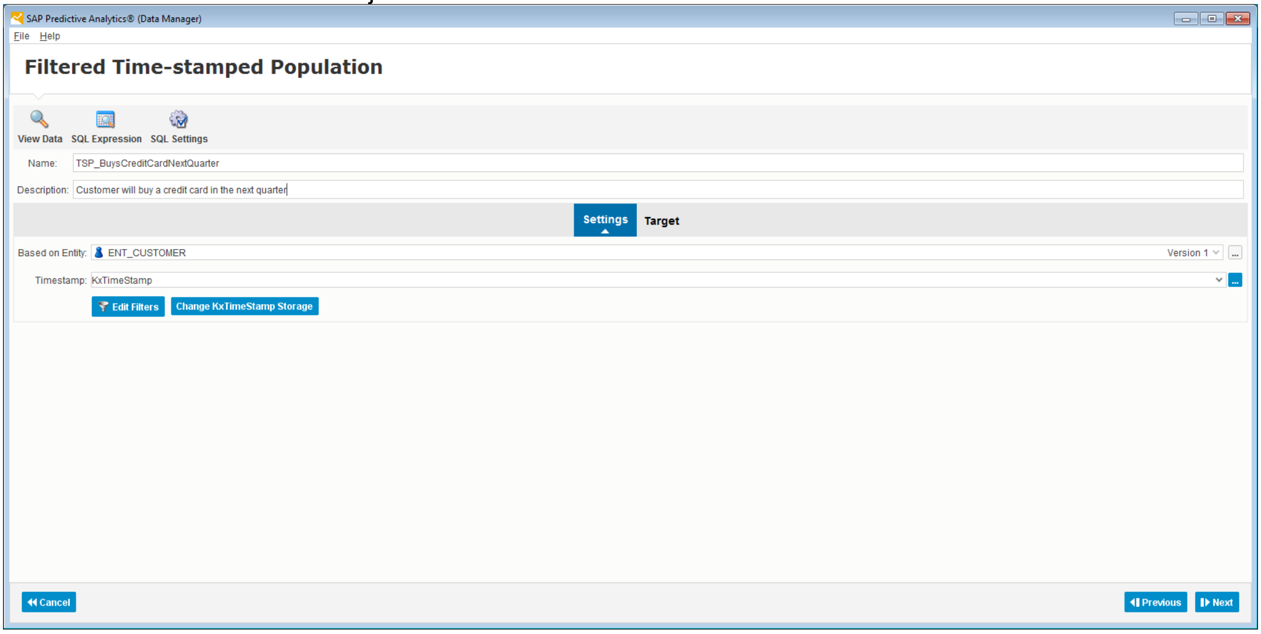
Select the Time-stamped population called "ENT_CUSTOMER\Population" and click "Edit".



Type	Name	Version	Date	Description
Entity	ENT_CUSTOMER	1	2015-08-04 19:42:22	
Time-stamped Population	ENT_CUSTOMERPopulation	1	2015-08-04 19:41:24	Default Time-stamped Population

Metadata H:\Desktop/DataManager *(Text Files)*

Notice how the population is linked to the entity we created earlier. Change the object's name to "TSP_BuysCreditCardNextQuarter" and give it a meaningful description so we can easily find out content once we have created more objects.


SAP Predictive Analytics® (Data Manager)
File Help
Filtered Time-stamped Population
View Data
SQL Expression
SQL Settings
Settings
Target
Edit Filters
Change KxTimeStamp Storage
Cancel
Previous
Next

Name: TSP_BuysCreditCardNextQuarter

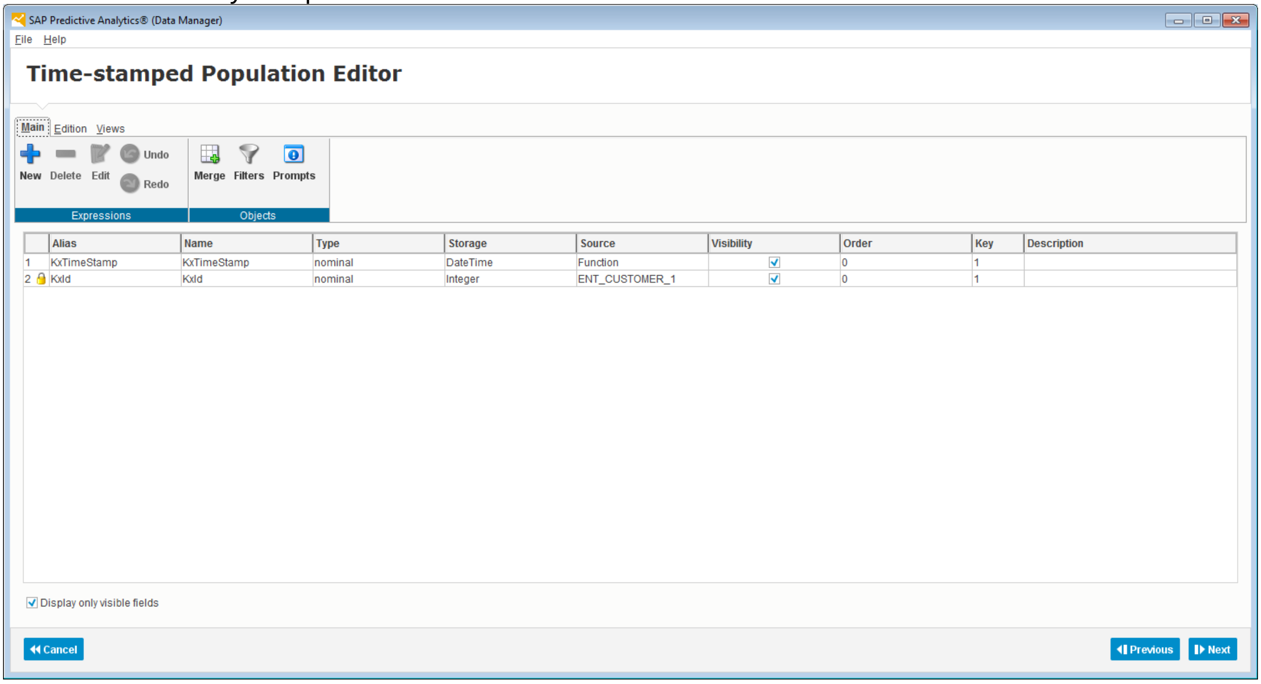
Description: Customer will buy a credit card in the next quarter

Based on Entity: ENT_CUSTOMER Version 1

Timestamp: KxTimeStamp

Now we create the filter. This filter ensures that based on the date-time parameter that is used, the dataset is filtered dynamically to include only customers without credit cards at that point in time.

Click on “Edit Filters” and you see the current structure of the data. So far it is very empty, because we have not created much. However, see the option “Display only visible fields”. Unticking this would display further fields from our entity. Keep it ticked.

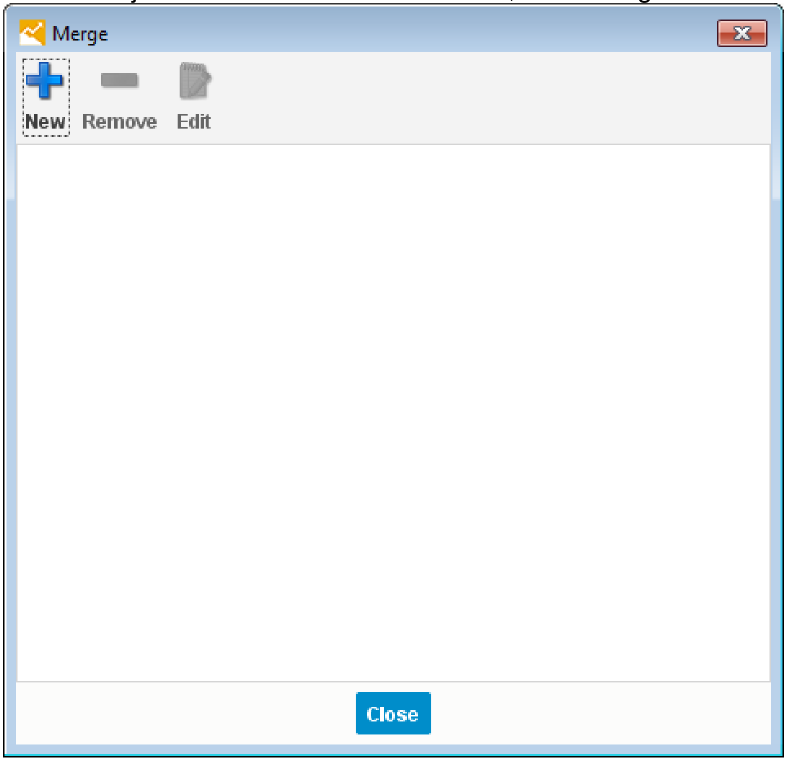

SAP Predictive Analytics® (Data Manager)
File Help
Time-stamped Population Editor
Main Edition Views
New Delete Edit Undo Redo
Expressions
Merge Filters Prompts
Objects
Cancel
Previous
Next

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxTimeStamp	KxTimeStamp	nominal	DateTime	Function	☑	0	1	
2	KxId	KxId	nominal	Integer	ENT_CUSTOMER_1	☑	0	1	

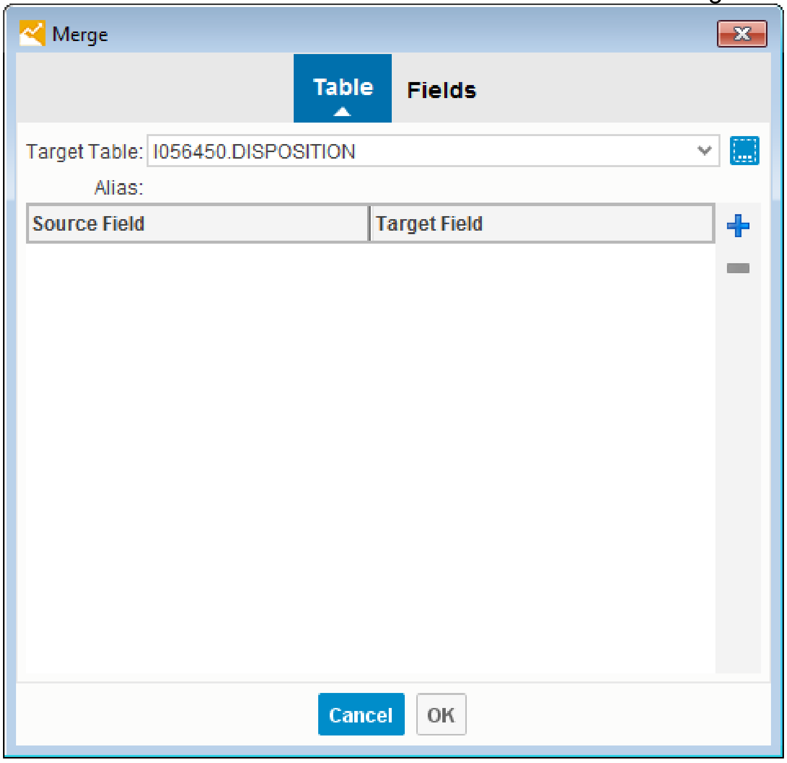
☑ Display only visible fields

The filter for the time-stamped population must be applied on an aggregate we still need to define. This aggregate will indicate whether the client does or does not have a credit card at the moment of the time-stamp. The information we are looking for is in the “CARD” table. Our entity is based on the “CLIENT” table. To get from the “CLIENT” table to the “CARD” table we have to link in the “DISPOSITION” table.

In order to join in the “DISPOSITION” table, click “Merge”.



Here click “New” and select the “DISPOSITION” table as target.

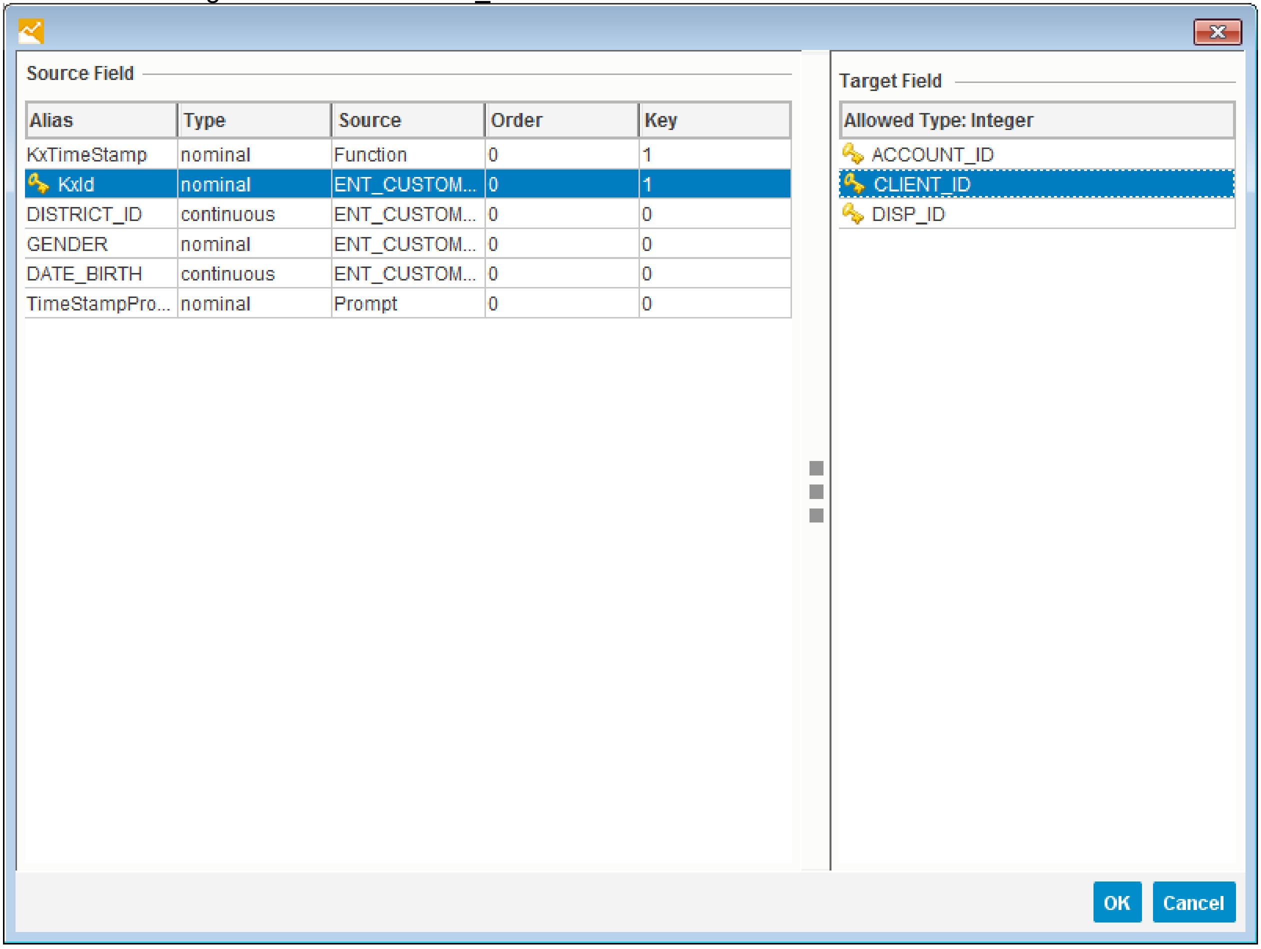


Target Table: I056450.DISPOSITION

Alias:

Source Field	Target Field

Now click the plus-sign on the right hand side to define the join. On the left-hand side select the “KxId” column. On the right select the “CLIENT_ID”.



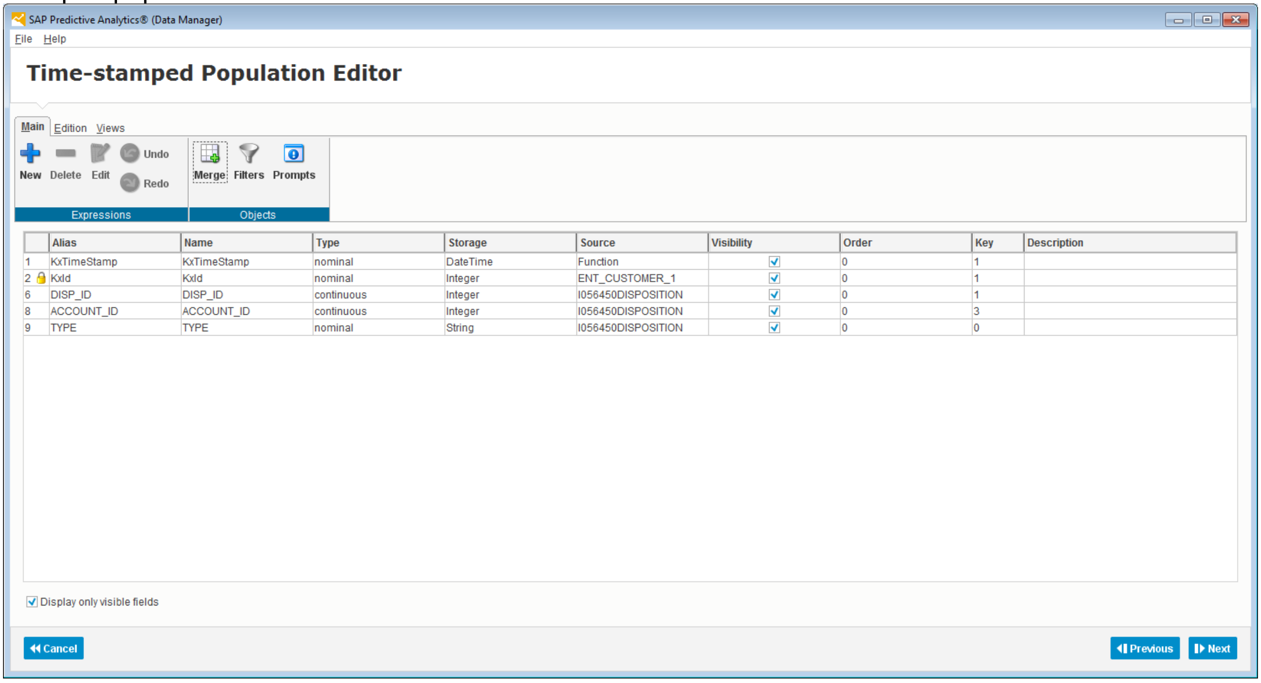
Source Field

Alias	Type	Source	Order	Key
KxTimeStamp	nominal	Function	0	1
KxId	nominal	ENT_CUSTOM...	0	1
DISTRICT_ID	continuous	ENT_CUSTOM...	0	0
GENDER	nominal	ENT_CUSTOM...	0	0
DATE_BIRTH	continuous	ENT_CUSTOM...	0	0
TimeStampPro...	nominal	Prompt	0	0

Target Field

Allowed Type: Integer
ACCOUNT_ID
CLIENT_ID
DISP_ID

Hit “OK” → “OK” → “Close” and you should see the columns of the “DISPOSITION” table in the time-stamped population editor.



Time-stamped Population Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxTimeStamp	KxTimeStamp	nominal	DateTime	Function	☑	0	1	
2	KxId	KxId	nominal	Integer	ENT_CUSTOMER_1	☑	0	1	
6	DISP_ID	DISP_ID	continuous	Integer	I056450DISPOSITION	☑	0	1	
8	ACCOUNT_ID	ACCOUNT_ID	continuous	Integer	I056450DISPOSITION	☑	0	3	
9	TYPE	TYPE	nominal	String	I056450DISPOSITION	☑	0	0	

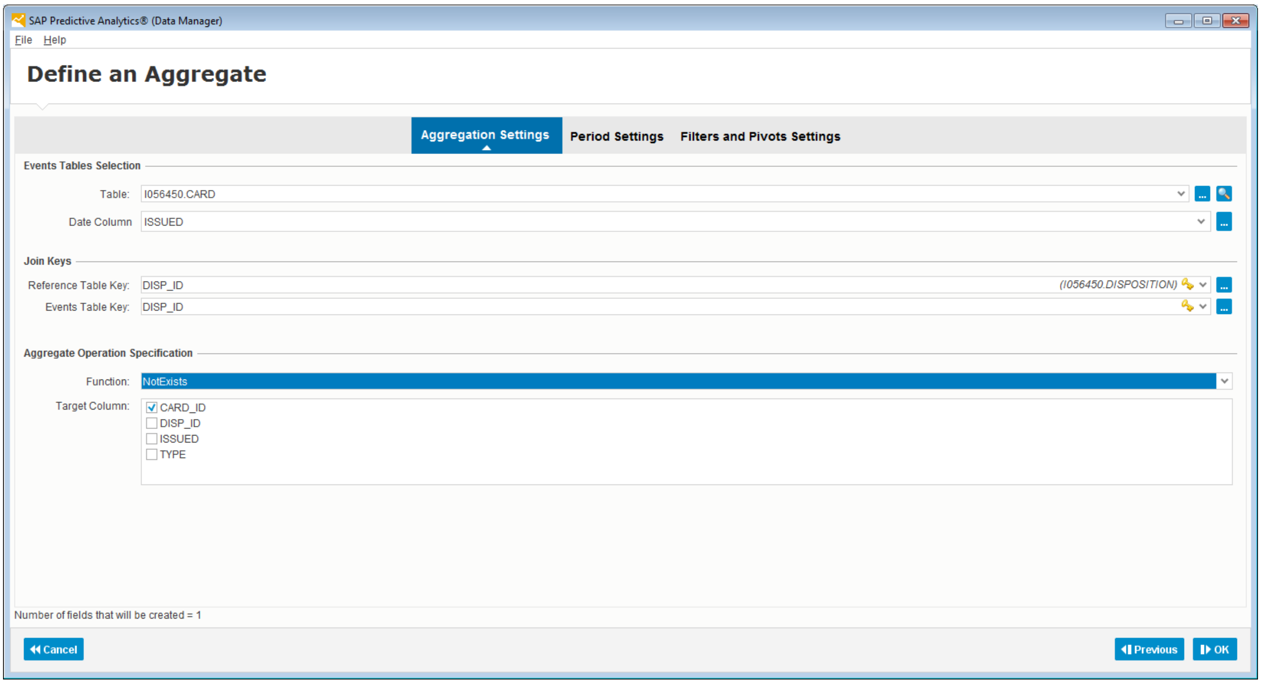
☑ Display only visible fields

Now we can create the aggregate that indicates whether a customer has a credit card at the given time stamp. The aggregate we create will include the link to the “CARD” table.

Click “New” → “New Aggregate”. Use the following settings as shown in the screenshot below:

- Link to the “CARD” table, which contains a date column named “ISSUED”. This date column will be used in relation with the time-stamp parameters.
- Link from the “DISP_ID” column of the “DISPOSITION” table to the “DISP_ID” of the “CARD” table.
- As aggregate use “NotExists” on the “CARD_ID” column.

If a client has not purchased a credit card, he will not have an entry in the “CARD” table and the aggregate will return TRUE.


SAP Predictive Analytics® (Data Manager)
File
Help
Aggregation Settings
Period Settings
Filters and Pivots Settings
Cancel
Previous
OK

Define an Aggregate

Events Tables Selection

Table: I056450.CARD

Date Column ISSUED

Join Keys

Reference Table Key: DISP_ID (I056450.DISPOSITION)

Events Table Key: DISP_ID

Aggregate Operation Specification

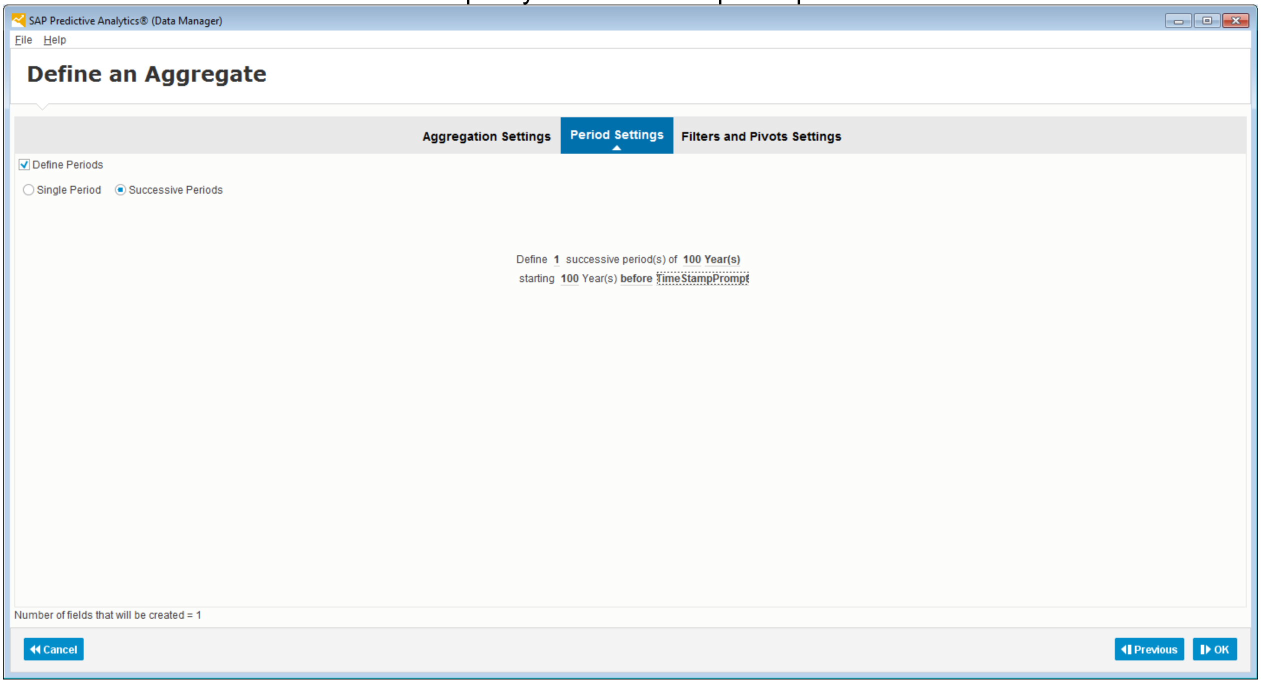
Function: NotExists

Target Column:
☑ CARD_ID
☐ DISP_ID
☐ ISSUED
☐ TYPE

Number of fields that will be created = 1

Click on the “Period Settings” tab, tick “Define Periods” and select “Successive Periods”. Here we define the time-range the aggregate relates to. Use the settings from the following screenshots.

We create one aggregate for a period of 100 years before the time-stamp. Be careful to get these settings correct. We use a period of 100 years starting 100 years before the time-stamp. The values can be clicked with the mouse. See below on how to specify the “TimeStampPrompt”.


SAP Predictive Analytics® (Data Manager)
File Help
Aggregation Settings
Period Settings
Filters and Pivots Settings
Cancel
Previous
OK

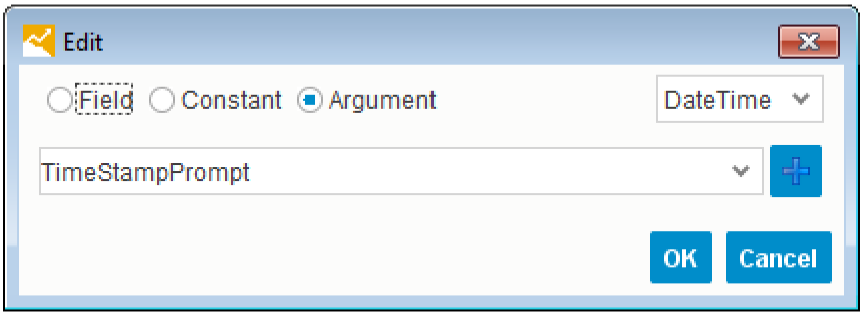
Define an Aggregate

☑ Define Periods

○ Single Period ● Successive Periods

Define 1 successive period(s) of 100 Year(s)
starting 100 Year(s) before TimeStampPrompt

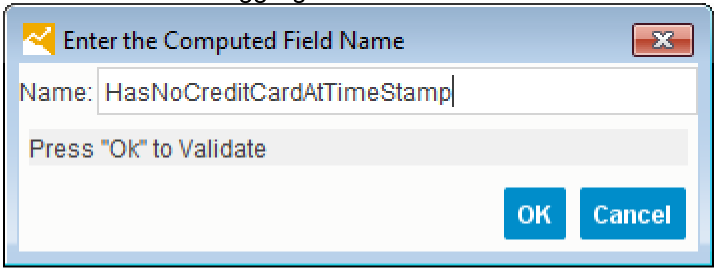
Number of fields that will be created = 1



○ Field ○ Constant ● Argument DateTime

TimeStampPrompt

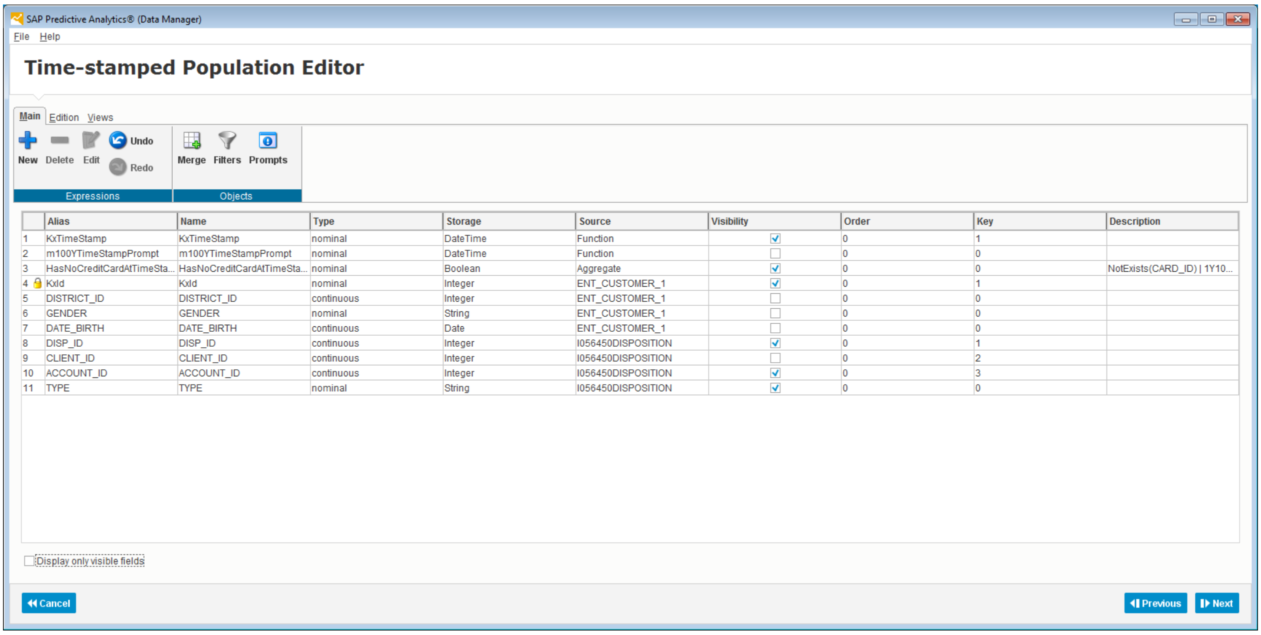
Please double-check your period settings are exactly as shown in the two screenshots above. Then hit “OK” and enter the new aggregate’s name “HasNoCreditCardAtTimeStamp”.



Name: HasNoCreditCardAtTimeStamp

Press "Ok" to Validate

Click “OK” and you see the new column. Should you not see the column, then untick “Display only visible fields” and it should appear. Now you can change the column’s visibility with a tick in the corresponding column.



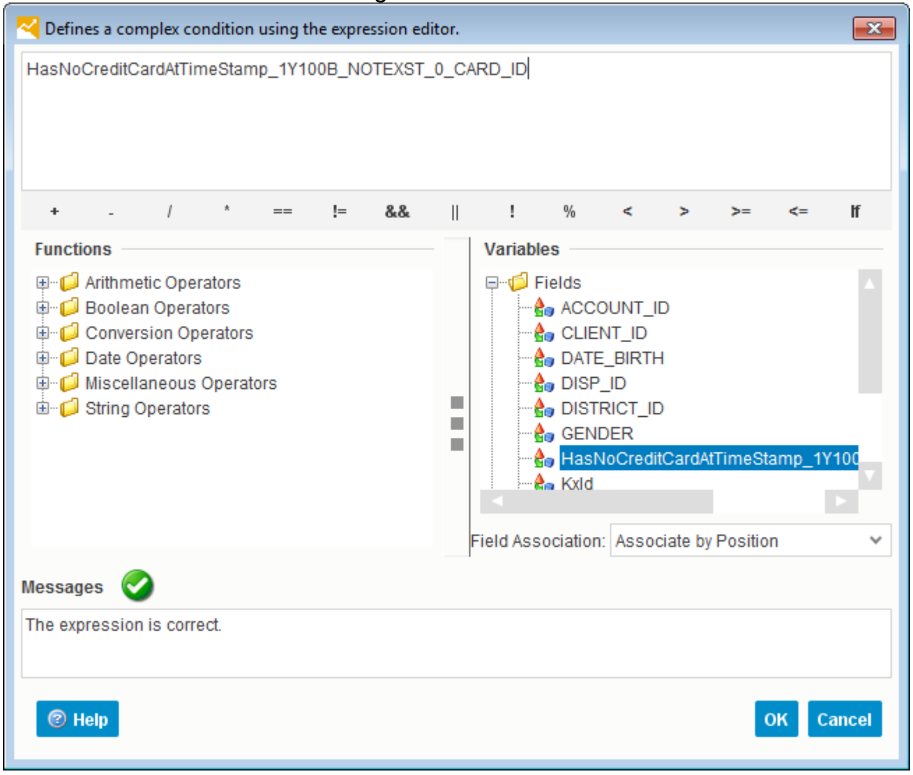
Time-stamped Population Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxTimeStamp	KxTimeStamp	nominal	DateTime	Function	☑	0	1	
2	m100YTimeStampPrompt	m100YTimeStampPrompt	nominal	DateTime	Function	☐	0	0	
3	HasNoCreditCardAtTimeSta...	HasNoCreditCardAtTimeSta...	nominal	Boolean	Aggregate	☑	0	0	NotExists(CARD_ID) | 1Y10...
4	KxId	KxId	nominal	Integer	ENT_CUSTOMER_1	☑	0	1	
5	DISTRICT_ID	DISTRICT_ID	continuous	Integer	ENT_CUSTOMER_1	☐	0	0	
6	GENDER	GENDER	nominal	String	ENT_CUSTOMER_1	☐	0	0	
7	DATE_BIRTH	DATE_BIRTH	continuous	Date	ENT_CUSTOMER_1	☐	0	0	
8	DISP_ID	DISP_ID	continuous	Integer	I056450DISPOSITION	☑	0	1	
9	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450DISPOSITION	☐	0	2	
10	ACCOUNT_ID	ACCOUNT_ID	continuous	Integer	I056450DISPOSITION	☑	0	3	
11	TYPE	TYPE	nominal	String	I056450DISPOSITION	☑	0	0	

☐ Display only visible fields

Now we can create the actual filter. So far we have created a column that indicates whether we want to include the entity (meaning person in this case) in our dataset. So we apply a filter on the column as we are interested only in clients that have no credit cards at the time of the time-stamp.

Click the “Filters” icon, then “New Condition”. You just need to double-click on the aggregate we have just created. You will find it on the right-hand side under “Variables”.



Defines a complex condition using the expression editor.

```
HasNoCreditCardAtTimeStamp_1Y100B_NOTEXST_0_CARD_ID
```

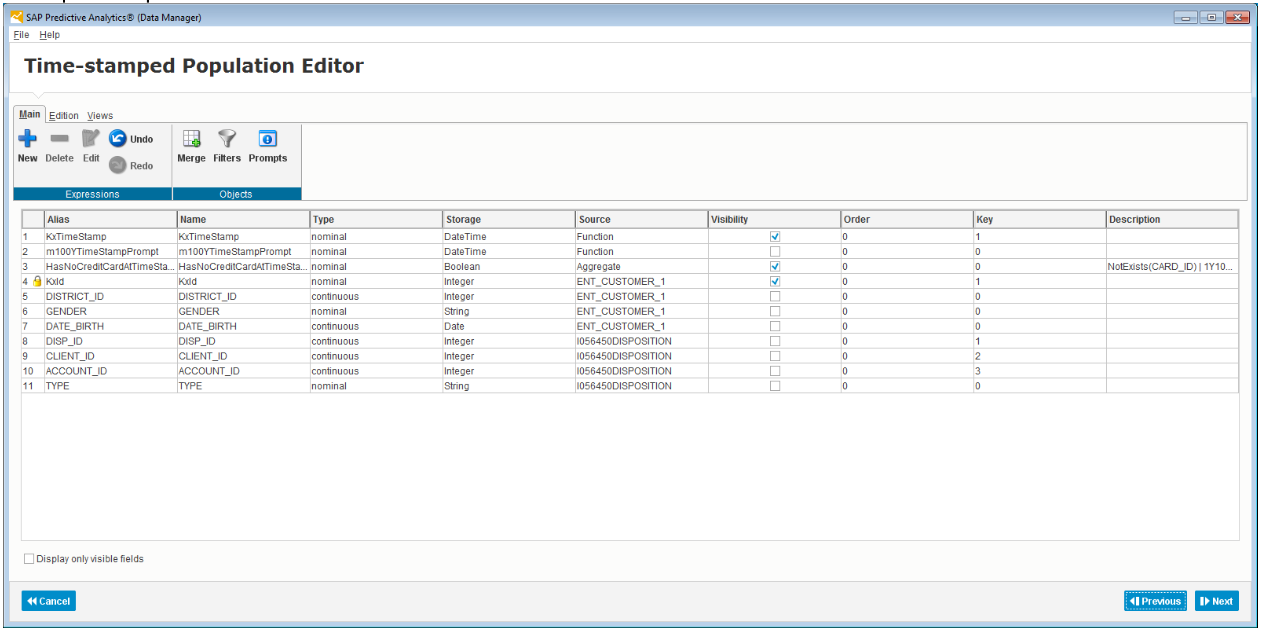
Field Association: Associate by Position

Messages

The expression is correct.

Hit “OK” → “Close” and the filter is active.

Now we need to define the target-variable. This is another aggregate. You should still be in the “Time-stamped Population Editor”.

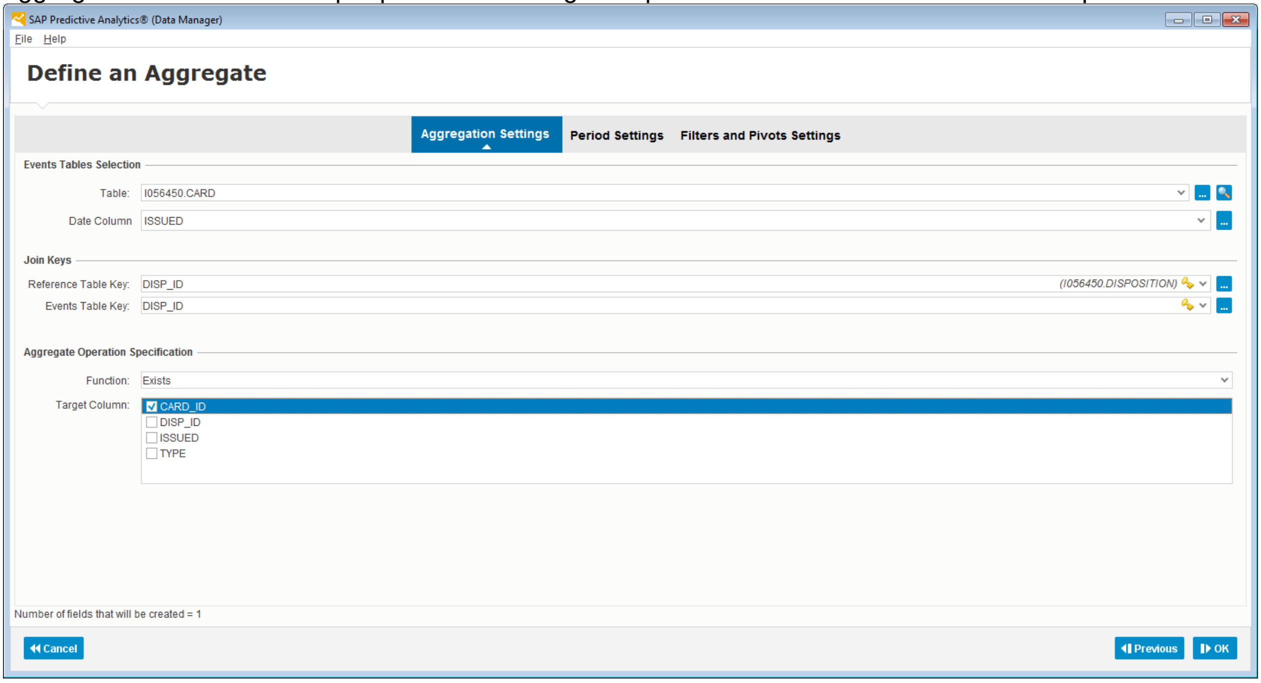


Time-stamped Population Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxTimeStamp	KxTimeStamp	nominal	DateTime	Function	☑	0	1	
2	m100YTimeStampPrompt	m100YTimeStampPrompt	nominal	DateTime	Function	☐	0	0	
3	HasNoCreditCardAtTimeSta...	HasNoCreditCardAtTimeSta...	nominal	Boolean	Aggregate	☑	0	0	NotExists(CARD_ID) | 1Y10...
4	KxId	KxId	nominal	Integer	ENT_CUSTOMER_1	☑	0	1	
5	DISTRICT_ID	DISTRICT_ID	continuous	Integer	ENT_CUSTOMER_1	☐	0	0	
6	GENDER	GENDER	nominal	String	ENT_CUSTOMER_1	☐	0	0	
7	DATE_BIRTH	DATE_BIRTH	continuous	Date	ENT_CUSTOMER_1	☐	0	0	
8	DISP_ID	DISP_ID	continuous	Integer	I056450DISPOSITION	☐	0	1	
9	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450DISPOSITION	☐	0	2	
10	ACCOUNT_ID	ACCOUNT_ID	continuous	Integer	I056450DISPOSITION	☐	0	3	
11	TYPE	TYPE	nominal	String	I056450DISPOSITION	☐	0	0	

☐ Display only visible fields

Click “New” → “New Aggregate” and use the following settings. Notice that we are now using “Exists” as aggregate. Now we look for people who have signed up for a credit card after the time stamp.


SAP Predictive Analytics® (Data Manager)
File Help
Aggregation Settings
Period Settings
Filters and Pivots Settings
Cancel
Previous
OK

Define an Aggregate

Events Tables Selection

Table: I056450.CARD

Date Column ISSUED

Join Keys

Reference Table Key: DISP_ID (I056450.DISPOSITION)

Events Table Key: DISP_ID

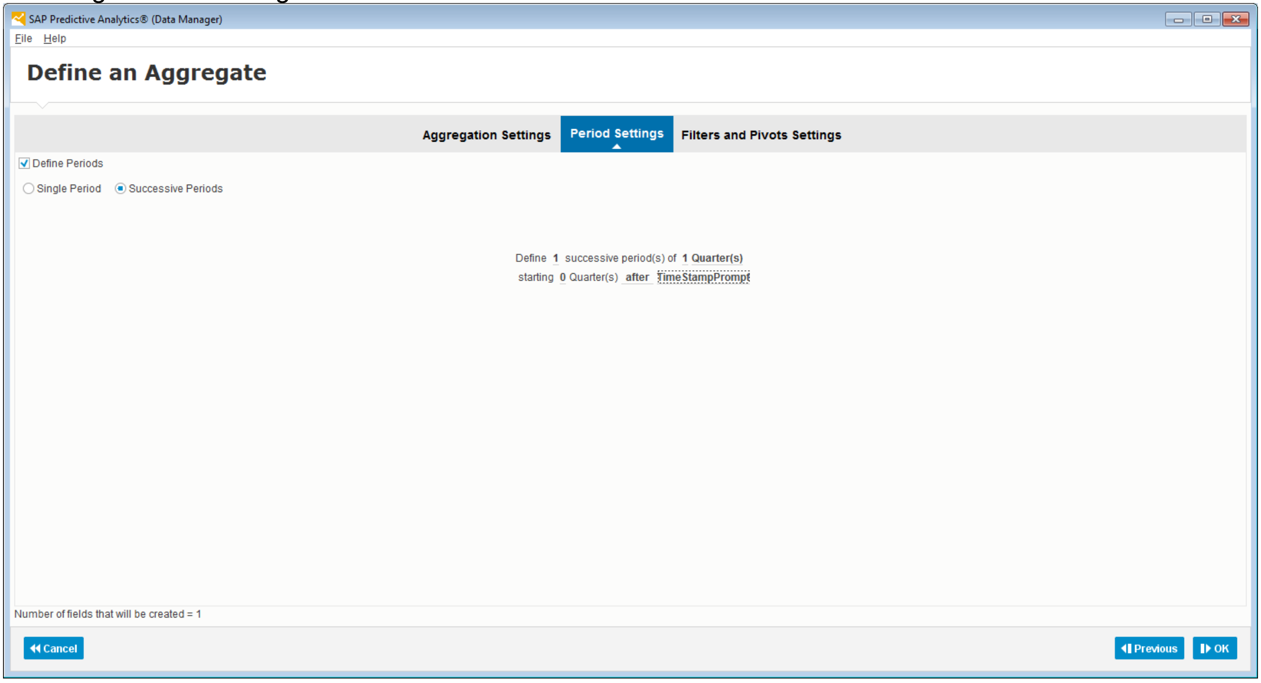
Aggregate Operation Specification

Function: Exists

Target Column:
☑ CARD_ID
☐ DISP_ID
☐ ISSUED
☐ TYPE

Number of fields that will be created = 1

As we want to identify those customers that bought the credit card in the quarter after the time stamp, set the following “Period Settings”.


SAP Predictive Analytics® (Data Manager)
File
Help
Aggregation Settings
Period Settings
Filters and Pivots Settings
Cancel
Previous
OK

Define an Aggregate

☑ Define Periods

○ Single Period ● Successive Periods

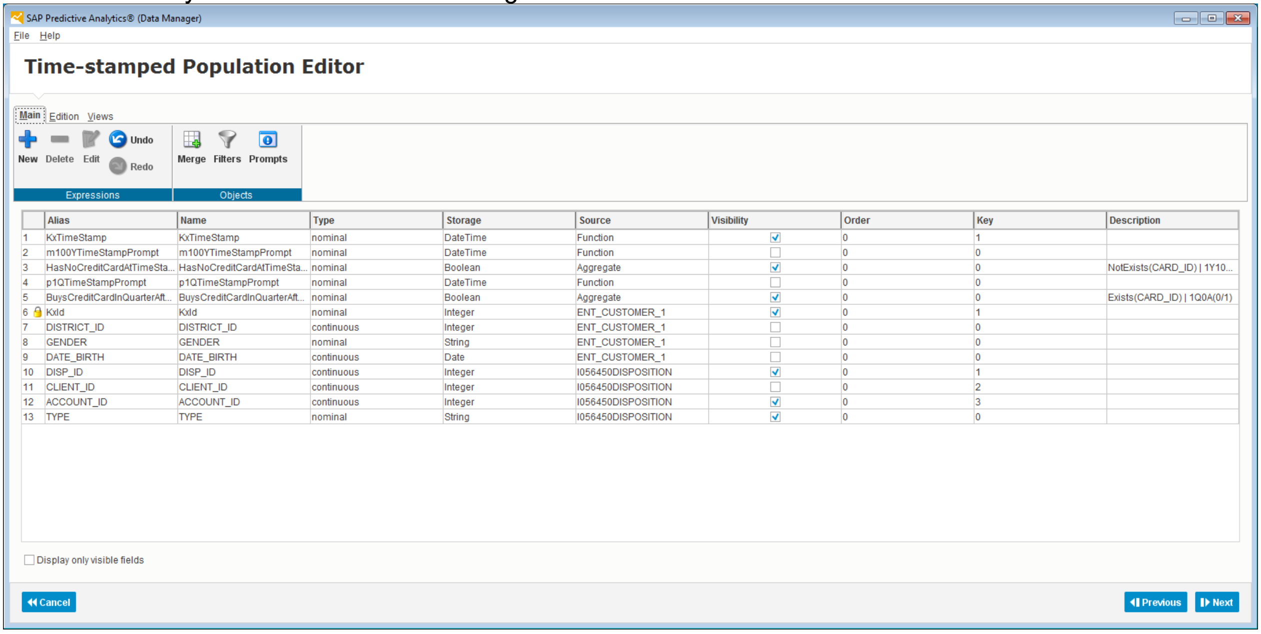
Define 1 successive period(s) of 1 Quarter(s)

starting 0 Quarter(s) after TimeStampPrompt

Number of fields that will be created = 1

Click “OK” and name the aggregate “BuysCreditCardInQuarterAfterTimeStamp”.

Click “OK” and you should see the following:

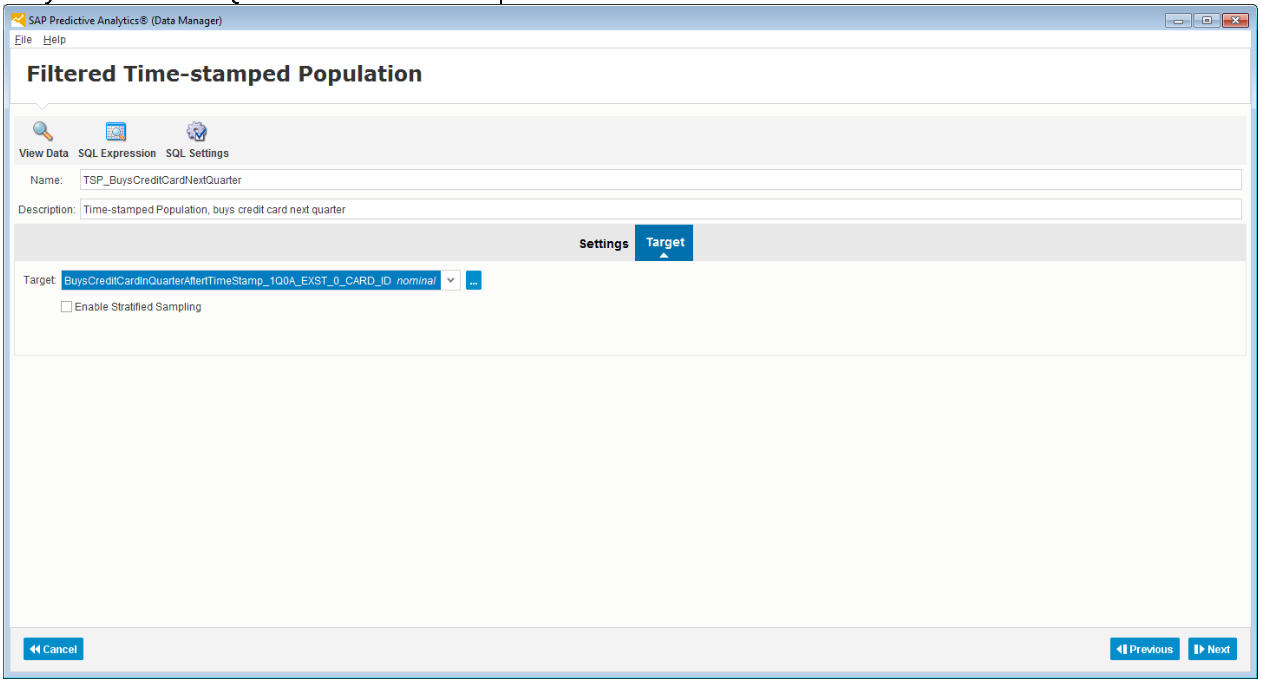


Time-stamped Population Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxTimeStamp	KxTimeStamp	nominal	DateTime	Function	☑	0	1	
2	m100YTimeStampPrompt	m100YTimeStampPrompt	nominal	DateTime	Function	□	0	0	
3	HasNoCreditCardAtTimeSta...	HasNoCreditCardAtTimeSta...	nominal	Boolean	Aggregate	☑	0	0	NotExists(CARD_ID) | 1Y10...
4	p1QTimeStampPrompt	p1QTimeStampPrompt	nominal	DateTime	Function	□	0	0	
5	BuysCreditCardInQuarterAft...	BuysCreditCardInQuarterAft...	nominal	Boolean	Aggregate	☑	0	0	Exists(CARD_ID) | 1Q0A(0/1)
6 🔒	KxId	KxId	nominal	Integer	ENT_CUSTOMER_1	☑	0	1	
7	DISTRICT_ID	DISTRICT_ID	continuous	Integer	ENT_CUSTOMER_1	□	0	0	
8	GENDER	GENDER	nominal	String	ENT_CUSTOMER_1	□	0	0	
9	DATE_BIRTH	DATE_BIRTH	continuous	Date	ENT_CUSTOMER_1	□	0	0	
10	DISP_ID	DISP_ID	continuous	Integer	I056450DISPOSITION	☑	0	1	
11	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450DISPOSITION	□	0	2	
12	ACCOUNT_ID	ACCOUNT_ID	continuous	Integer	I056450DISPOSITION	☑	0	3	
13	TYPE	TYPE	nominal	String	I056450DISPOSITION	☑	0	0	

□ Display only visible fields

Click “Next”, go to the “Target” tab and set the target to our new aggregate “BuysCreditCardInQuarterAftertTimeStamp”.



Filtered Time-stamped Population

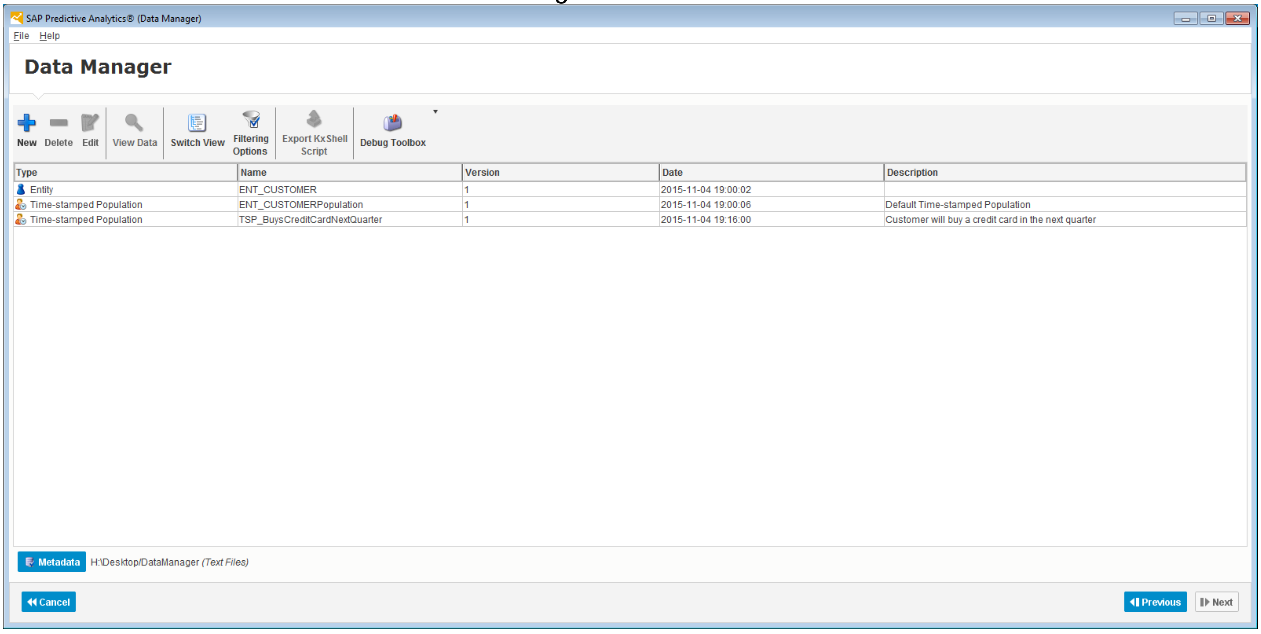
Name: TSP_BuysCreditCardNextQuarter

Description: Time-stamped Population, buys credit card next quarter

Target: BuysCreditCardInQuarterAftertTimeStamp_1Q0A_EXST_0_CARD_ID *nominal*

□ Enable Stratified Sampling

Click “Next” and we are back in the Data Manager.



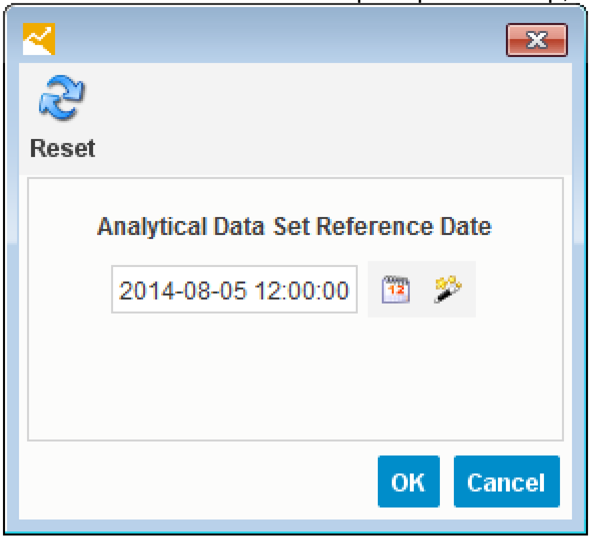
Data Manager

Type	Name	Version	Date	Description
Entity	ENT_CUSTOMER	1	2015-11-04 19:00:02	
Time-stamped Population	ENT_CUSTOMERPopulation	1	2015-11-04 19:00:06	Default Time-stamped Population
Time-stamped Population	TSP_BuysCreditCardNextQuarter	1	2015-11-04 19:16:00	Customer will buy a credit card in the next quarter

Metadata H:\Desktop\DataManager *(Text Files)*

Since we saved the time-stamped population under a new name, there are two different time-stamped populations. The default version cannot be deleted, so just ignore it.

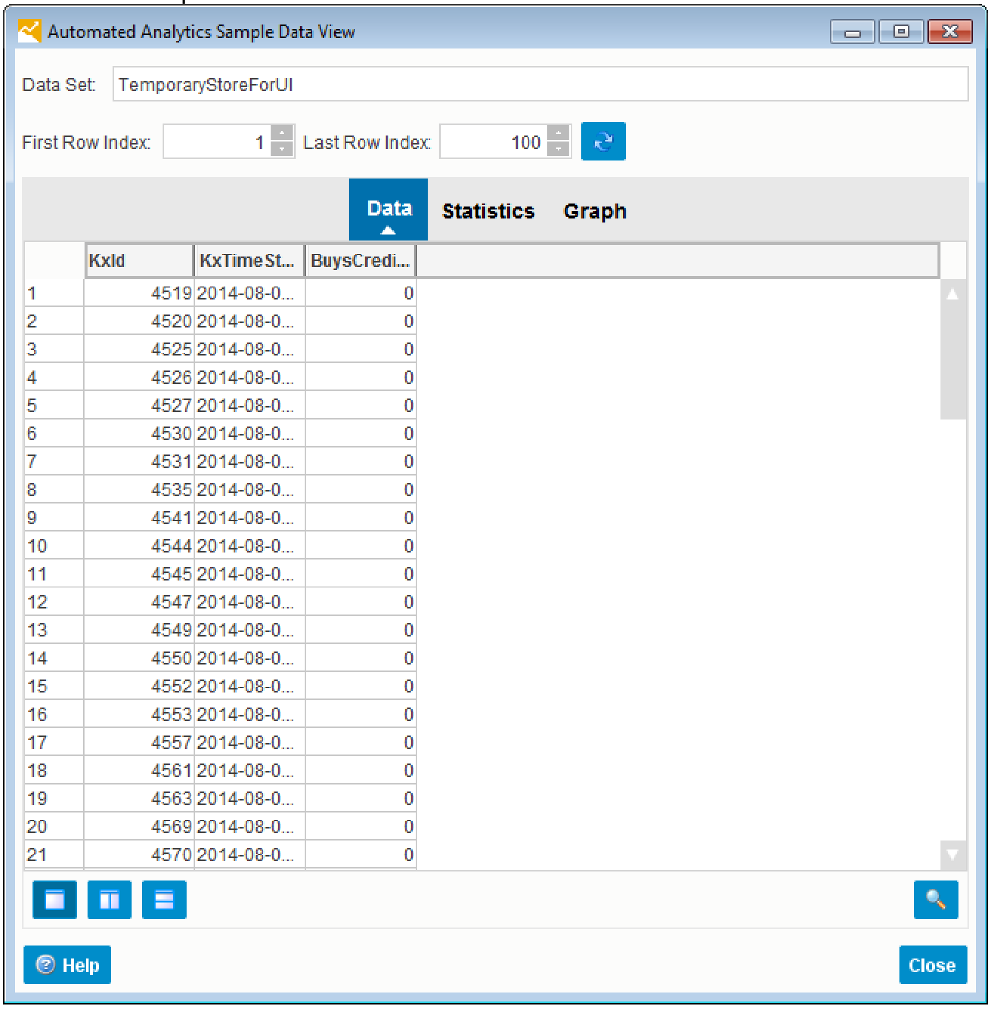
Before proceeding we quickly test our time-stamped population. Select “TSP_BuysCreditCardNextQuarter” and hit “View Data”. The time prompt comes up, enter the 5th of August 2014 and hit “OK”.



Analytical Data Set Reference Date

2014-08-05 12:00:00

You see a first preview of the data.



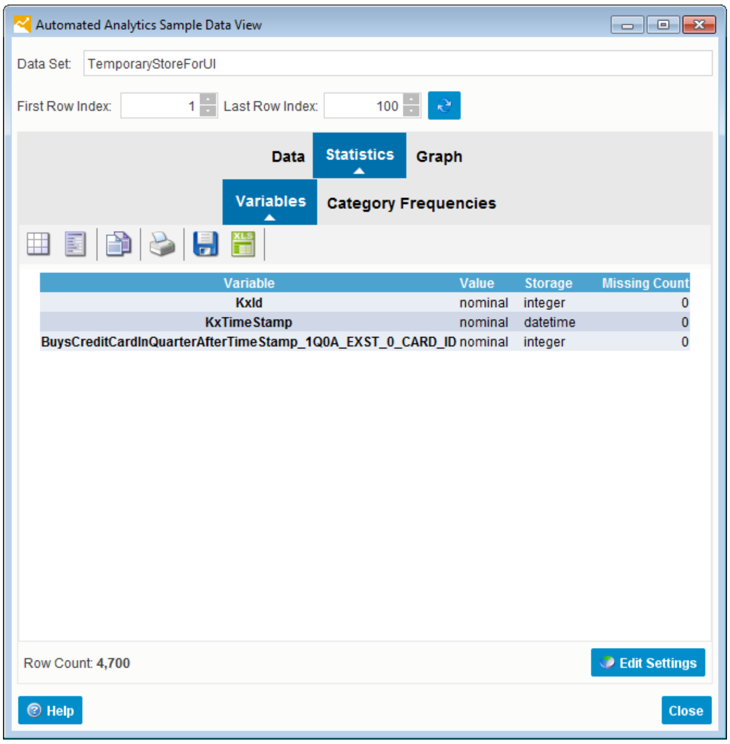
Data Set: TemporaryStoreForUI

First Row Index: 1 Last Row Index: 100

	KxId	KxTimeSt...	BuysCredi...
1	4519	2014-08-0...	0
2	4520	2014-08-0...	0
3	4525	2014-08-0...	0
4	4526	2014-08-0...	0
5	4527	2014-08-0...	0
6	4530	2014-08-0...	0
7	4531	2014-08-0...	0
8	4535	2014-08-0...	0
9	4541	2014-08-0...	0
10	4544	2014-08-0...	0
11	4545	2014-08-0...	0
12	4547	2014-08-0...	0
13	4549	2014-08-0...	0
14	4550	2014-08-0...	0
15	4552	2014-08-0...	0
16	4553	2014-08-0...	0
17	4557	2014-08-0...	0
18	4561	2014-08-0...	0
19	4563	2014-08-0...	0
20	4569	2014-08-0...	0
21	4570	2014-08-0...	0

We want to have a look at the target variable. How many positive cases do we have in the historic quarter after the timestamp?

Select the “Statistics” tab. Click “Compute statistics over the whole dataset”. This analyses the whole dataset.



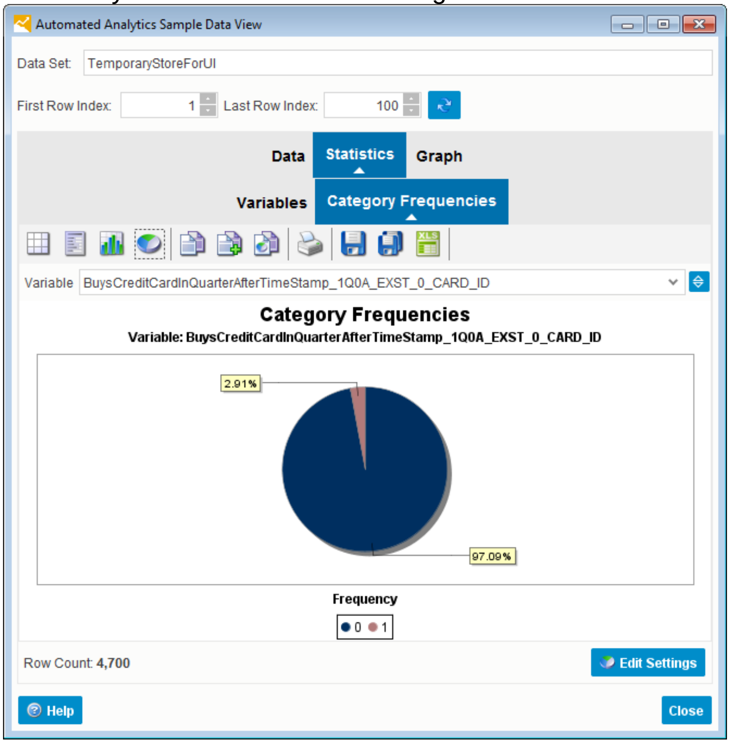
Data Set: TemporaryStoreForUI

First Row Index: 1 Last Row Index: 100

Variable	Value	Storage	Missing Count
KxId	nominal	integer	0
KxTimeStamp	nominal	datetime	0
BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID	nominal	integer	0

Row Count: 4,700

Click the “Category Frequency” tab and select our target variable from the drop-down. Click the pie-chart icon and you should see the following distribution.



Automated Analytics Sample Data View
Data Set: TemporaryStoreForUI
First Row Index: 1
Last Row Index: 100
Data
Statistics
Graph
Variables
Category Frequencies
Variable BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID
Category Frequencies
Variable: BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID
2.91%
97.09%
Frequency
● 0 ● 1
Row Count: 4,700
Edit Settings
Help
Close

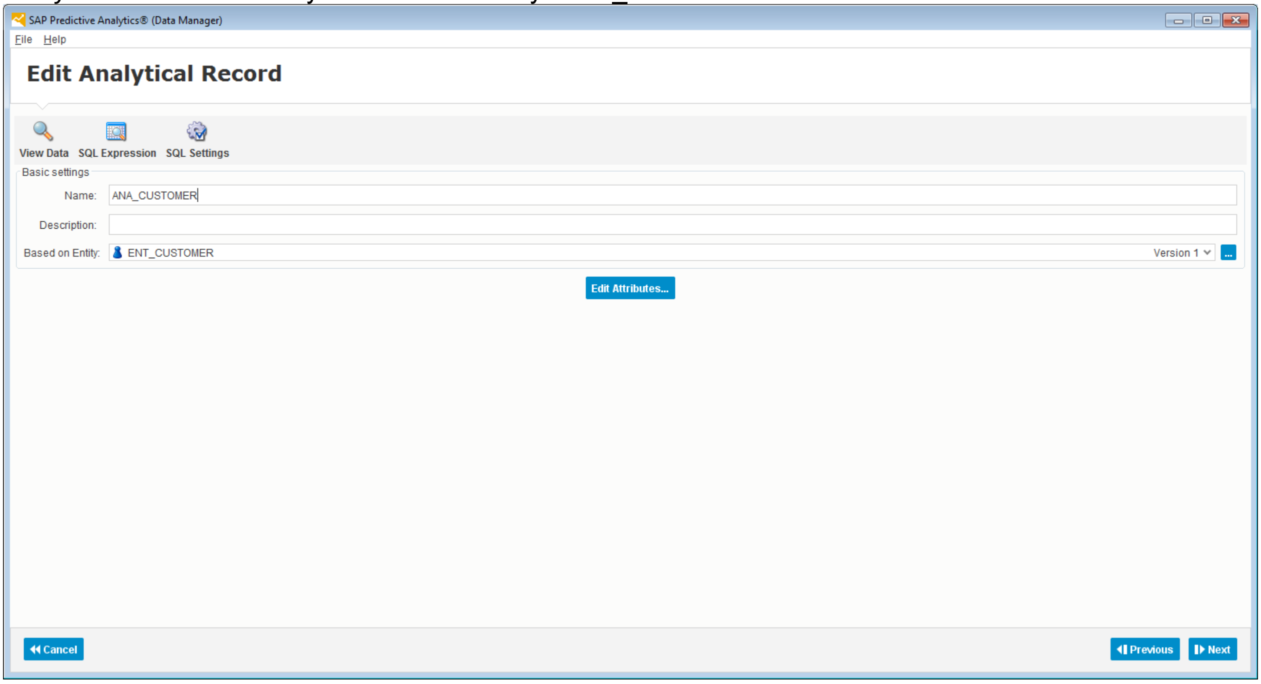
The dataset contains 4700 rows. Only 2.91% of these are positive cases, so 137 customer purchased their first credit card in the following quarter. Ideally we would like to have 1000 positive cases to train on. However, our data set if fairly small with under 5000 customers and we continue trying to create a strong and robust model. Should you encounter a similar situation with your own data, you could try to increase the size of the dataset or the length of the time frame the target variable is based on.

Click “Close”.

Step 3 – Add Analytical Record

Now we add the analytical record, which will describe our entity, the customer. Typically this is where you will spend most of the time, creating variables that describe your entity.

In “Data Manager” click “New” → “Analytical Record”. Give it the name “ANA_CUSTOMER”. Notice that the analytical record is already linked to our entity “ENT_CUSTOMER”.



Edit Analytical Record

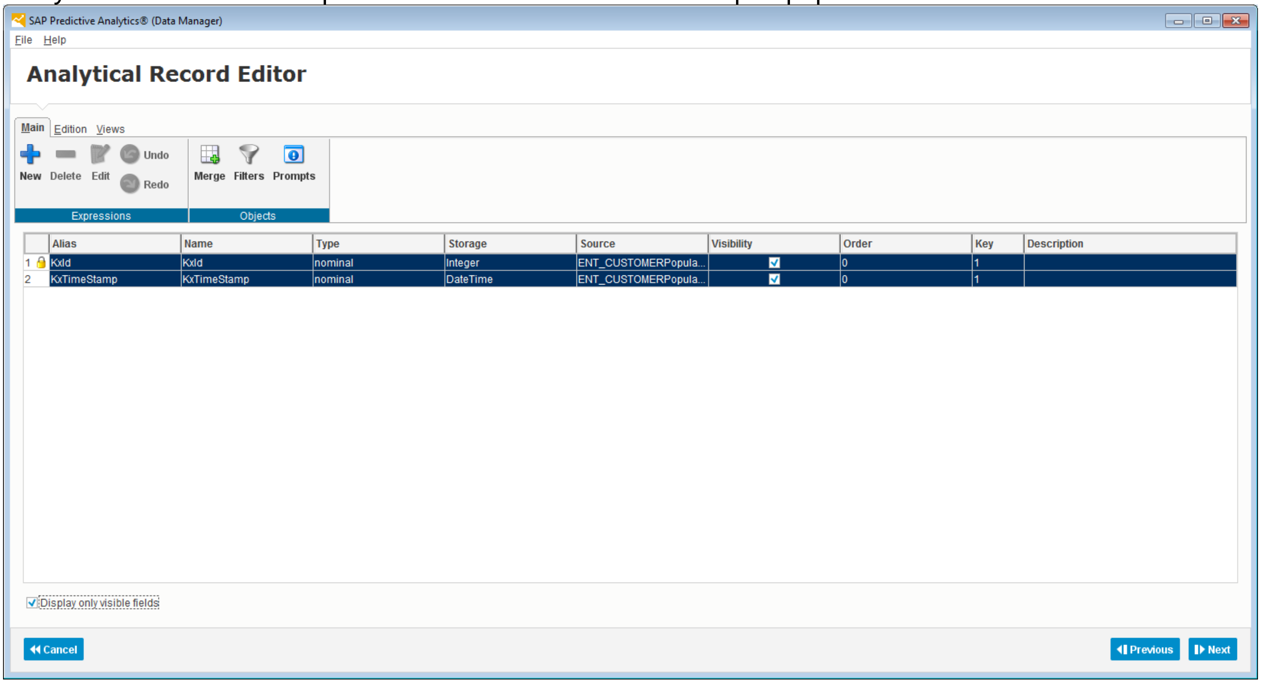
Basic settings

Name: ANA_CUSTOMER

Description:

Based on Entity: ENT_CUSTOMER Version 1

To add columns to the analytical record, hit “Edit Attributes”. So far the analytical record contains only the entity ID and the time-stamp. The columns from the time-stamped population are not listed.

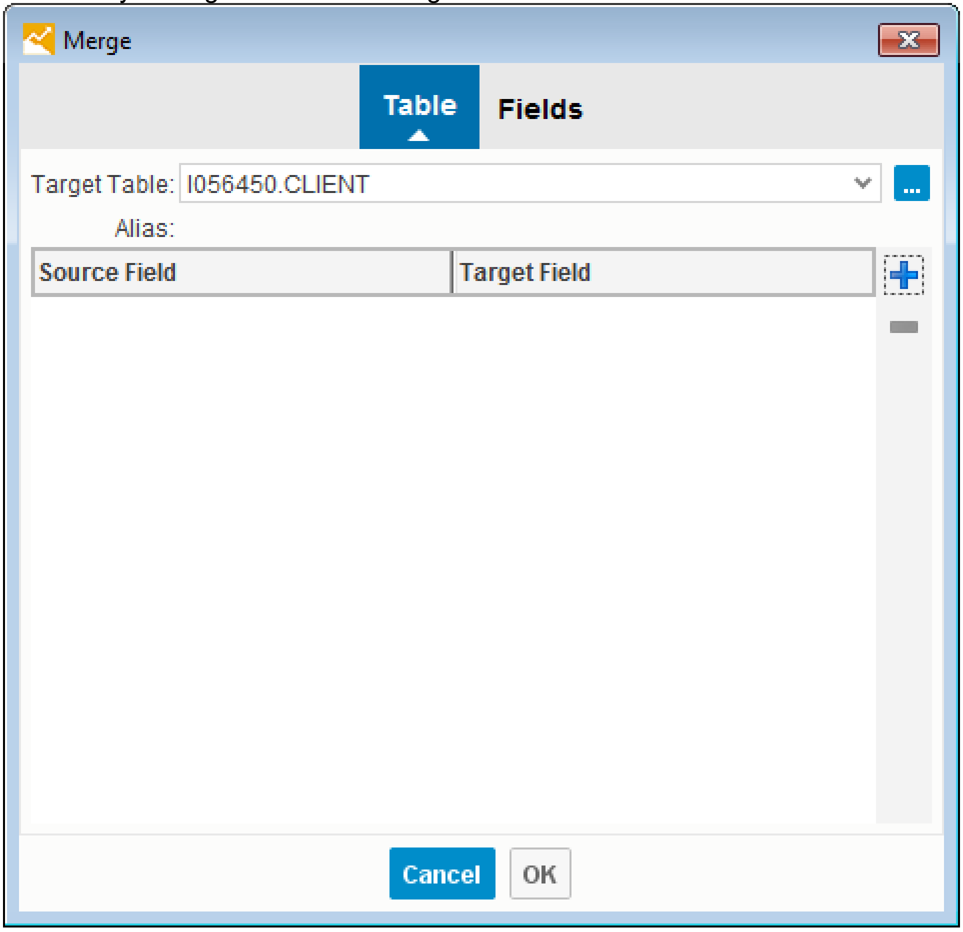


Analytical Record Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxId	KxId	nominal	Integer	ENT_CUSTOMERPopula...	☑	0	1	
2	KxTimeStamp	KxTimeStamp	nominal	DateTime	ENT_CUSTOMERPopula...	☑	0	1	

☑ Display only visible fields

As the entity does not yet have the additional columns from the “CLIENT” table, on which the entity is based, we start by adding these. Click “Merge” → “New”. Select the “CLIENT” table as done earlier.

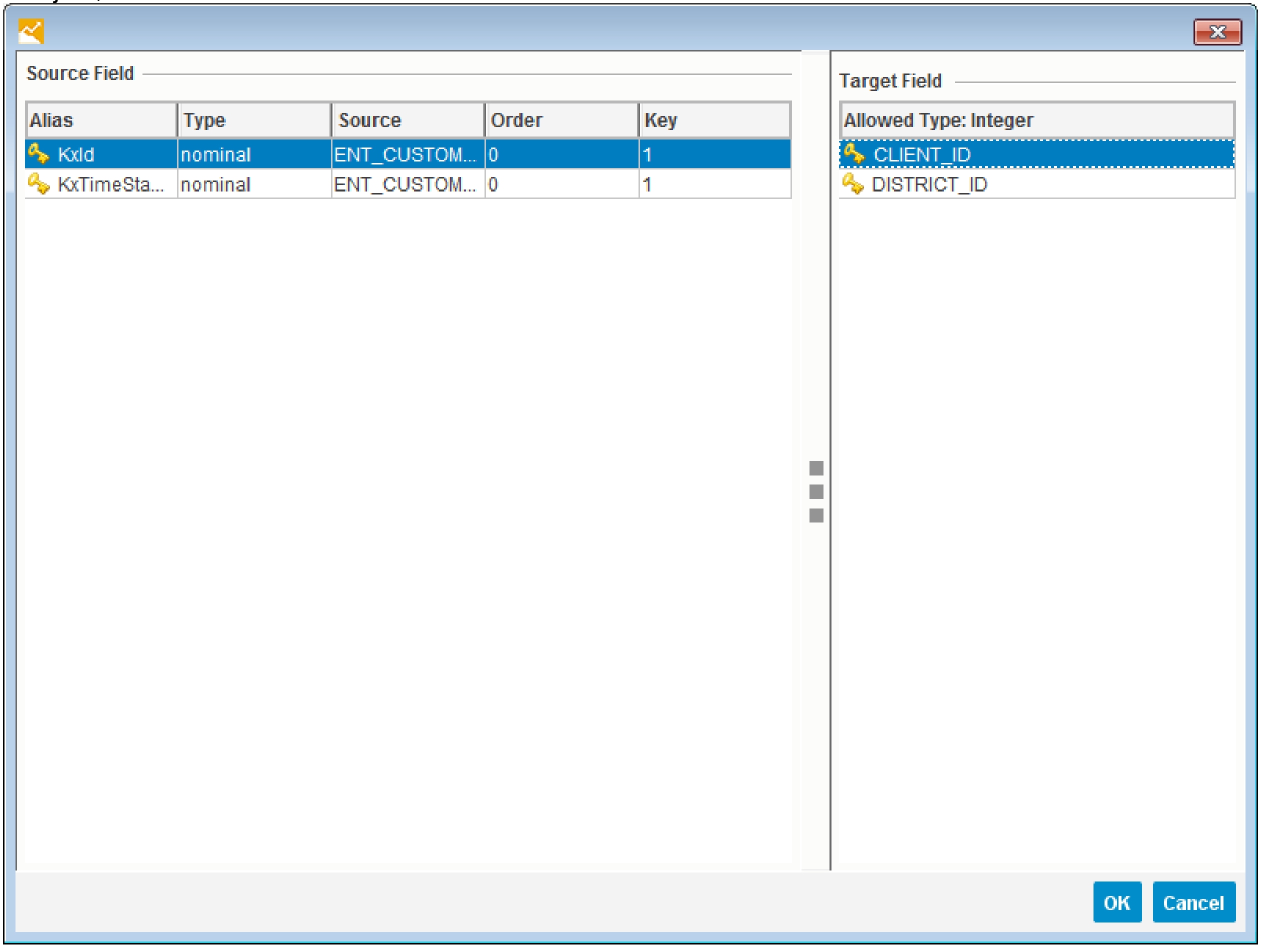


Target Table: I056450.CLIENT

Alias:

Source Field	Target Field

Click the plus-sign and join from the entity’s “KxId” column to the “CLIENT_ID” column. In effect, this is a self-join, as both columns are the same.



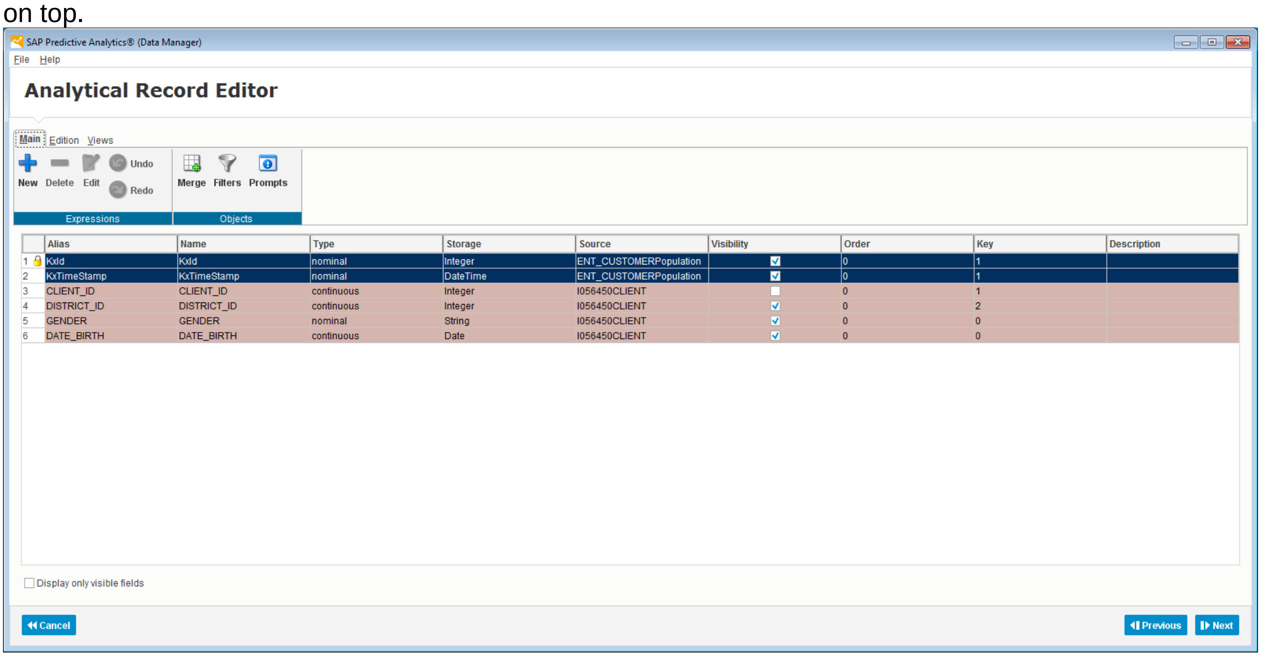
Source Field

Alias	Type	Source	Order	Key
KxId	nominal	ENT_CUSTOM...	0	1
KxTimeSta...	nominal	ENT_CUSTOM...	0	1

Target Field

Allowed Type: Integer
CLIENT_ID
DISTRICT_ID

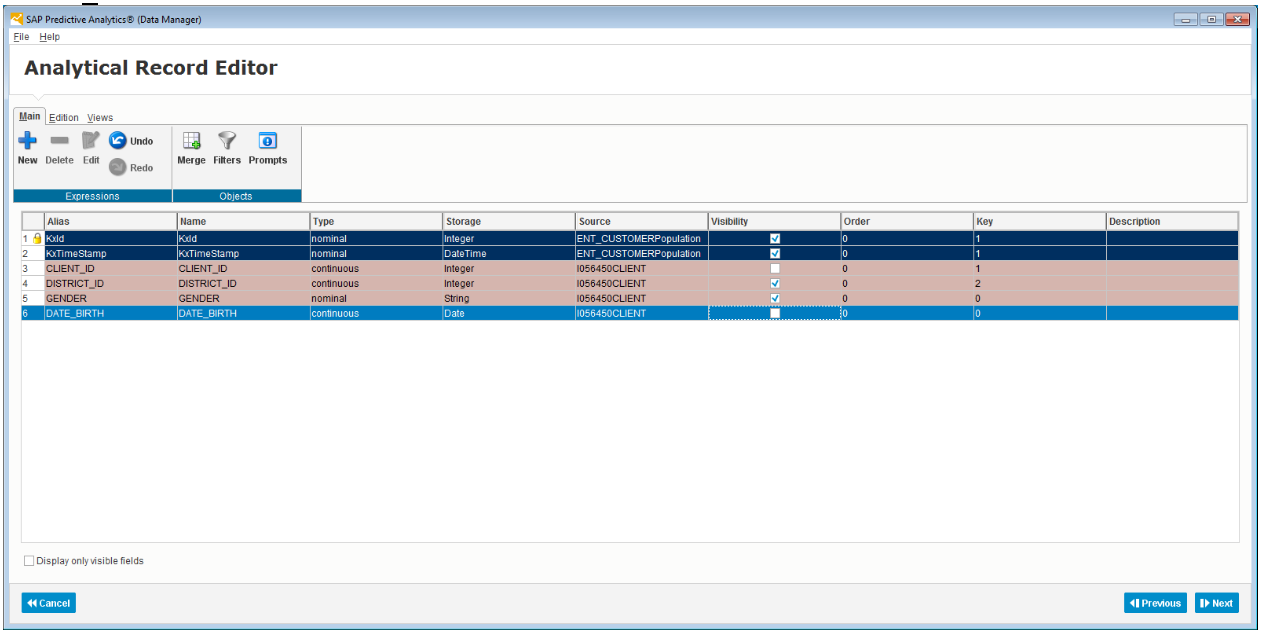
Hit “OK” → “OK” → “Close”. You now see the additional columns. They are shown in the same color to indicate they come from the same table. Should you not like the color, you can change it in the “Edition” tab on top.



	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxId	KxId	nominal	Integer	ENT_CUSTOMERPopulation	☑	0	1	
2	KxTimeStamp	KxTimeStamp	nominal	DateTime	ENT_CUSTOMERPopulation	☑	0	1	
3	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450CLIENT	□	0	1	
4	DISTRICT_ID	DISTRICT_ID	continuous	Integer	I056450CLIENT	☑	0	2	
5	GENDER	GENDER	nominal	String	I056450CLIENT	☑	0	0	
6	DATE_BIRTH	DATE_BIRTH	continuous	Date	I056450CLIENT	☑	0	0	

□ Display only visible fields

However, we should not use the date of birth. Instead we will later calculate the client's age at the given time-stamp. Make sure the option "Display only visible fields" is unticked and deselect the option "Visibility" for the "DATE_BIRTH" column.



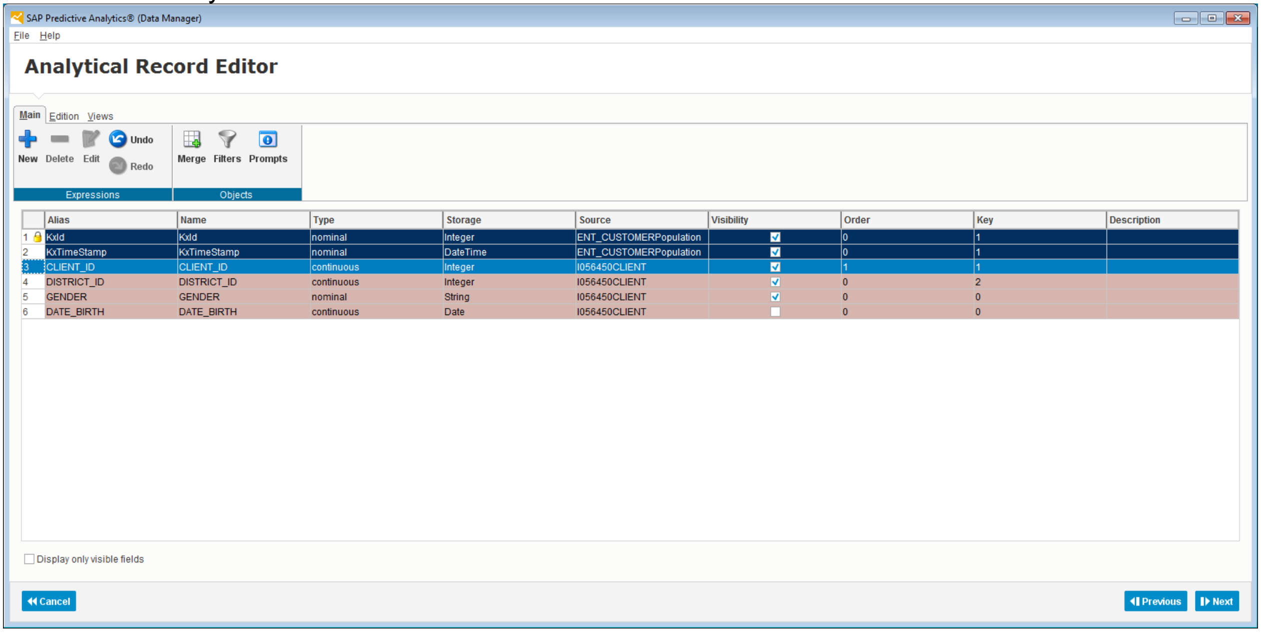
Analytical Record Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxId	KxId	nominal	Integer	ENT_CUSTOMERPopulation	☑	0	1	
2	KxTimeStamp	KxTimeStamp	nominal	DateTime	ENT_CUSTOMERPopulation	☑	0	1	
3	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450CLIENT	□	0	1	
4	DISTRICT_ID	DISTRICT_ID	continuous	Integer	I056450CLIENT	☑	0	2	
5	GENDER	GENDER	nominal	String	I056450CLIENT	☑	0	0	
6	DATE_BIRTH	DATE_BIRTH	continuous	Date	I056450CLIENT	□	0	0	

□ Display only visible fields

Now we need to instruct Data Manager to sort the data consistently. This is important to obtain reproducible models, which we will try to improve in multiple iterations. This is possible by instructing Data Manager to sort the data on a specific column. If not instructed, the database can return the data in different sequence every time we query it. This would change how data is split between the estimation and validation subsets, which as a result can change the model that is produced.

To obtain consistent results, enter the value 1 in the Order for the "Client_ID" column. Make sure the column's "Visibility" is ticked.



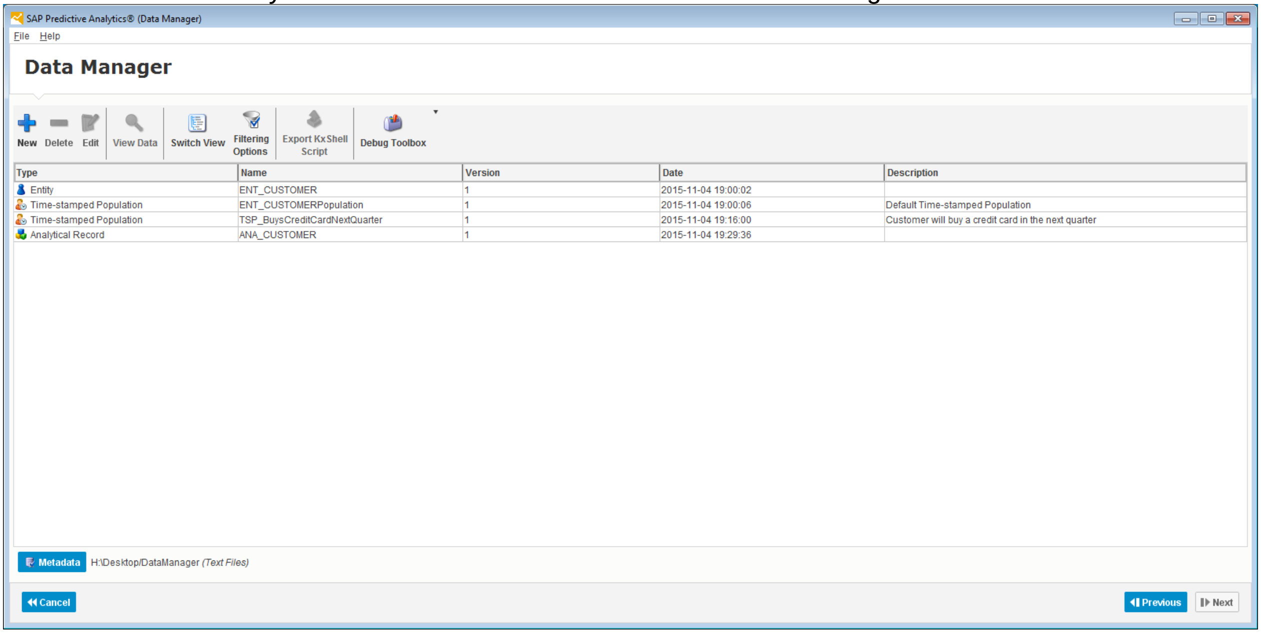
Analytical Record Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxId	KxId	nominal	Integer	ENT_CUSTOMERPopulation	☑	0	1	
2	KxTimeStamp	KxTimeStamp	nominal	DateTime	ENT_CUSTOMERPopulation	☑	0	1	
3	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450CLIENT	☑	1	1	
4	DISTRICT_ID	DISTRICT_ID	continuous	Integer	I056450CLIENT	☑	0	2	
5	GENDER	GENDER	nominal	String	I056450CLIENT	☑	0	0	
6	DATE_BIRTH	DATE_BIRTH	continuous	Date	I056450CLIENT	□	0	0	

□ Display only visible fields

Now we are ready to create a first predictive model. At this stage we are just testing that everything works well, we are not yet expecting a strong result. We will improve the model later by adding more content to the analytical record.

Click “Next” twice and you are back on the main screen of the Data Manager.

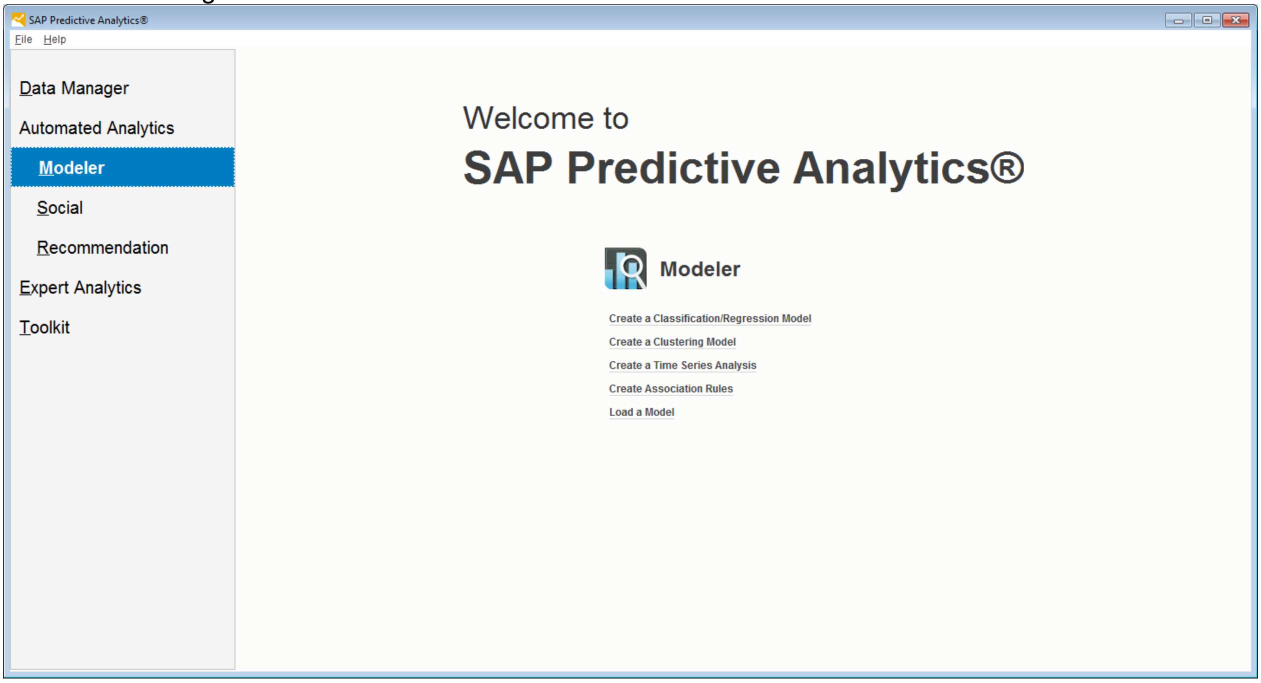


Type	Name	Version	Date	Description
Entity	ENT_CUSTOMER	1	2015-11-04 19:00:02	
Time-stamped Population	ENT_CUSTOMERPopulation	1	2015-11-04 19:00:06	Default Time-stamped Population
Time-stamped Population	TSP_BuysCreditCardNextQuarter	1	2015-11-04 19:16:00	Customer will buy a credit card in the next quarter
Analytical Record	ANA_CUSTOMER	1	2015-11-04 19:29:36	

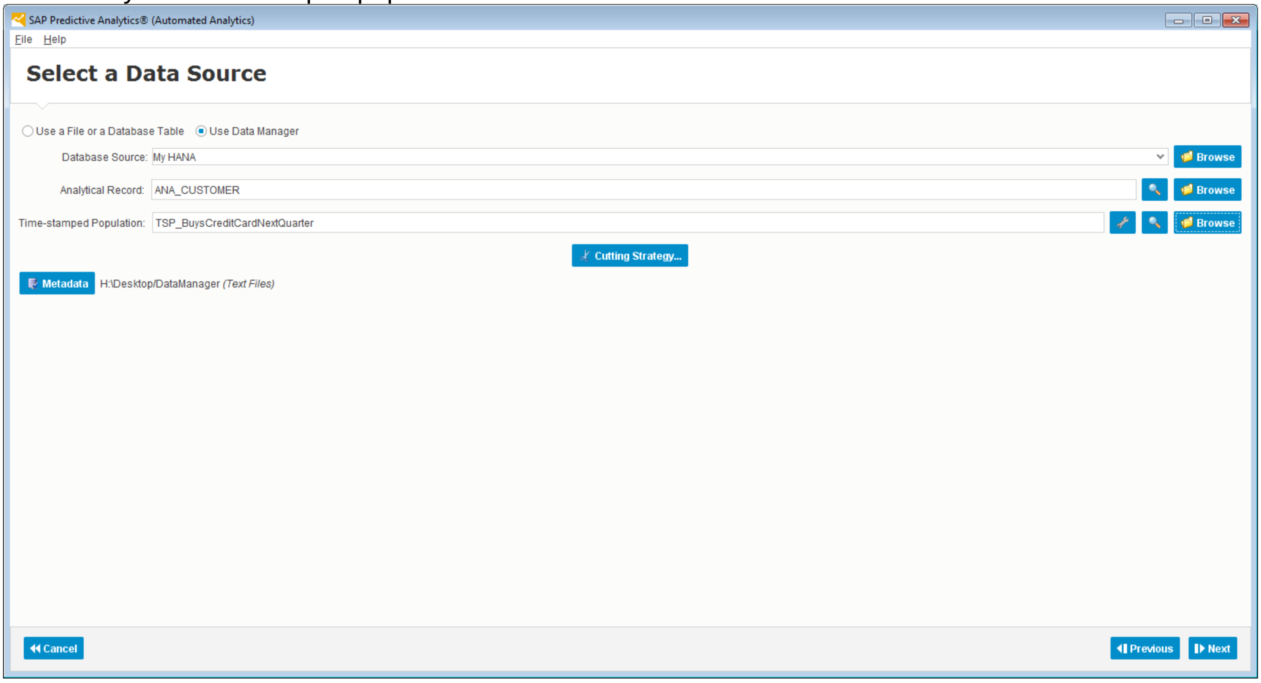
Step 4 – Create First Prediction to Test Data Model

Click “Cancel” to get to the main screen of SAP Predictive Analytics. This document assumes that you have created some classification model beforehand. Therefore these steps will be described a little shorter.

On the left hand side, select “Modeler” in the “Automated Analytics” section and click “Create Classification/Regression” in the center area.


SAP Predictive Analytics®
File Help
Data Manager
Automated Analytics
Modeler
Social
Recommendation
Expert Analytics
Toolkit
Welcome to
SAP Predictive Analytics®
Modeler
Create a Classification/Regression Model
Create a Clustering Model
Create a Time Series Analysis
Create Association Rules
Load a Model

Now select “Use Data Manager”. Choose the ODBC source, select the analytical record we have created followed by the time-stamped population.



○ Use a File or a Database Table ● Use Data Manager

Database Source: My HANA

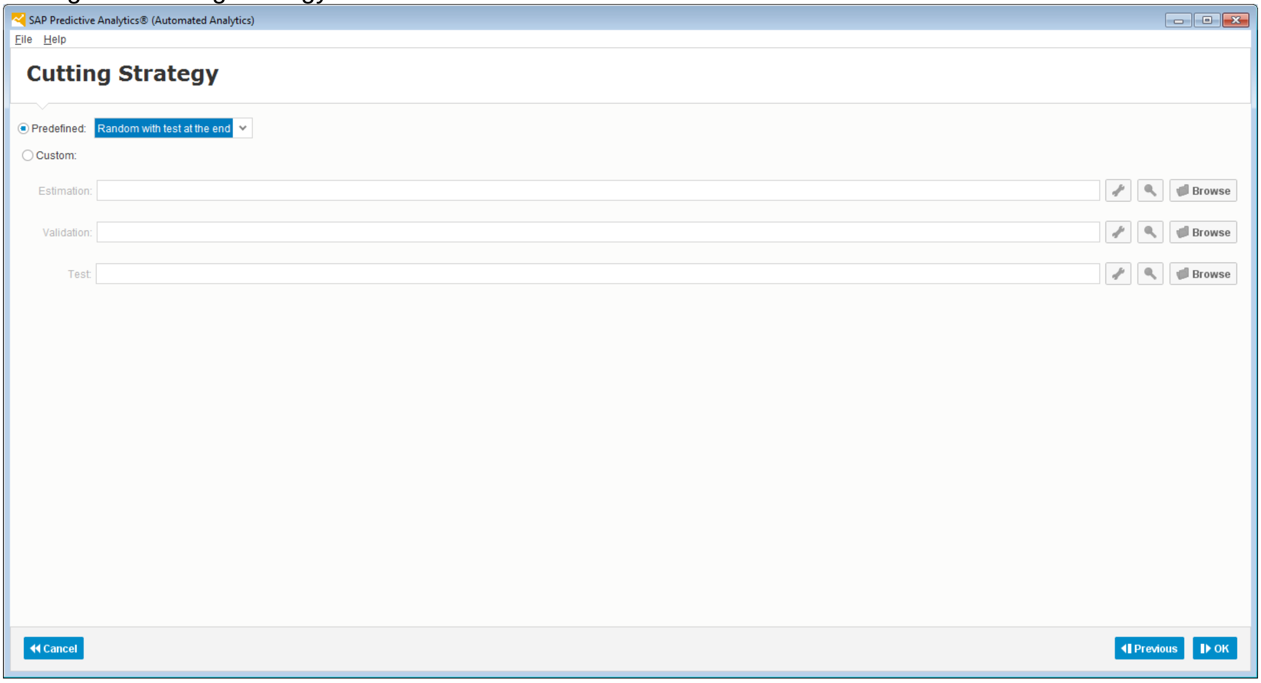
Analytical Record: ANA_CUSTOMER

Time-stamped Population: TSP_BuysCreditCardNextQuarter

Metadata H:\Desktop/DataManager *(Text Files)*

Eventually we want to test the model on previously unseen data. It is best to get in the habit of changing the cutting strategy, so that it sets some data aside to test the model's performance at the end on new data.

Change the cutting strategy to "Random with test at the end".



Cutting Strategy

● Predefined: Random with test at the end

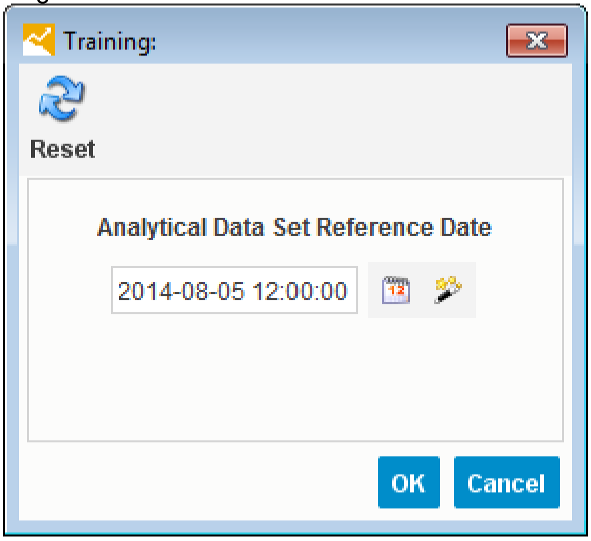
○ Custom:

Estimation:

Validation:

Test:

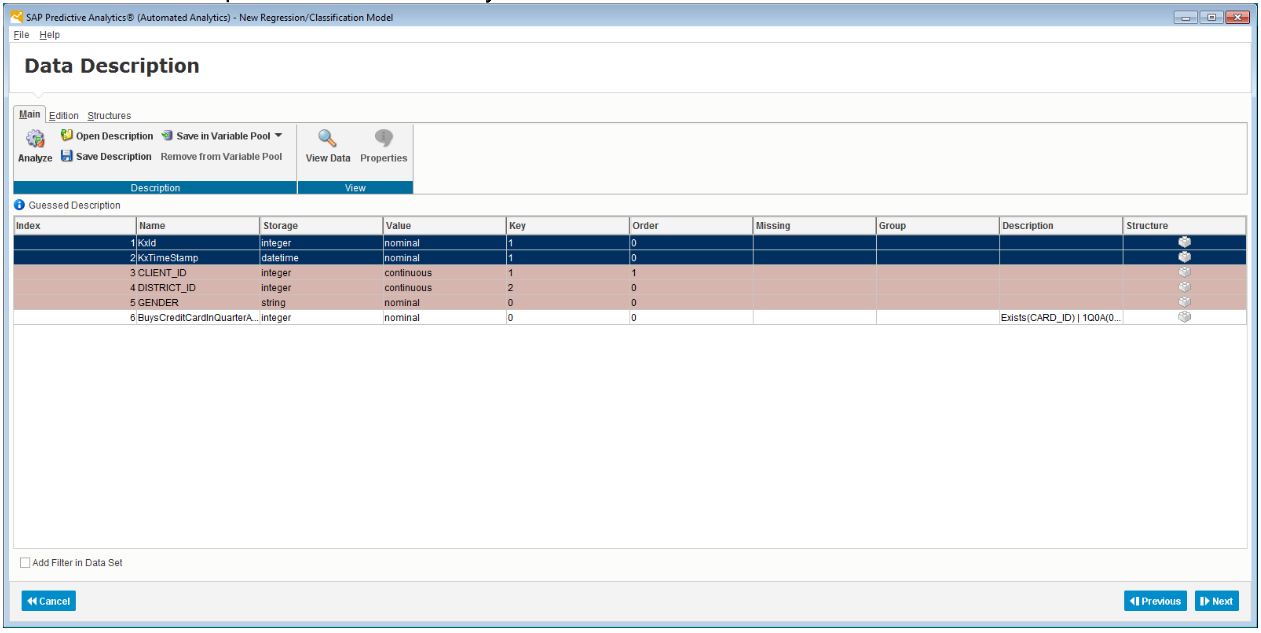
"OK" this and hit "Next". You will be prompted for the time-stamp. Select the same date as before, 5th of August 2014. Click "OK".



Analytical Data Set Reference Date

2014-08-05 12:00:00

In the “Data Description” screen hit “Analyze” to retrieve the data structure.

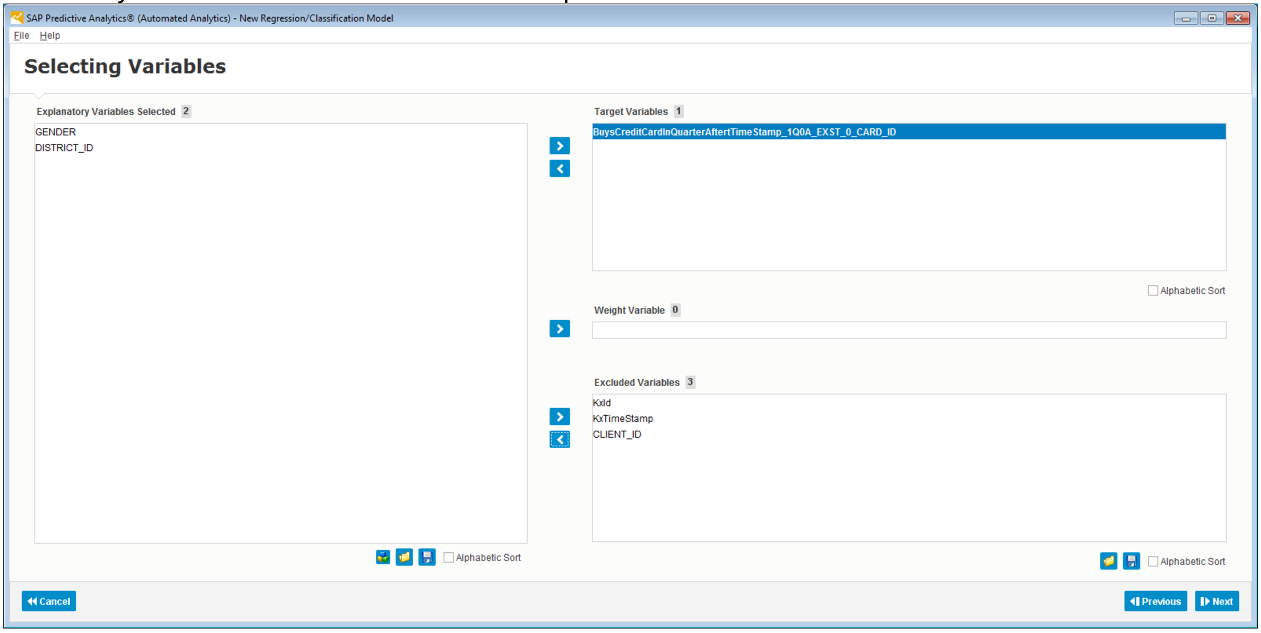


Data Description

Guessed Description

Index	Name	Storage	Value	Key	Order	Missing	Group	Description	Structure
1	KxId	integer	nominal	1	0				
2	KxTimeStamp	datetime	nominal	1	0				
3	CLIENT_ID	integer	continuous	1	1				
4	DISTRICT_ID	integer	continuous	2	0				
5	GENDER	string	nominal	0	0				
6	BuysCreditCardInQuarterA...	integer	nominal	0	0			Exists(CARD_ID) | 1Q0A(0...	

Click “Next” and chose “GENDER” and “DISTRICT_ID” as explanatory variables. Exclude “KxID”, “KxTimeStamp” and “CLIENT_ID” as they are not adding any value. The target variable should already be set to “BuysCreditCardInQuarterAfterTimeStamp”.


SAP Predictive Analytics® (Automated Analytics) - New Regression/Classification Model
File Help
□ Alphabetic Sort
□ Alphabetic Sort
□ Alphabetic Sort
Cancel
Previous
Next

Selecting Variables

Explanatory Variables Selected 2

GENDER
DISTRICT_ID

Target Variables 1

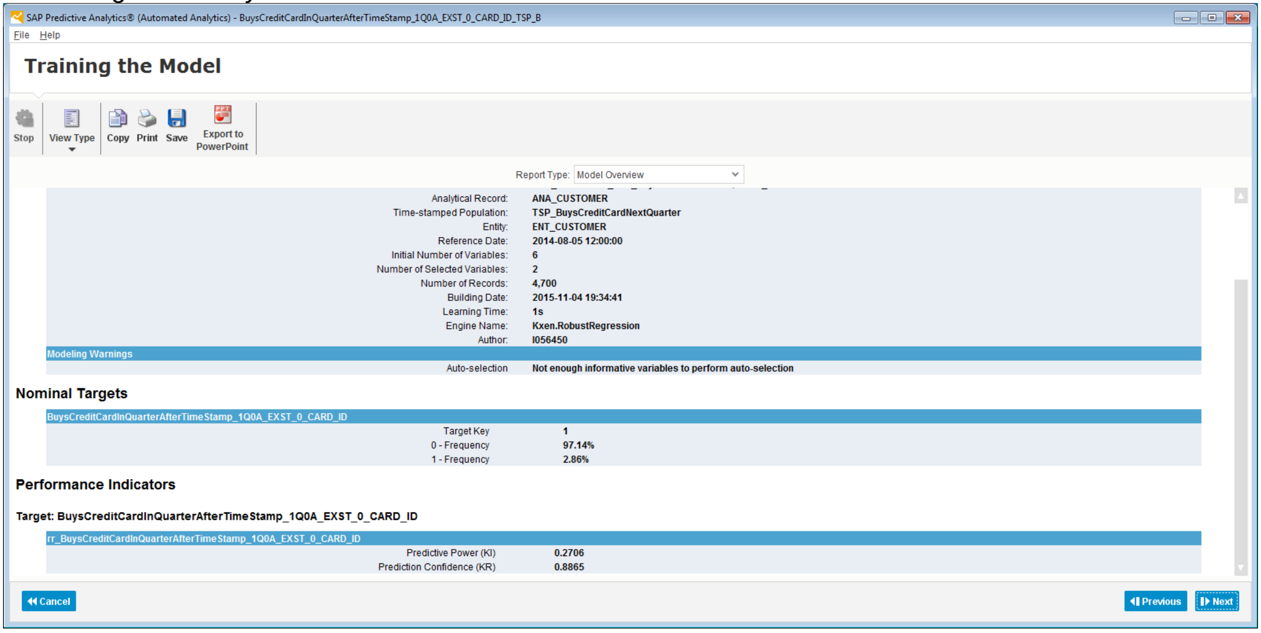
BuysCreditCardInQuarterAftertTimeStamp_1Q0A_EXST_0_CARD_ID

Weight Variable 0

Excluded Variables 3

KxId
KxTimeStamp
CLIENT_ID

Hit “Next” followed by “Generate” to create the predictive model. You should get a summary screen, confirming that a very first model has been created.


SAP Predictive Analytics® (Automated Analytics) - BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID_TSP_B
File Help
Stop
View Type
Copy
Print
Save
Export to PowerPoint
Cancel
Previous
Next

Training the Model

Report Type: Model Overview

Analytical Record:	ANA_CUSTOMER
Time-stamped Population:	TSP_BuysCreditCardNextQuarter
Entity:	ENT_CUSTOMER
Reference Date:	2014-08-05 12:00:00
Initial Number of Variables:	6
Number of Selected Variables:	2
Number of Records:	4,700
Building Date:	2015-11-04 19:34:41
Learning Time:	1s
Engine Name:	Kxen.RobustRegression
Author:	I056450

Modeling Warnings

Auto-selection	Not enough informative variables to perform auto-selection

Nominal Targets

BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID

Target Key	1
0 - Frequency	97.14%
1 - Frequency	2.86%

Performance Indicators

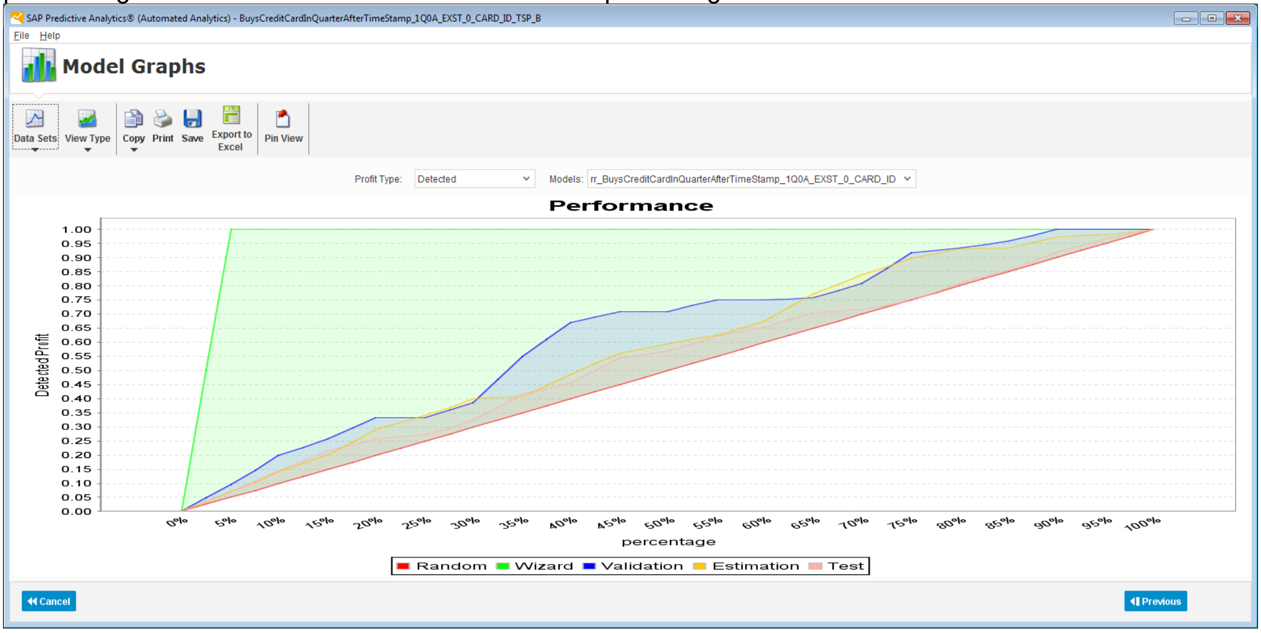
Target: BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID

rr_BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID

Predictive Power (KI)	0.2706
Prediction Confidence (KR)	0.8865

The test has been successful in that a very basic model was created without any errors. Note the Predictive Power of 0.2706 and the Predictive Confidence of 0.8865.

If you would like to see the model’s ability graphically, click on “Next” → “Model Graphs”. To see the model’s performance on all data sets, click the “Data Sets” icon and tick “All Data Sets”. Now you also see the model performing on data that has not been used at all in producing the model.



Profit Type: Detected

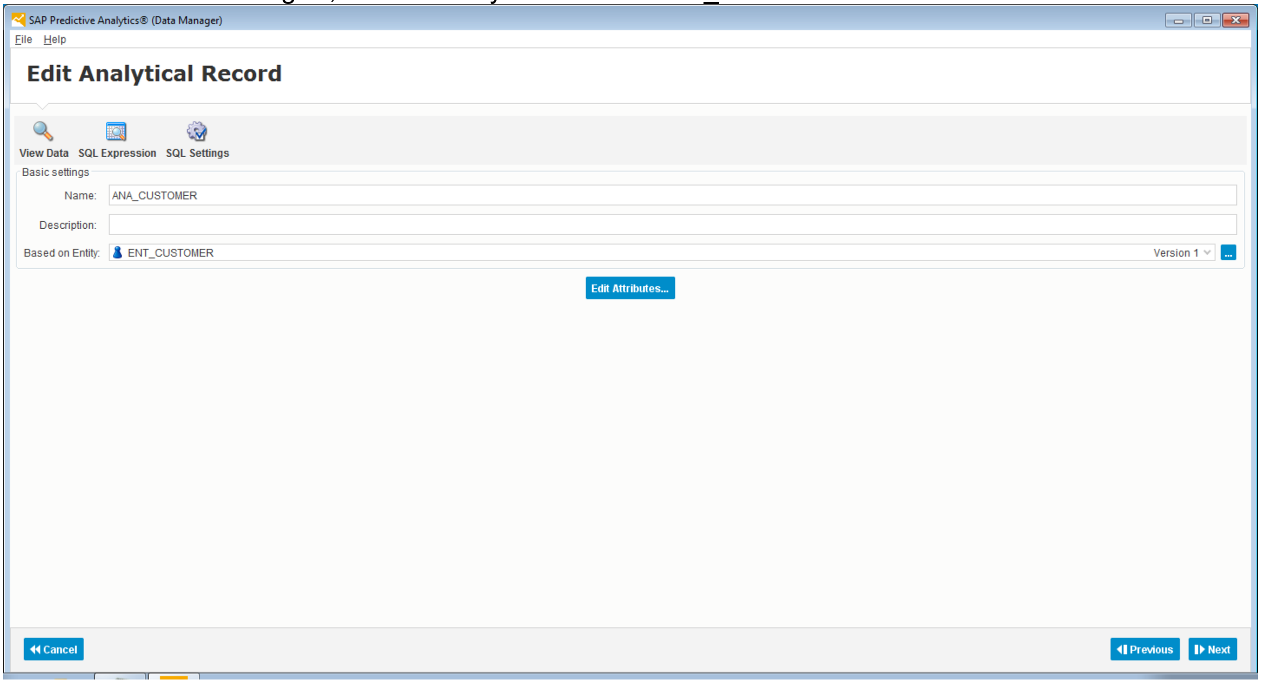
Models: rr_BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID

You can now “Cancel” this and go to the “Home Panel” to continue enhancing the analytical record.

Step 5 – Add static columns to the Analytical Record

Now we continue adding columns to the analytical record to better describe our entity, the customer, and its behaviour. At any time, feel free to test your data model with a classification model.

Back in the “Data Manager”, edit the analytical record “ANA_CUSTOMER”.



Edit Analytical Record

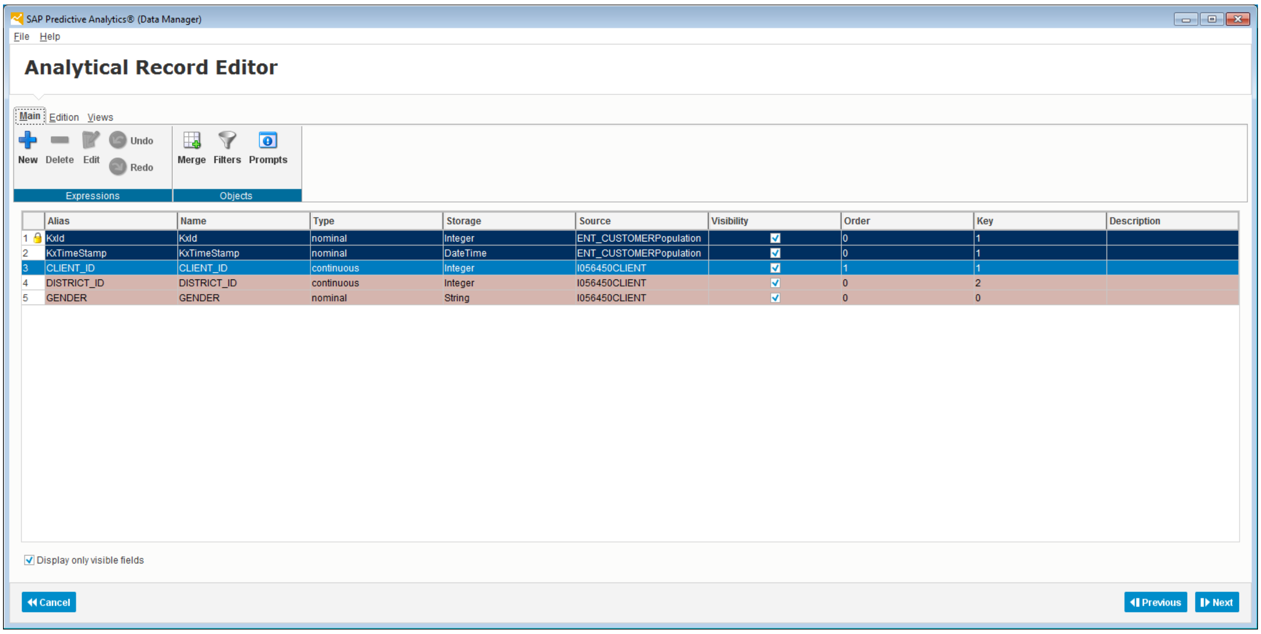
Basic settings

Name: ANA_CUSTOMER

Description:

Based on Entity: ENT_CUSTOMER Version 1

Click “Edit Attributes”.

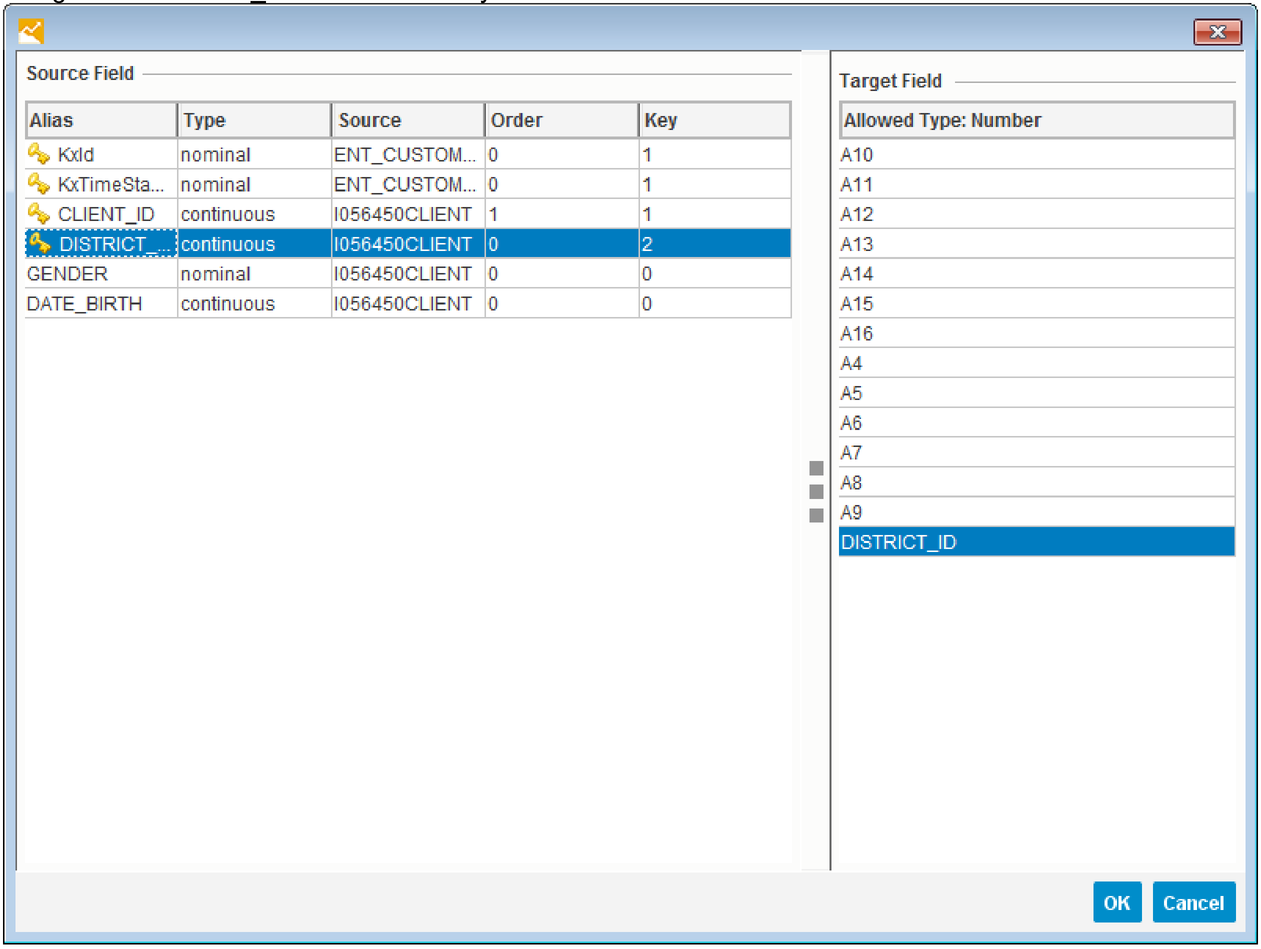


Analytical Record Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxId	KxId	nominal	Integer	ENT_CUSTOMERPopulation	☑	0	1	
2	KxTimeStamp	KxTimeStamp	nominal	DateTime	ENT_CUSTOMERPopulation	☑	0	1	
3	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450CLIENT	☑	1	1	
4	DISTRICT_ID	DISTRICT_ID	continuous	Integer	I056450CLIENT	☑	0	2	
5	GENDER	GENDER	nominal	String	I056450CLIENT	☑	0	0	

☑ Display only visible fields

Now we add the data from the “DISTRICT” table. Select “Merge” “→ “New”. Join in the “DISTRICT” table using the “DISTRICT_ID” columns as keys.



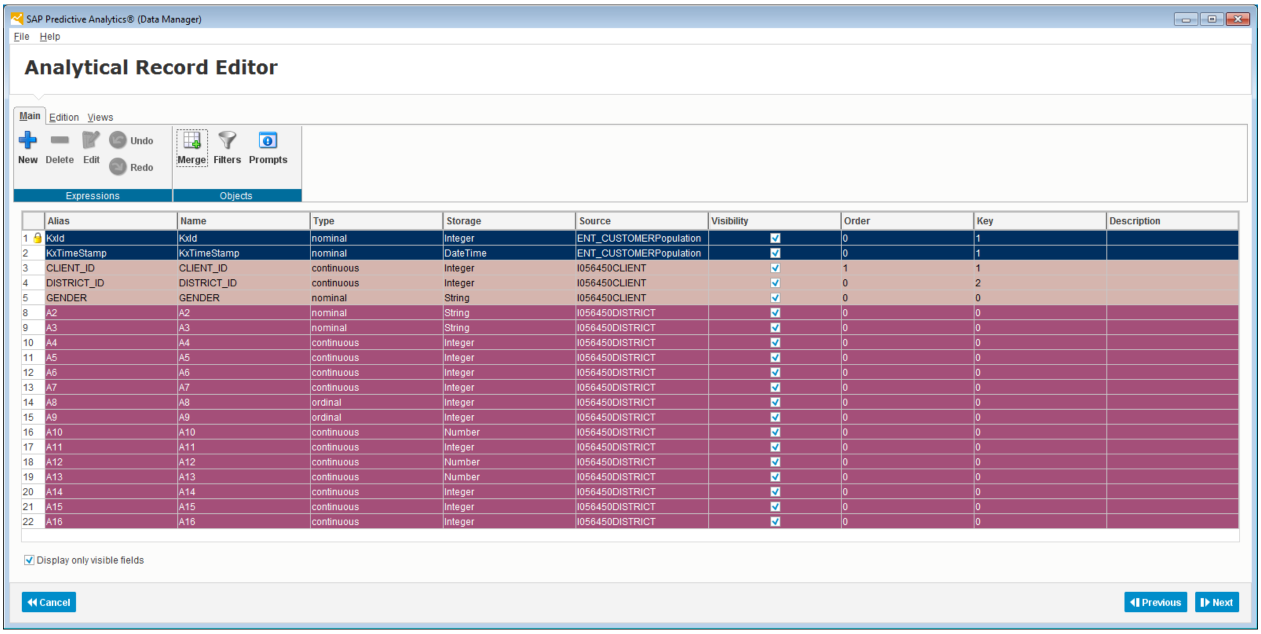
Source Field

Alias	Type	Source	Order	Key
KxId	nominal	ENT_CUSTOM...	0	1
KxTimeSta...	nominal	ENT_CUSTOM...	0	1
CLIENT_ID	continuous	I056450CLIENT	1	1
DISTRICT_...	continuous	I056450CLIENT	0	2
GENDER	nominal	I056450CLIENT	0	0
DATE_BIRTH	continuous	I056450CLIENT	0	0

Target Field

Allowed Type: Number
A10
A11
A12
A13
A14
A15
A16
A4
A5
A6
A7
A8
A9
DISTRICT_ID

Click “OK” → “OK” → “Close” and the additional columns are listed.



Analytical Record Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	KxId	KxId	nominal	Integer	ENT_CUSTOMERPopulation	☑	0	1	
2	KxTimeStamp	KxTimeStamp	nominal	DateTime	ENT_CUSTOMERPopulation	☑	0	1	
3	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450CLIENT	☑	1	1	
4	DISTRICT_ID	DISTRICT_ID	continuous	Integer	I056450CLIENT	☑	0	2	
5	GENDER	GENDER	nominal	String	I056450CLIENT	☑	0	0	
8	A2	A2	nominal	String	I056450DISTRICT	☑	0	0	
9	A3	A3	nominal	String	I056450DISTRICT	☑	0	0	
10	A4	A4	continuous	Integer	I056450DISTRICT	☑	0	0	
11	A5	A5	continuous	Integer	I056450DISTRICT	☑	0	0	
12	A6	A6	continuous	Integer	I056450DISTRICT	☑	0	0	
13	A7	A7	continuous	Integer	I056450DISTRICT	☑	0	0	
14	A8	A8	ordinal	Integer	I056450DISTRICT	☑	0	0	
15	A9	A9	ordinal	Integer	I056450DISTRICT	☑	0	0	
16	A10	A10	continuous	Number	I056450DISTRICT	☑	0	0	
17	A11	A11	continuous	Integer	I056450DISTRICT	☑	0	0	
18	A12	A12	continuous	Number	I056450DISTRICT	☑	0	0	
19	A13	A13	continuous	Number	I056450DISTRICT	☑	0	0	
20	A14	A14	continuous	Integer	I056450DISTRICT	☑	0	0	
21	A15	A15	continuous	Integer	I056450DISTRICT	☑	0	0	
22	A16	A16	continuous	Integer	I056450DISTRICT	☑	0	0	

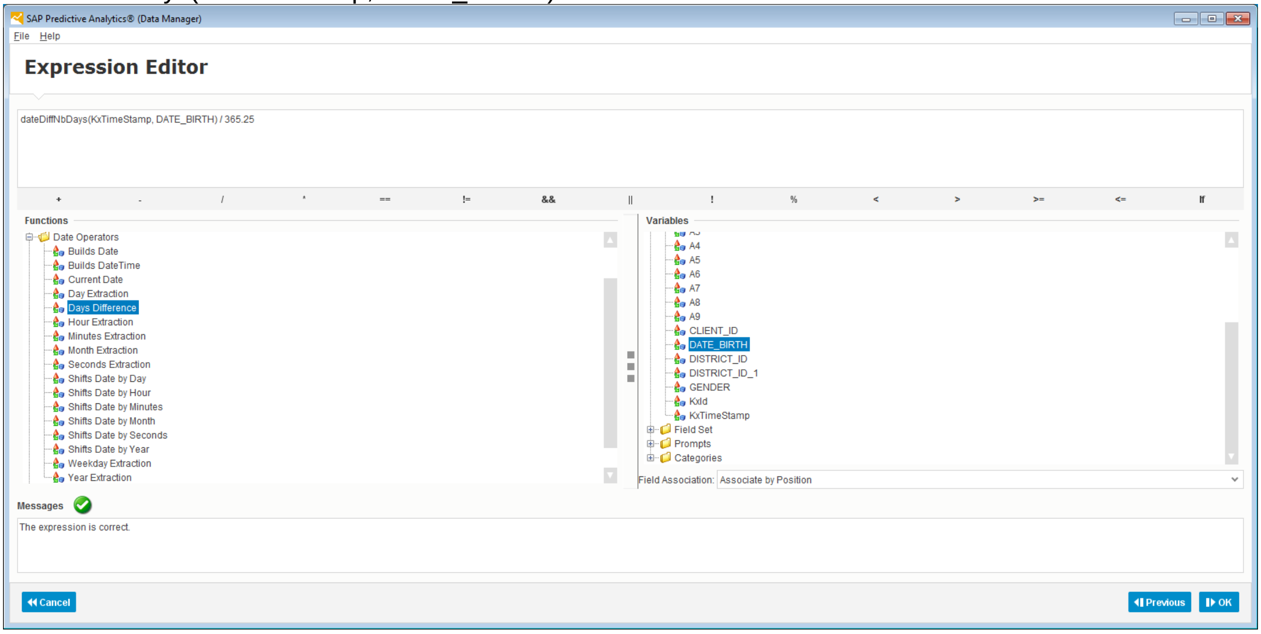
☑ Display only visible fields

Feel free to test the data model with a new Classification model if you like. When done, continue on the next page.

Step 6 – Add calculated column to the Analytical Record

Earlier on we excluded the client's birthdate. Now we use this information to calculate the person's age at the moment of the time-stamp.

Click "New" → "Expression Editor". Construct the following formula:
dateDiffNbDays(KxTimeStamp, DATE_BIRTH) / 365.25

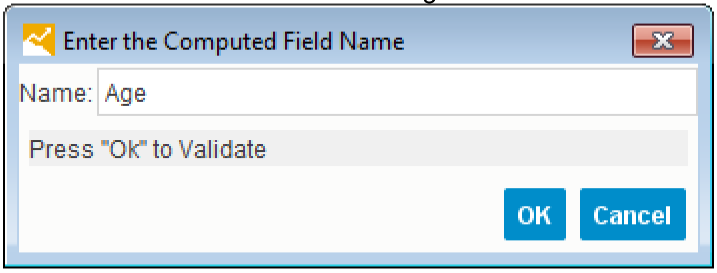


dateDiffNbDays(KxTimeStamp, DATE_BIRTH) / 365.25

Field Association: Associate by Position

The expression is correct.

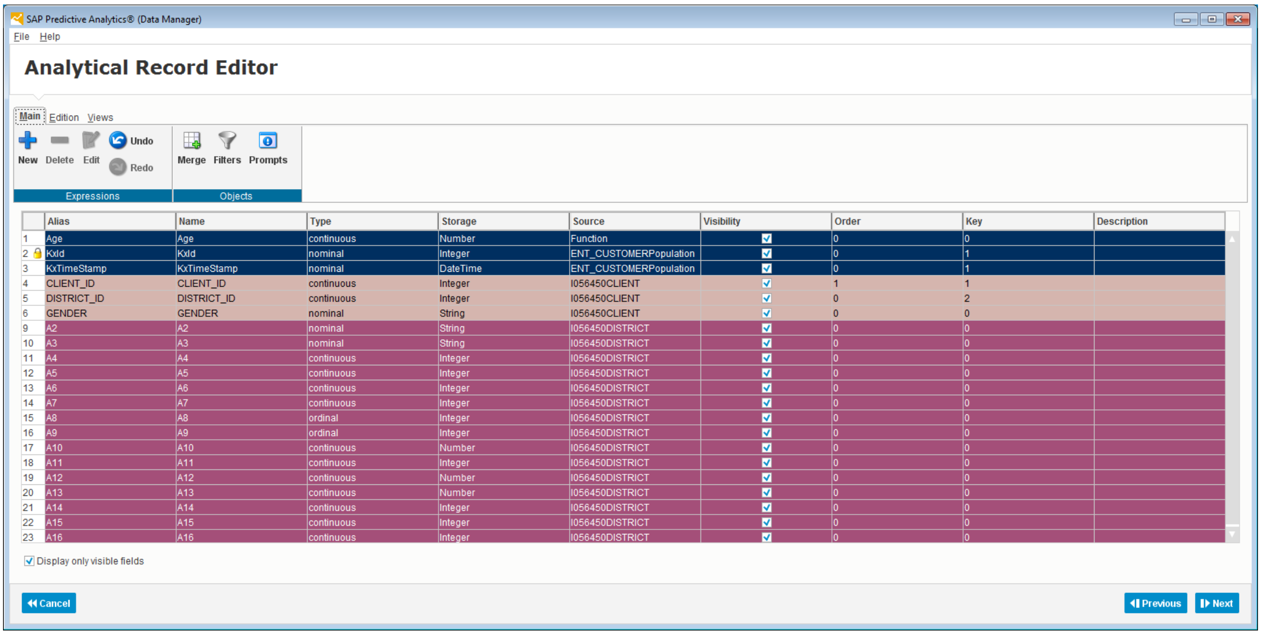
Click "OK" and name the column "Age". Hit "OK".



Name: Age

Press "Ok" to Validate

Our view of the customer is now more detailed.

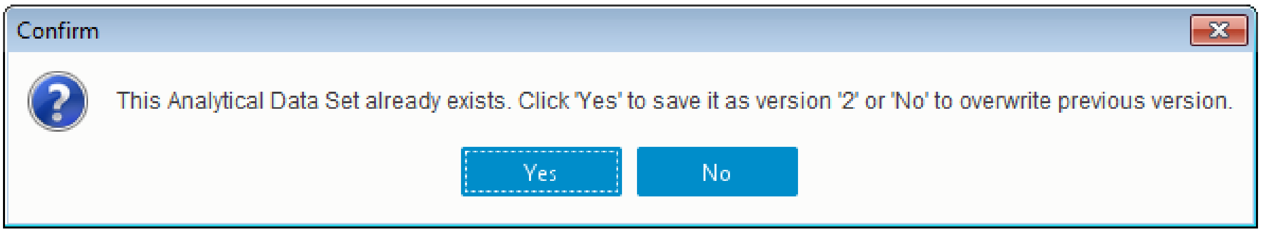


Analytical Record Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	Age	Age	continuous	Number	Function	☑	0	0	
2 🔒	KxId	KxId	nominal	Integer	ENT_CUSTOMERPopulation	☑	0	1	
3	KxTimeStamp	KxTimeStamp	nominal	DateTime	ENT_CUSTOMERPopulation	☑	0	1	
4	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450CLIENT	☑	1	1	
5	DISTRICT_ID	DISTRICT_ID	continuous	Integer	I056450CLIENT	☑	0	2	
6	GENDER	GENDER	nominal	String	I056450CLIENT	☑	0	0	
9	A2	A2	nominal	String	I056450DISTRICT	☑	0	0	
10	A3	A3	nominal	String	I056450DISTRICT	☑	0	0	
11	A4	A4	continuous	Integer	I056450DISTRICT	☑	0	0	
12	A5	A5	continuous	Integer	I056450DISTRICT	☑	0	0	
13	A6	A6	continuous	Integer	I056450DISTRICT	☑	0	0	
14	A7	A7	continuous	Integer	I056450DISTRICT	☑	0	0	
15	A8	A8	ordinal	Integer	I056450DISTRICT	☑	0	0	
16	A9	A9	ordinal	Integer	I056450DISTRICT	☑	0	0	
17	A10	A10	continuous	Number	I056450DISTRICT	☑	0	0	
18	A11	A11	continuous	Integer	I056450DISTRICT	☑	0	0	
19	A12	A12	continuous	Number	I056450DISTRICT	☑	0	0	
20	A13	A13	continuous	Number	I056450DISTRICT	☑	0	0	
21	A14	A14	continuous	Integer	I056450DISTRICT	☑	0	0	
22	A15	A15	continuous	Integer	I056450DISTRICT	☑	0	0	
23	A16	A16	continuous	Integer	I056450DISTRICT	☑	0	0	

☑ Display only visible fields

Click “Next” twice to end editing the analytical record. When asked whether to save the new logic as a new version, you could select “Yes” of course. For simplicity, in this document we overwrite the existing version. So select “No”.

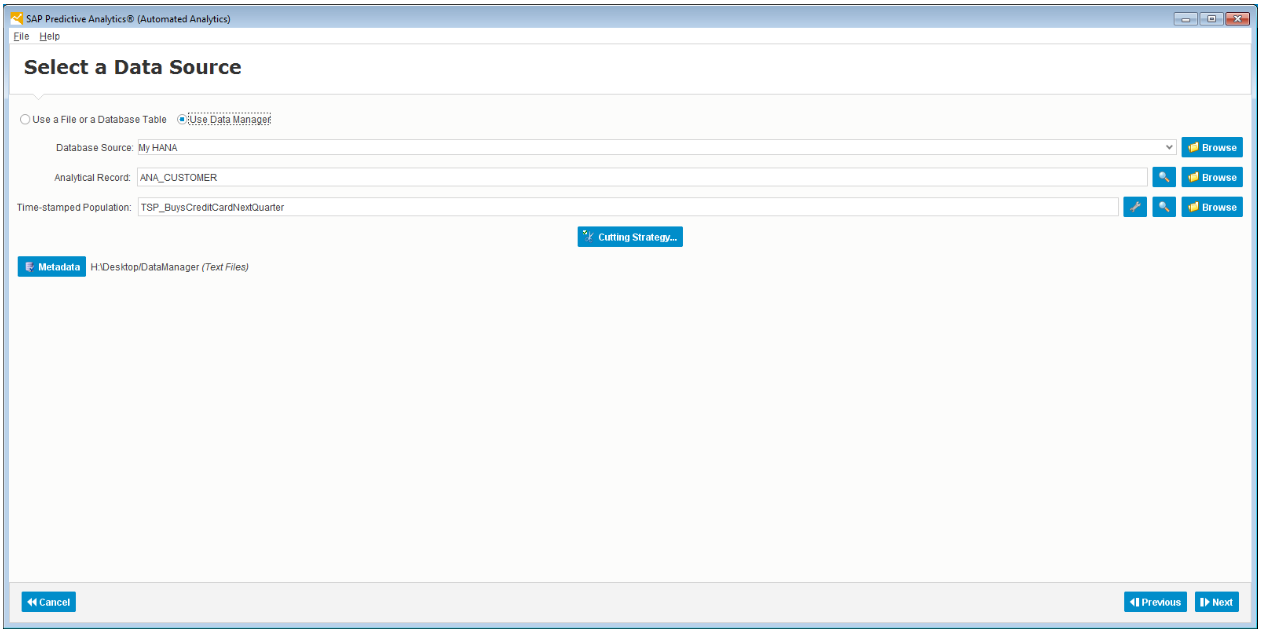


This Analytical Data Set already exists. Click 'Yes' to save it as version '2' or 'No' to overwrite previous version.

You are now back in the “Data Manager”.

Let’s create a new predictive model and see if it has improved. Select “Cancel” and create a classification as before.

The source:



Select a Data Source

○ Use a File or a Database Table ● Use Data Manager

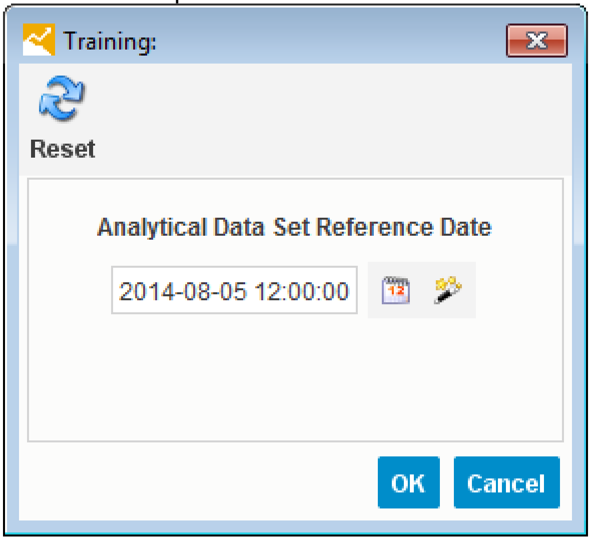
Database Source: My HANA

Analytical Record: ANA_CUSTOMER

Time-stamped Population: TSP_BuysCreditCardNextQuarter

Metadata H:\Desktop\DataManager *(Text Files)*

The time-stamp:

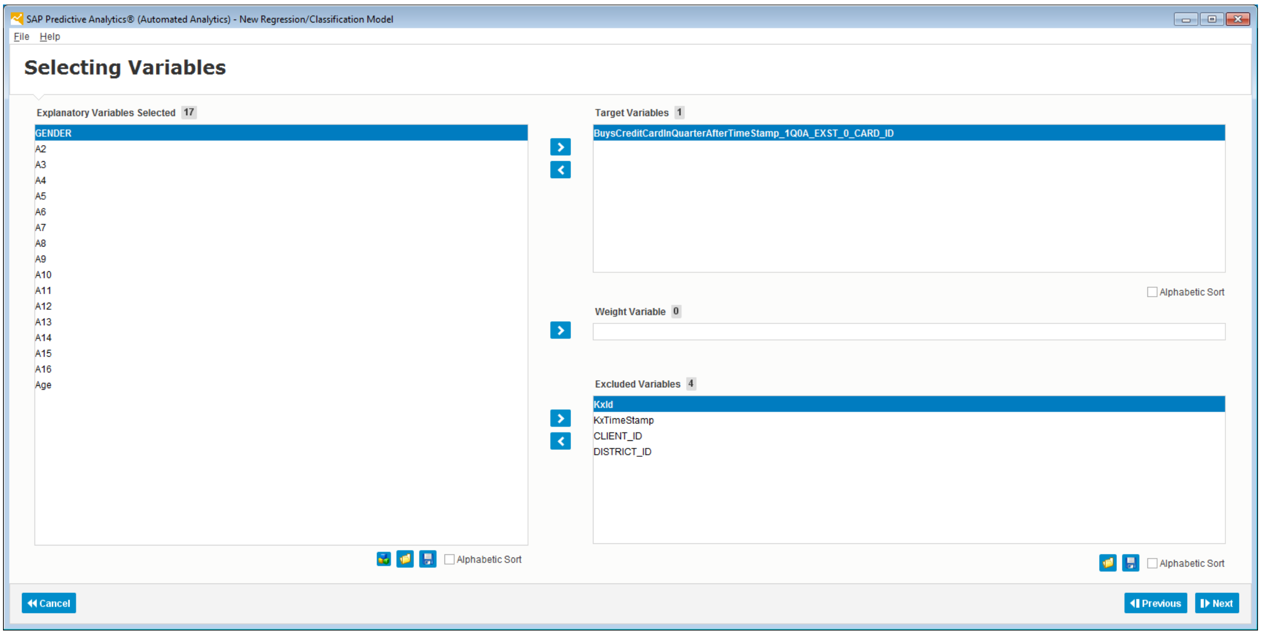


Training:

Analytical Data Set Reference Date

2014-08-05 12:00:00

The selected variables:



SAP Predictive Analytics® (Automated Analytics) - New Regression/Classification Model

File Help

Selecting Variables

Explanatory Variables Selected 17

GENDER
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
Age

Target Variables 1

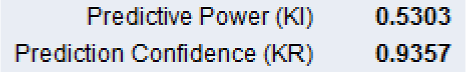
BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID

Weight Variable 0

Excluded Variables 4

KxId
KxTimeStamp
CLIENT_ID
DISTRICT_ID

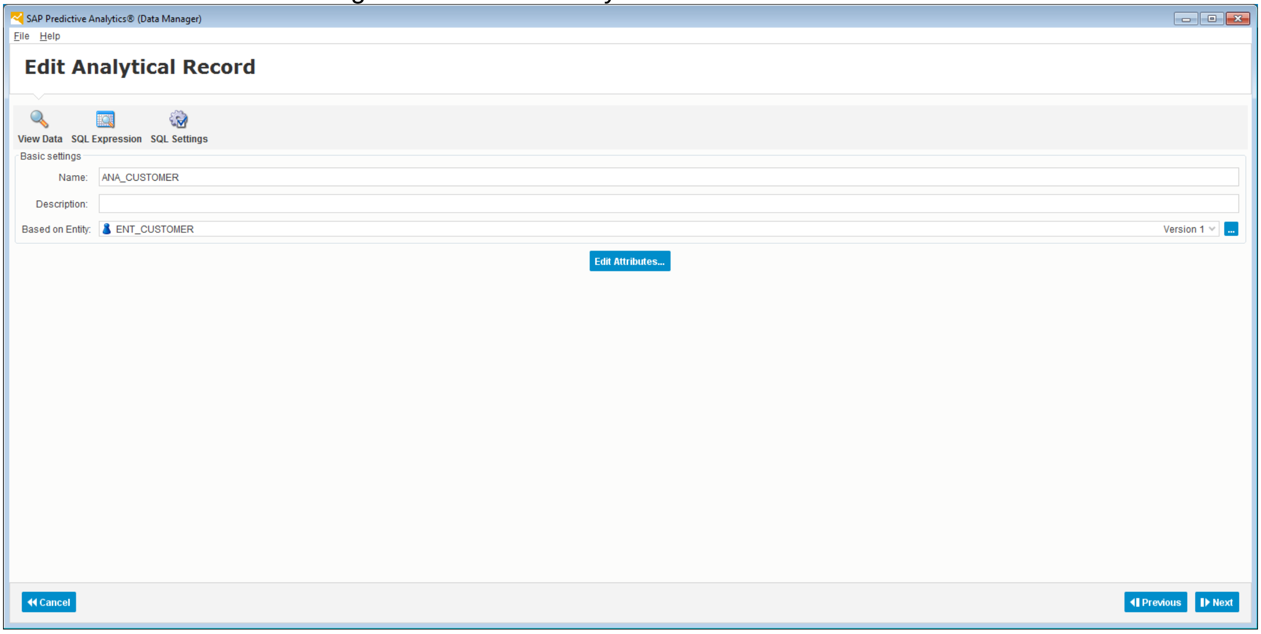
Create the model and we can see that it has already improved by a big margin.



Predictive Power (KI)	0.5303
Prediction Confidence (KR)	0.9357

Step 7 – Add time-related behavioural data
So far we have described the customer itself, ie the gender, age or where the person lives. Now we focus on the person’s behavior. We will look at the sum of the transaction amounts on the account, broken down by the four quarters before the time-stamp.

Get back into the “Data Manager” and edit the analytical record.



Edit Analytical Record

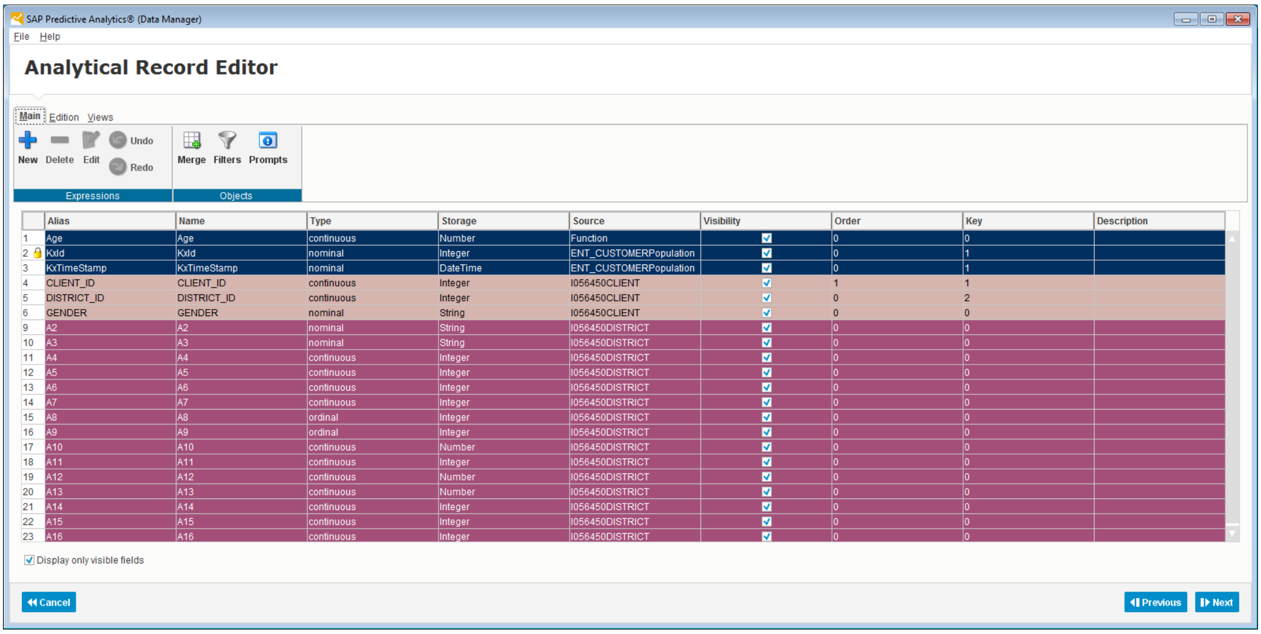
Basic settings

Name: ANA_CUSTOMER

Description:

Based on Entity: ENT_CUSTOMER Version 1

Click “Edit Attributes”.



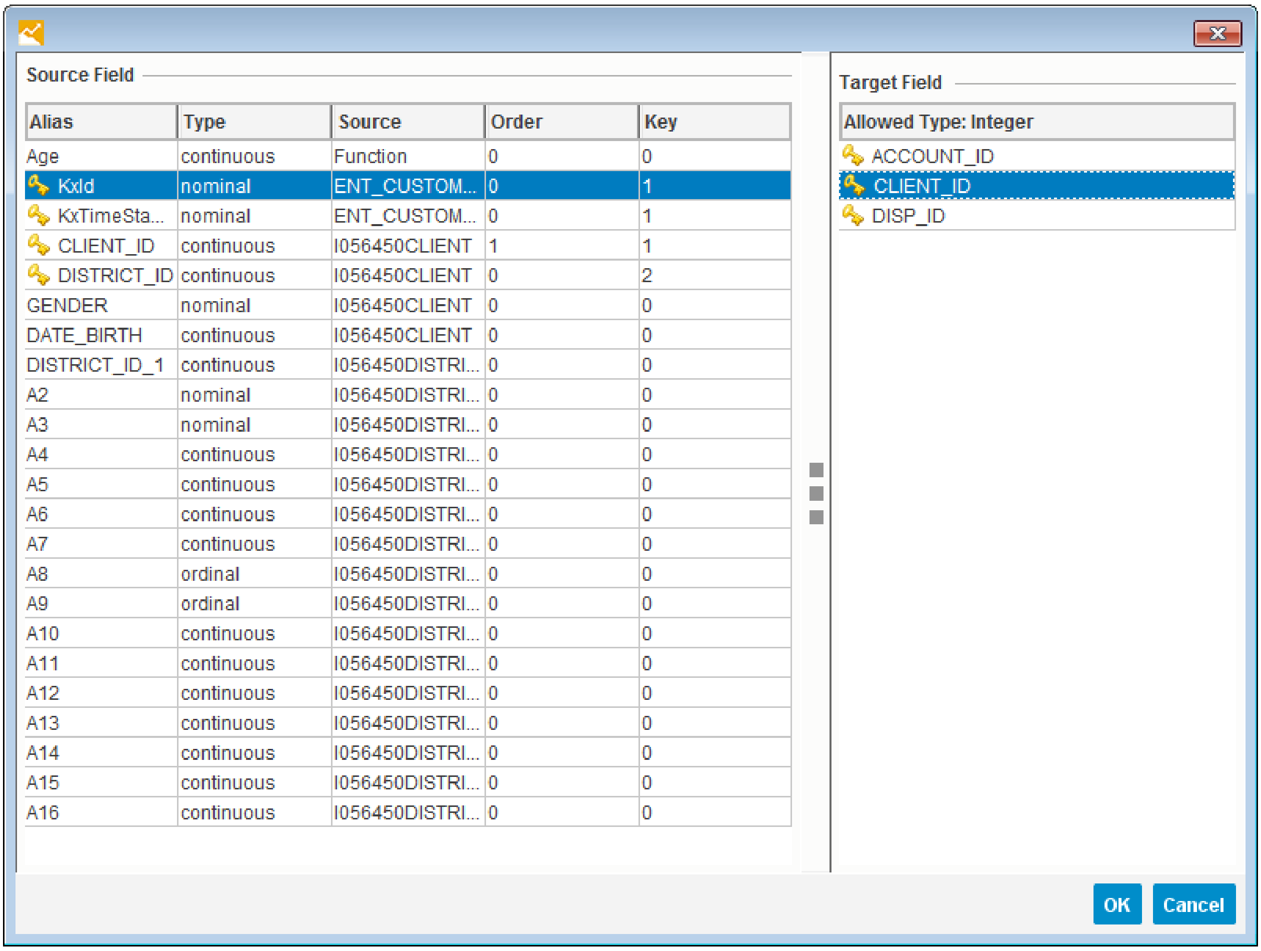
Analytical Record Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
1	Age	Age	continuous	Number	Function	☑	0	0	
2	KxId	KxId	nominal	Integer	ENT_CUSTOMERPopulation	☑	0	1	
3	KxTimeStamp	KxTimeStamp	nominal	DateTime	ENT_CUSTOMERPopulation	☑	0	1	
4	CLIENT_ID	CLIENT_ID	continuous	Integer	I056450CLIENT	☑	1	1	
5	DISTRICT_ID	DISTRICT_ID	continuous	Integer	I056450CLIENT	☑	0	2	
6	GENDER	GENDER	nominal	String	I056450CLIENT	☑	0	0	
9	A2	A2	nominal	String	I056450DISTRICT	☑	0	0	
10	A3	A3	nominal	String	I056450DISTRICT	☑	0	0	
11	A4	A4	continuous	Integer	I056450DISTRICT	☑	0	0	
12	A5	A5	continuous	Integer	I056450DISTRICT	☑	0	0	
13	A6	A6	continuous	Integer	I056450DISTRICT	☑	0	0	
14	A7	A7	continuous	Integer	I056450DISTRICT	☑	0	0	
15	A8	A8	ordinal	Integer	I056450DISTRICT	☑	0	0	
16	A9	A9	ordinal	Integer	I056450DISTRICT	☑	0	0	
17	A10	A10	continuous	Number	I056450DISTRICT	☑	0	0	
18	A11	A11	continuous	Integer	I056450DISTRICT	☑	0	0	
19	A12	A12	continuous	Number	I056450DISTRICT	☑	0	0	
20	A13	A13	continuous	Number	I056450DISTRICT	☑	0	0	
21	A14	A14	continuous	Integer	I056450DISTRICT	☑	0	0	
22	A15	A15	continuous	Integer	I056450DISTRICT	☑	0	0	
23	A16	A16	continuous	Integer	I056450DISTRICT	☑	0	0	

☑ Display only visible fields

To include the behavior in the model we have to go through the “DISPOSITON” and “ACCOUNT” table to get to the “TRANSACTIONS” table. Therefore merge in first the “DISPOSITION” table followed by the “ACCOUNT” table.

The “DISPOSITION” table:



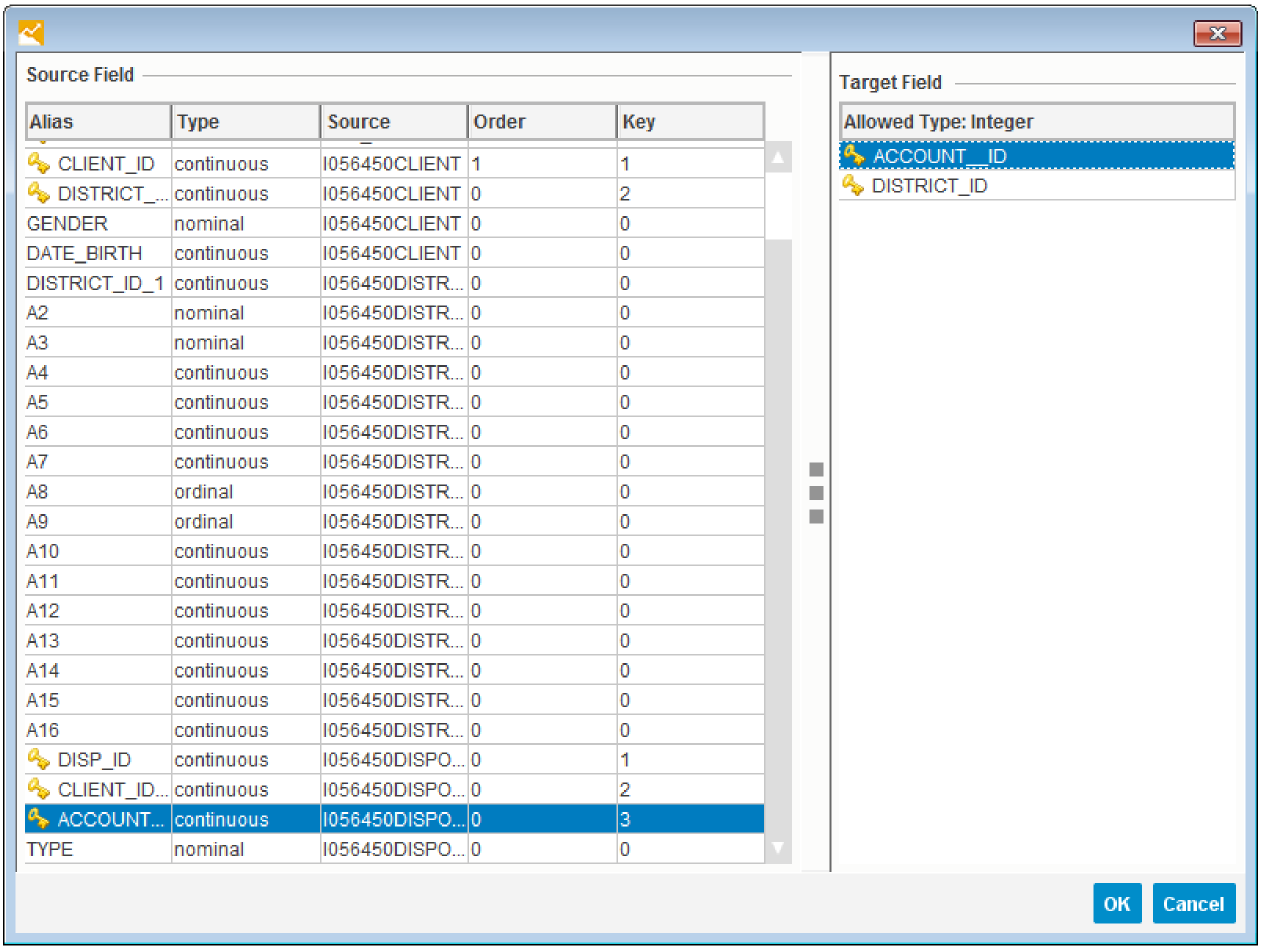
Source Field

Alias	Type	Source	Order	Key
Age	continuous	Function	0	0
KxId	nominal	ENT_CUSTOM...	0	1
KxTimeSta...	nominal	ENT_CUSTOM...	0	1
CLIENT_ID	continuous	I056450CLIENT	1	1
DISTRICT_ID	continuous	I056450CLIENT	0	2
GENDER	nominal	I056450CLIENT	0	0
DATE_BIRTH	continuous	I056450CLIENT	0	0
DISTRICT_ID_1	continuous	I056450DISTRI...	0	0
A2	nominal	I056450DISTRI...	0	0
A3	nominal	I056450DISTRI...	0	0
A4	continuous	I056450DISTRI...	0	0
A5	continuous	I056450DISTRI...	0	0
A6	continuous	I056450DISTRI...	0	0
A7	continuous	I056450DISTRI...	0	0
A8	ordinal	I056450DISTRI...	0	0
A9	ordinal	I056450DISTRI...	0	0
A10	continuous	I056450DISTRI...	0	0
A11	continuous	I056450DISTRI...	0	0
A12	continuous	I056450DISTRI...	0	0
A13	continuous	I056450DISTRI...	0	0
A14	continuous	I056450DISTRI...	0	0
A15	continuous	I056450DISTRI...	0	0
A16	continuous	I056450DISTRI...	0	0

Target Field

Allowed Type: Integer
ACCOUNT_ID
CLIENT_ID
DISP_ID

The “ACCOUNT” table:



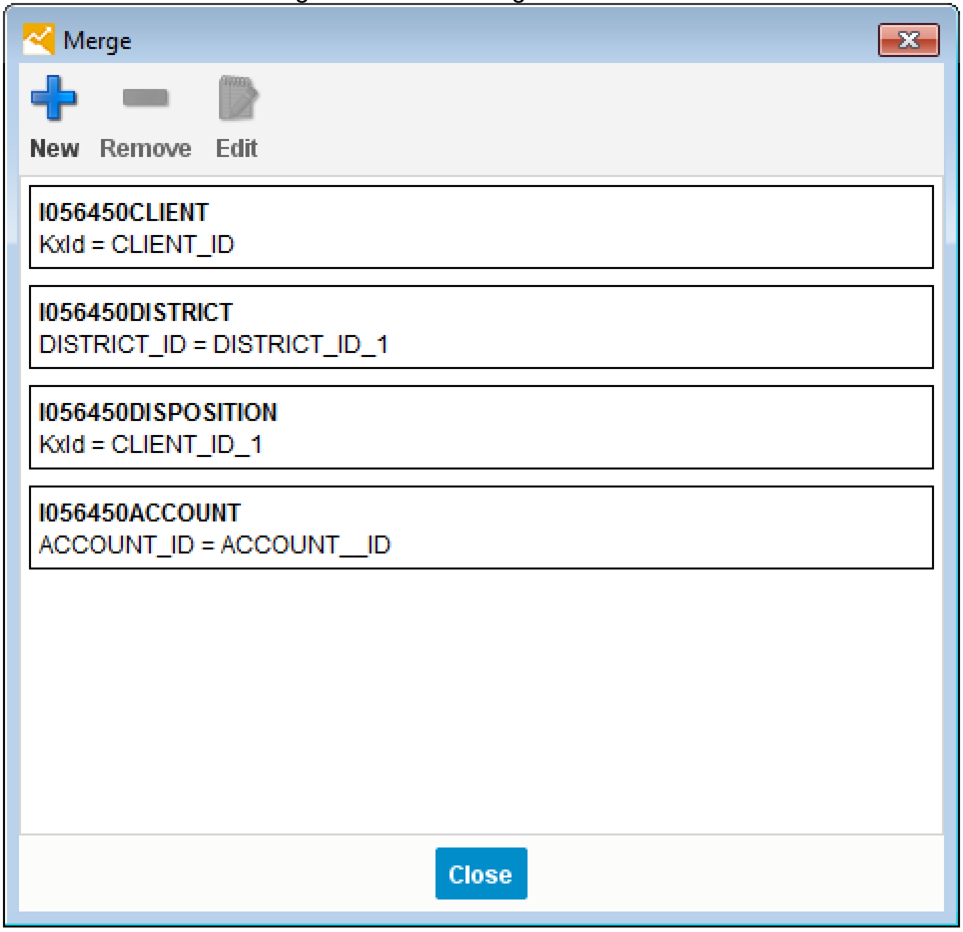
Source Field

Alias	Type	Source	Order	Key
CLIENT_ID	continuous	I056450CLIENT	1	1
DISTRICT_...	continuous	I056450CLIENT	0	2
GENDER	nominal	I056450CLIENT	0	0
DATE_BIRTH	continuous	I056450CLIENT	0	0
DISTRICT_ID_1	continuous	I056450DISTR...	0	0
A2	nominal	I056450DISTR...	0	0
A3	nominal	I056450DISTR...	0	0
A4	continuous	I056450DISTR...	0	0
A5	continuous	I056450DISTR...	0	0
A6	continuous	I056450DISTR...	0	0
A7	continuous	I056450DISTR...	0	0
A8	ordinal	I056450DISTR...	0	0
A9	ordinal	I056450DISTR...	0	0
A10	continuous	I056450DISTR...	0	0
A11	continuous	I056450DISTR...	0	0
A12	continuous	I056450DISTR...	0	0
A13	continuous	I056450DISTR...	0	0
A14	continuous	I056450DISTR...	0	0
A15	continuous	I056450DISTR...	0	0
A16	continuous	I056450DISTR...	0	0
DISP_ID	continuous	I056450DISPO...	0	1
CLIENT_ID...	continuous	I056450DISPO...	0	2
ACCOUNT...	continuous	I056450DISPO...	0	3
TYPE	nominal	I056450DISPO...	0	0

Target Field

Allowed Type: Integer
ACCOUNT__ID
DISTRICT_ID

This leads to the following collection of merges:



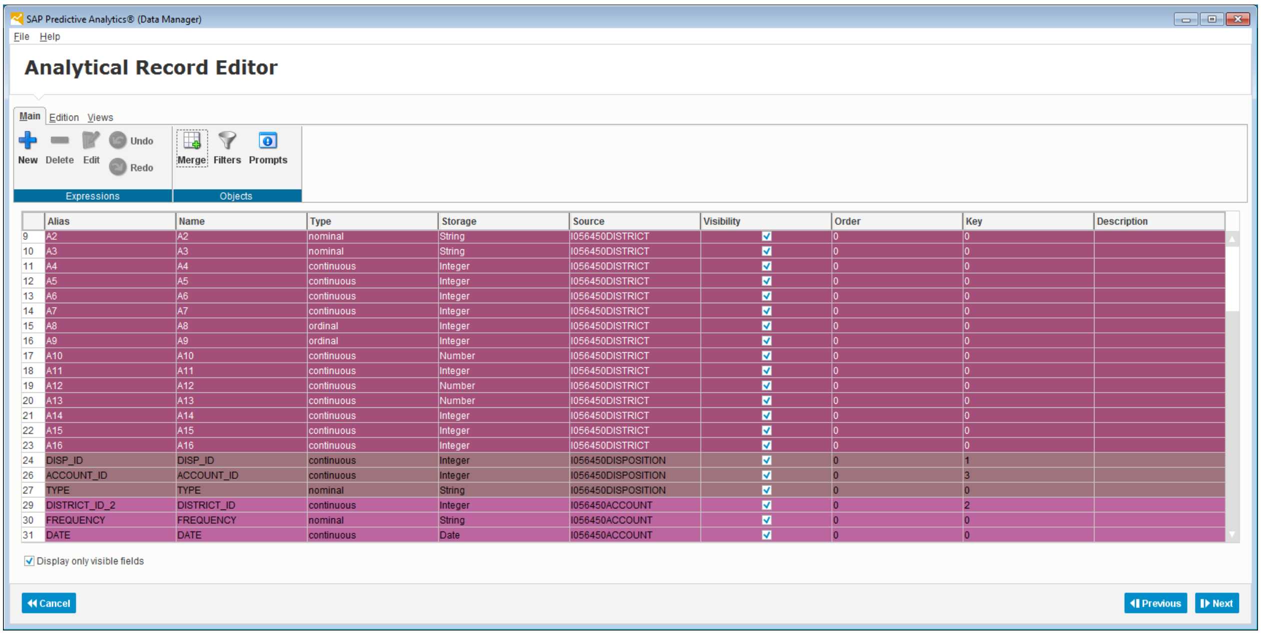
I056450CLIENT
KxId = CLIENT_ID

I056450DISTRICT
DISTRICT_ID = DISTRICT_ID_1

I056450DISPOSITION
KxId = CLIENT_ID_1

I056450ACCOUNT
ACCOUNT_ID = ACCOUNT__ID

Close this window and the analytical record contains the new columns. You may have to scroll down to see them.

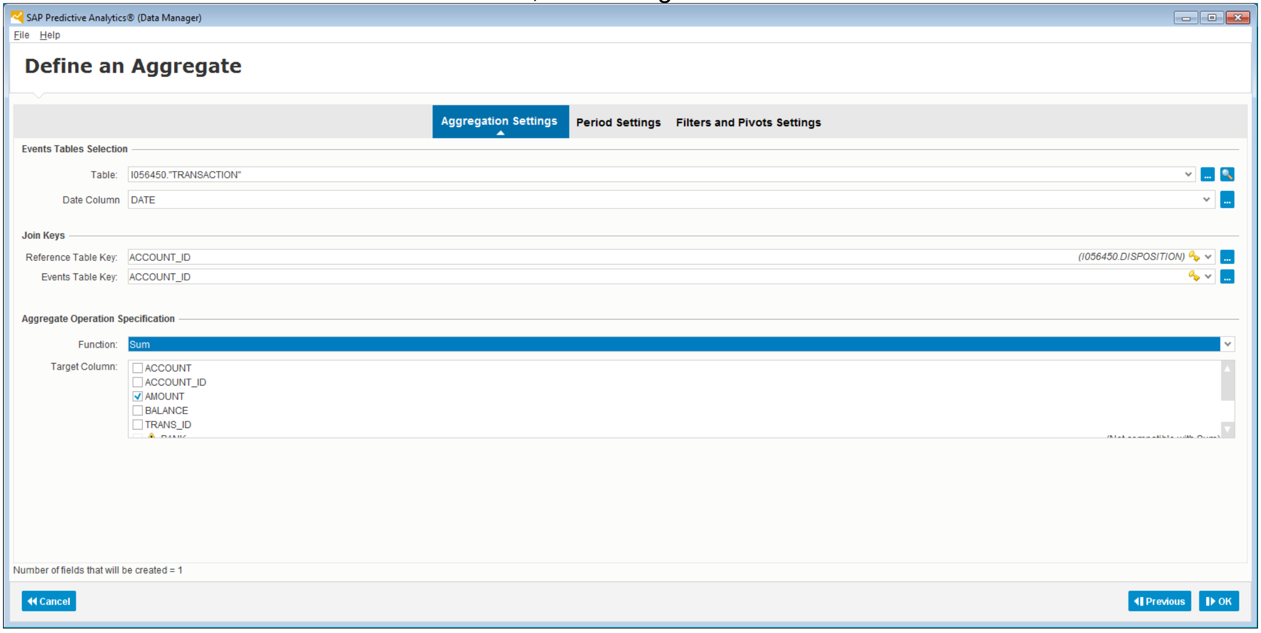


Analytical Record Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
9	A2	A2	nominal	String	I056450DISTRICT	☑	0	0	
10	A3	A3	nominal	String	I056450DISTRICT	☑	0	0	
11	A4	A4	continuous	Integer	I056450DISTRICT	☑	0	0	
12	A5	A5	continuous	Integer	I056450DISTRICT	☑	0	0	
13	A6	A6	continuous	Integer	I056450DISTRICT	☑	0	0	
14	A7	A7	continuous	Integer	I056450DISTRICT	☑	0	0	
15	A8	A8	ordinal	Integer	I056450DISTRICT	☑	0	0	
16	A9	A9	ordinal	Integer	I056450DISTRICT	☑	0	0	
17	A10	A10	continuous	Number	I056450DISTRICT	☑	0	0	
18	A11	A11	continuous	Integer	I056450DISTRICT	☑	0	0	
19	A12	A12	continuous	Number	I056450DISTRICT	☑	0	0	
20	A13	A13	continuous	Number	I056450DISTRICT	☑	0	0	
21	A14	A14	continuous	Integer	I056450DISTRICT	☑	0	0	
22	A15	A15	continuous	Integer	I056450DISTRICT	☑	0	0	
23	A16	A16	continuous	Integer	I056450DISTRICT	☑	0	0	
24	DISP_ID	DISP_ID	continuous	Integer	I056450DISPOSITION	☑	0	1	
26	ACCOUNT_ID	ACCOUNT_ID	continuous	Integer	I056450DISPOSITION	☑	0	3	
27	TYPE	TYPE	nominal	String	I056450DISPOSITION	☑	0	0	
29	DISTRICT_ID_2	DISTRICT_ID	continuous	Integer	I056450ACCOUNT	☑	0	2	
30	FREQUENCY	FREQUENCY	nominal	String	I056450ACCOUNT	☑	0	0	
31	DATE	DATE	continuous	Date	I056450ACCOUNT	☑	0	0	

☑ Display only visible fields

With these tables available, we can now use the “TRANSACTION” table. Select “New” → “New Aggregate”. Link to the “TRANSACTION” table as follows, calculating the sum of the transaction amounts:


SAP Predictive Analytics® (Data Manager)
File Help
Aggregation Settings
Period Settings
Filters and Pivots Settings
Cancel
Previous
OK

Define an Aggregate

Events Tables Selection

Table: I056450."TRANSACTION"

Date Column DATE

Join Keys

Reference Table Key: ACCOUNT_ID (I056450.DISPOSITION)

Events Table Key: ACCOUNT_ID

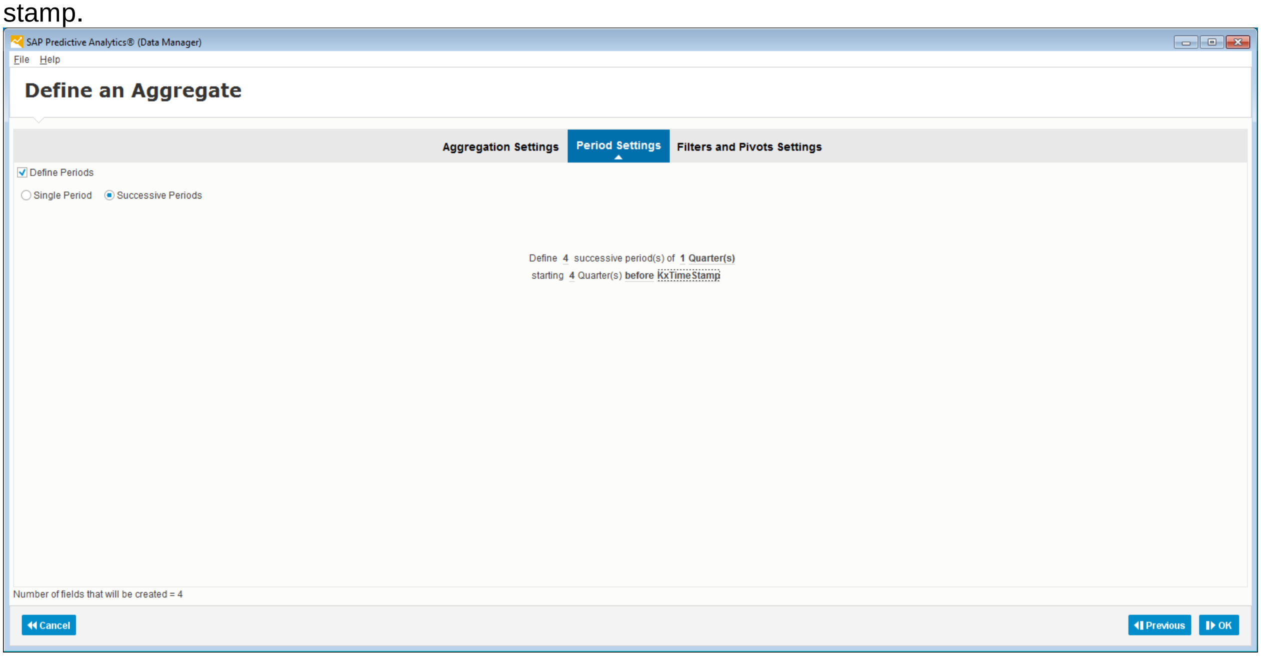
Aggregate Operation Specification

Function: Sum

Target Column:
- □ ACCOUNT
- □ ACCOUNT_ID
- ☑ AMOUNT
- □ BALANCE
- □ TRANS_ID

Number of fields that will be created = 1

Continue in the “Period Settings”. Have the sum calculated individually for the 4 quarters before the time-stamp.


SAP Predictive Analytics® (Data Manager)
File Help
Aggregation Settings
Period Settings
Filters and Pivots Settings
Cancel
Previous
OK

Define an Aggregate

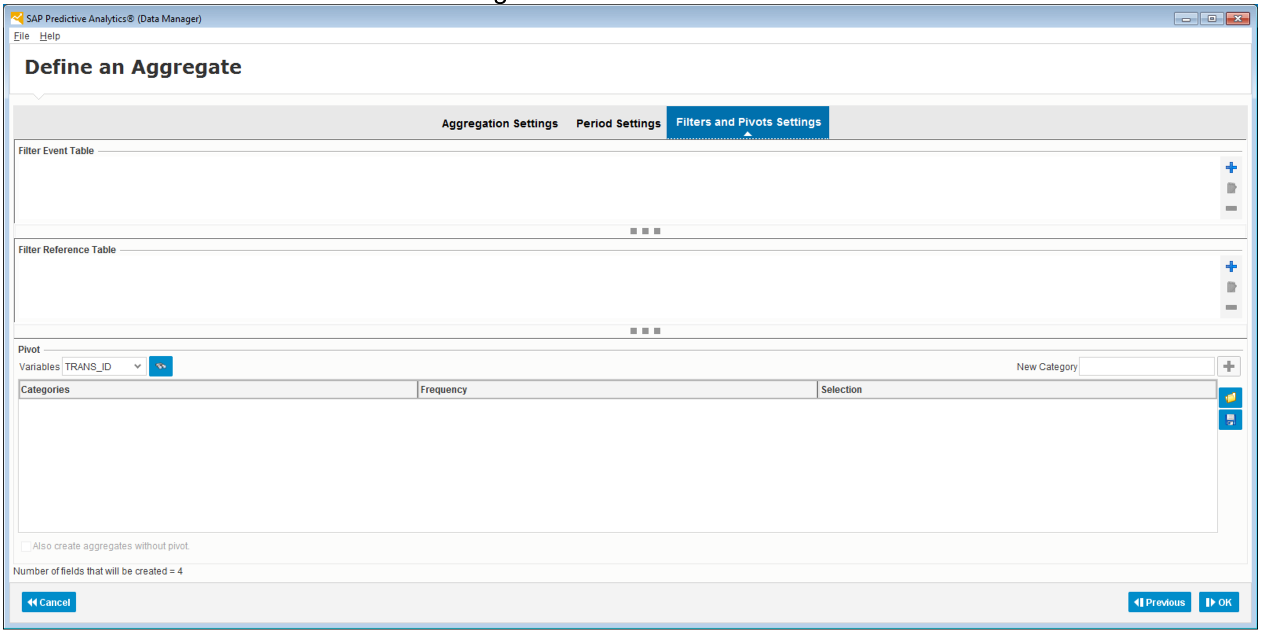
☑ Define Periods

○ Single Period ● Successive Periods

Define 4 successive period(s) of 1 Quarter(s)
starting 4 Quarter(s) before KxTimeStamp

Number of fields that will be created = 4

These quarterly aggregates could already help produce a better model. However, let’s make it even more detailed. Go into “Filters and Pivots Settings”.



SAP Predictive Analytics® (Data Manager)

Define an Aggregate

Filter Event Table

Filter Reference Table

Pivot

Variables TRANS_ID

Categories	Frequency	Selection

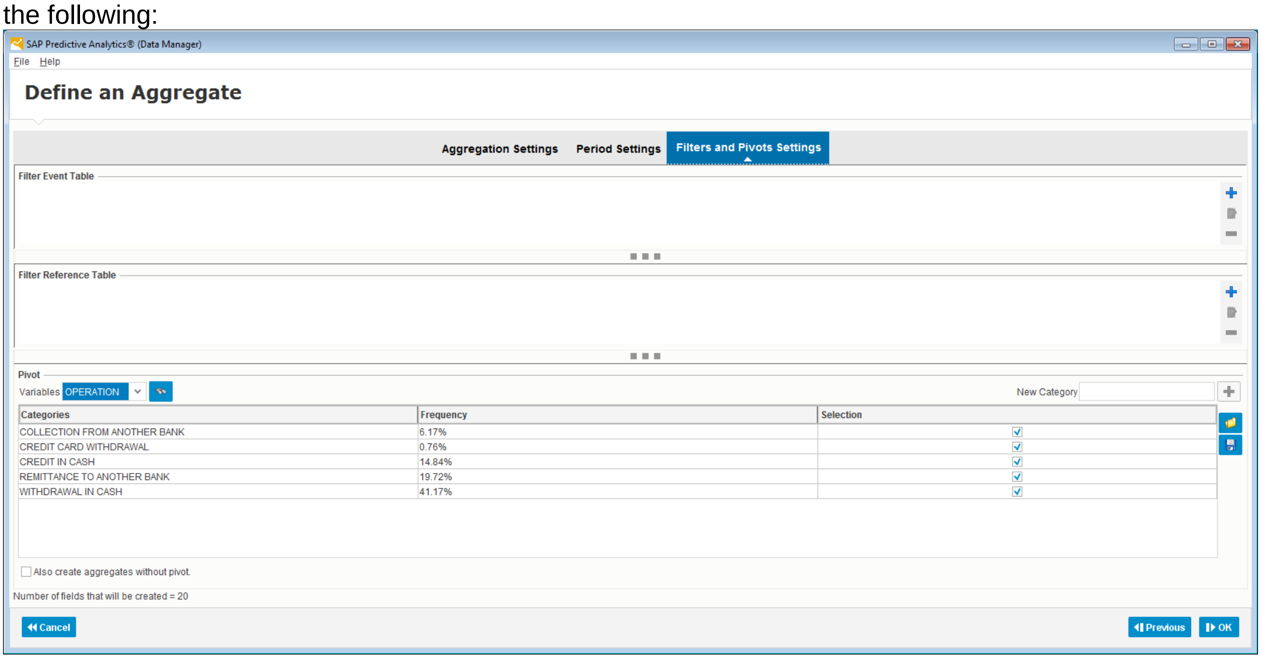
□ Also create aggregates without pivot.

Number of fields that will be created = 4

Here we break down the quarterly value in more detailed aggregates. The “TRANSACTION” table specifies the transaction’s operation, ie a cash withdrawal or credit card withdrawal. We will create individual variables by quarter by operation for the last 4 quarters.

At the bottom left in the “Pivot” section, set “Variable” to “OPERATION”. Click the binoculars next to it and chose “from the whole dataset”. This will load all possible values from the “OPERATION” column. Just be patient in case this takes a few seconds. With larger datasets you will want to pick one of the other options to prevent that all data gets downloaded on your client. Each of these values will become a quarterly aggregation variable, for each of the 4 specified quarters.

Once complete, the “Variable” drop down might change. Set it back to “OPERATION” and you should see the following:



SAP Predictive Analytics® (Data Manager)

Define an Aggregate

Filter Event Table

Filter Reference Table

Pivot

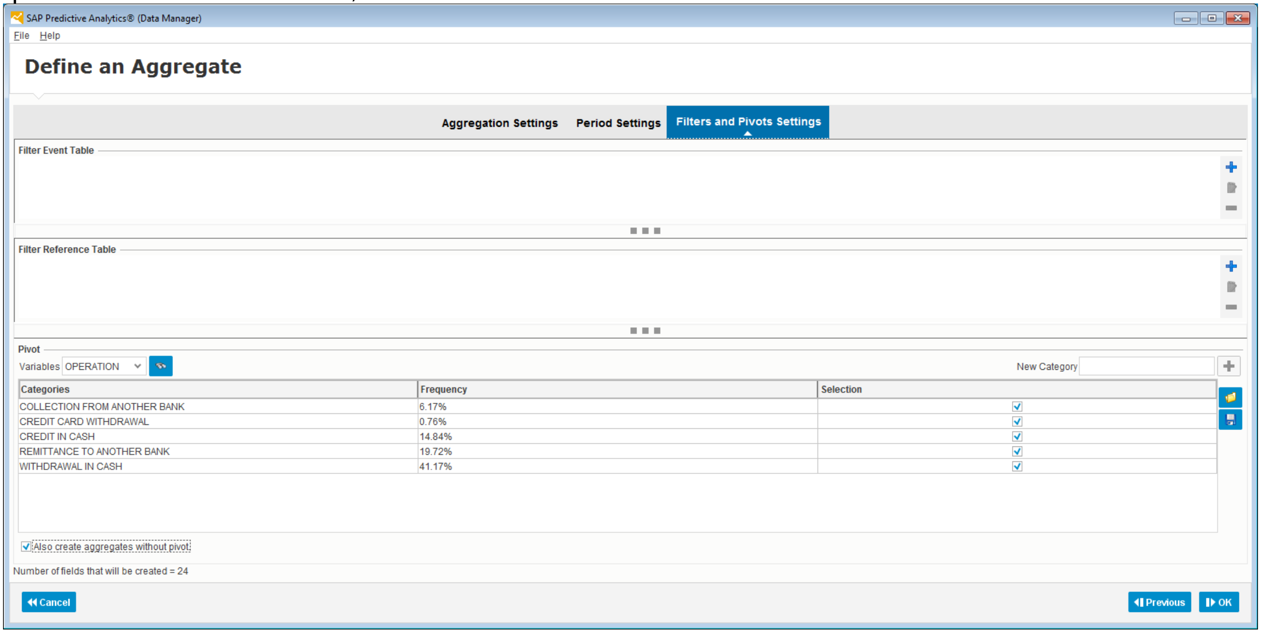
Variables OPERATION

Categories	Frequency	Selection
COLLECTION FROM ANOTHER BANK	6.17%	☑
CREDIT CARD WITHDRAWAL	0.76%	☑
CREDIT IN CASH	14.84%	☑
REMITTANCE TO ANOTHER BANK	19.72%	☑
WITHDRAWAL IN CASH	41.17%	☑

□ Also create aggregates without pivot.

Number of fields that will be created = 20

Select “Also create aggregates without pivot”. This will create another quarterly variable with the total sum. So in addition to the 5 values in “OPERATION” a 6th column with the total value is calculated. 6 columns by 4 quarters makes 24 new fields, which is also indicated at the bottom left of the screen.


SAP Predictive Analytics® (Data Manager)
File Help

Define an Aggregate

Aggregation Settings
Period Settings
Filters and Pivots Settings

Filter Event Table

Filter Reference Table

Pivot

Variables OPERATION

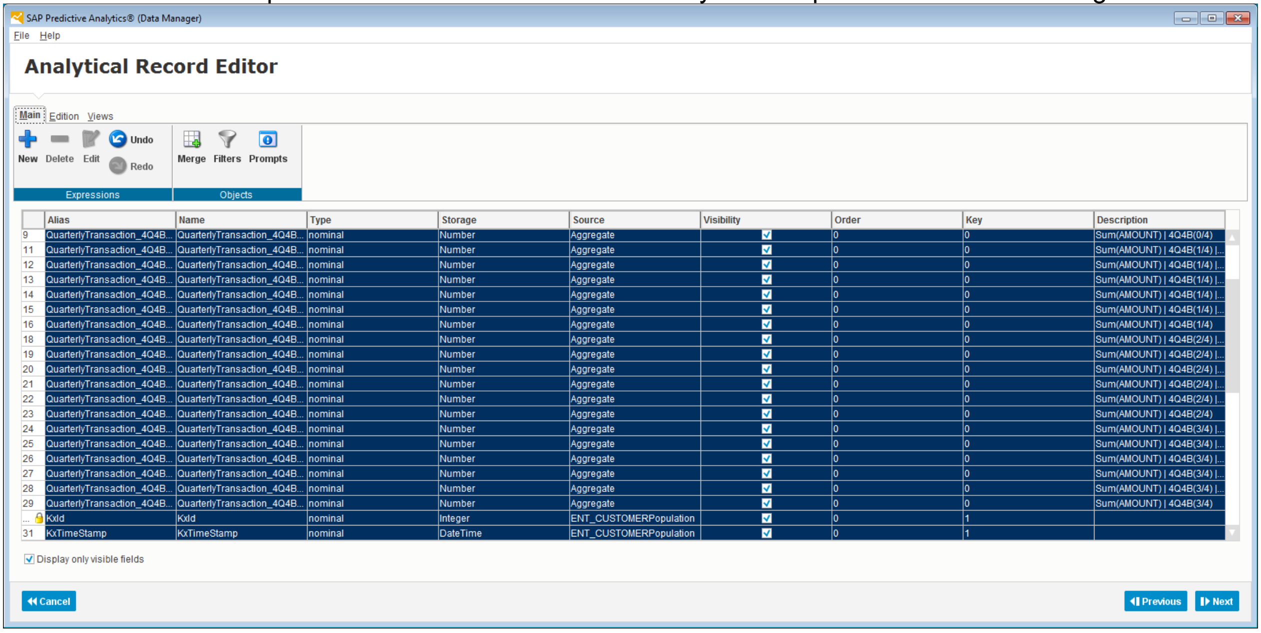
New Category

Categories	Frequency	Selection
COLLECTION FROM ANOTHER BANK	6.17%	☑
CREDIT CARD WITHDRAWAL	0.76%	☑
CREDIT IN CASH	14.84%	☑
REMITTANCE TO ANOTHER BANK	19.72%	☑
WITHDRAWAL IN CASH	41.17%	☑

☑ Also create aggregates without pivot

Number of fields that will be created = 24

Click “OK” and name the variable “call it QuarterlyTransaction”. Hit “OK” and you will see the new variables. Each variable has the prefix that we had chosen followed by a description of the variable’s logic.

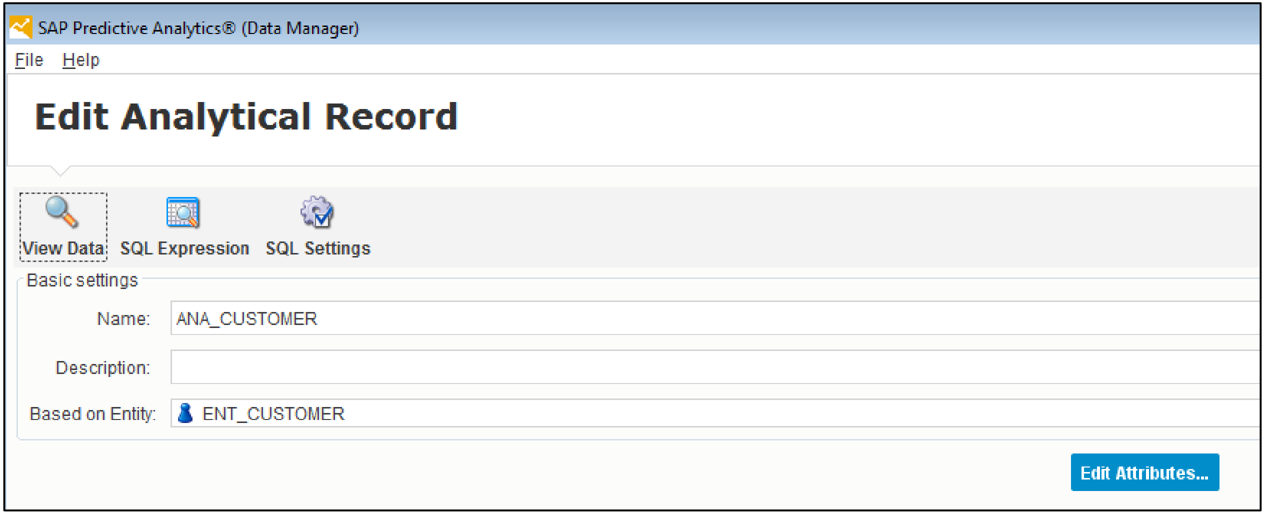


Analytical Record Editor

	Alias	Name	Type	Storage	Source	Visibility	Order	Key	Description
9	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(0/4)
11	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(1/4) |...
12	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(1/4) |...
13	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(1/4) |...
14	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(1/4) |...
15	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(1/4) |...
16	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(1/4)
18	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(2/4) |...
19	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(2/4) |...
20	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(2/4) |...
21	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(2/4) |...
22	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(2/4) |...
23	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(2/4)
24	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(3/4) |...
25	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(3/4) |...
26	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(3/4) |...
27	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(3/4) |...
28	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(3/4) |...
29	QuarterlyTransaction_4Q4B...	QuarterlyTransaction_4Q4B...	nominal	Number	Aggregate	☑	0	0	Sum(AMOUNT) | 4Q4B(3/4)
...	KxId	KxId	nominal	Integer	ENT_CUSTOMERPopulation	☑	0	1	
31	KxTimeStamp	KxTimeStamp	nominal	DateTime	ENT_CUSTOMERPopulation	☑	0	1	

☑ Display only visible fields

Click “Next” once. If you are interested to have a look at the new colums, click the “View Data” icon.



Edit Analytical Record

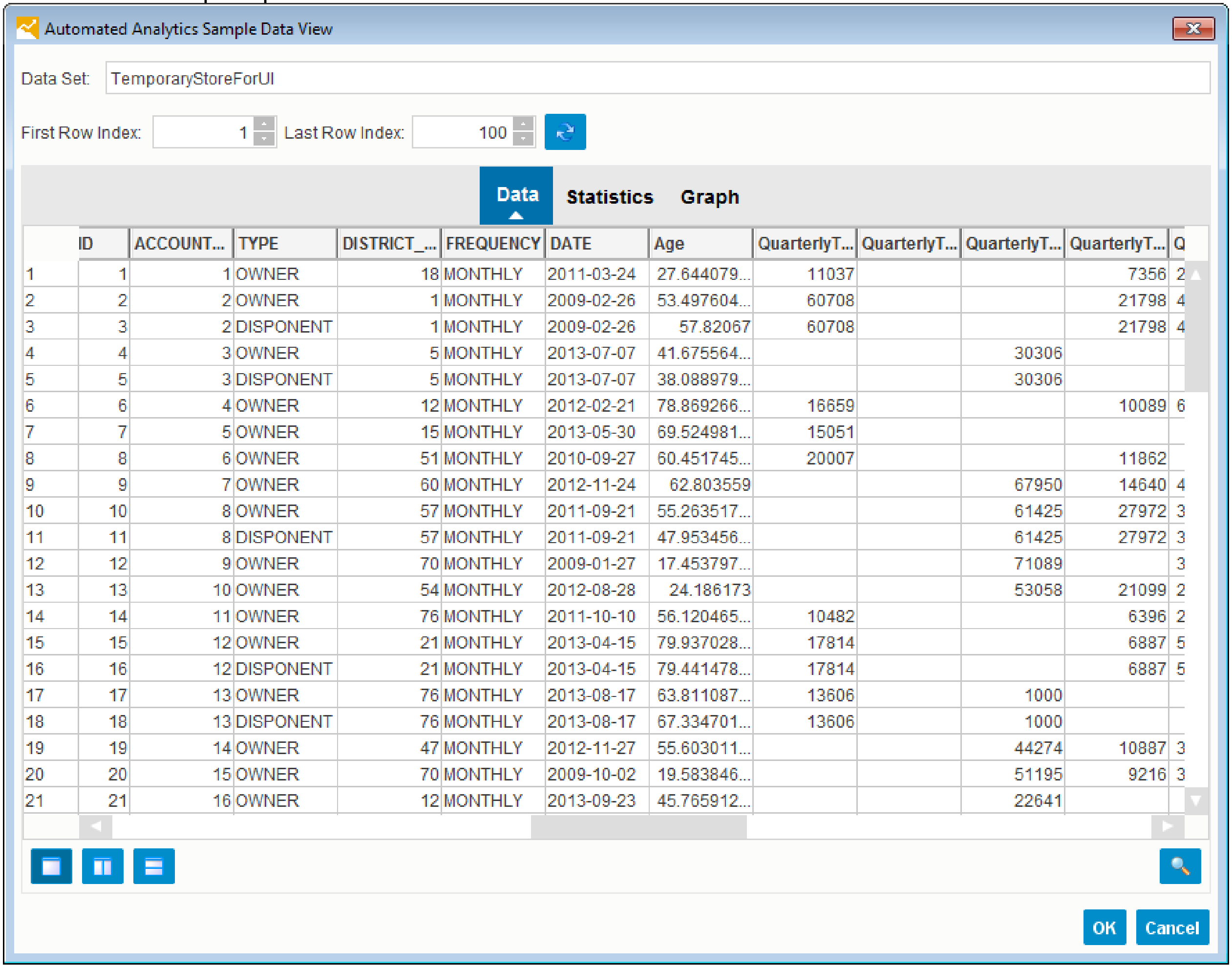
Basic settings

Name: ANA_CUSTOMER

Description:

Based on Entity: ENT_CUSTOMER

Confirm the date prompt and see the first 100 rows.



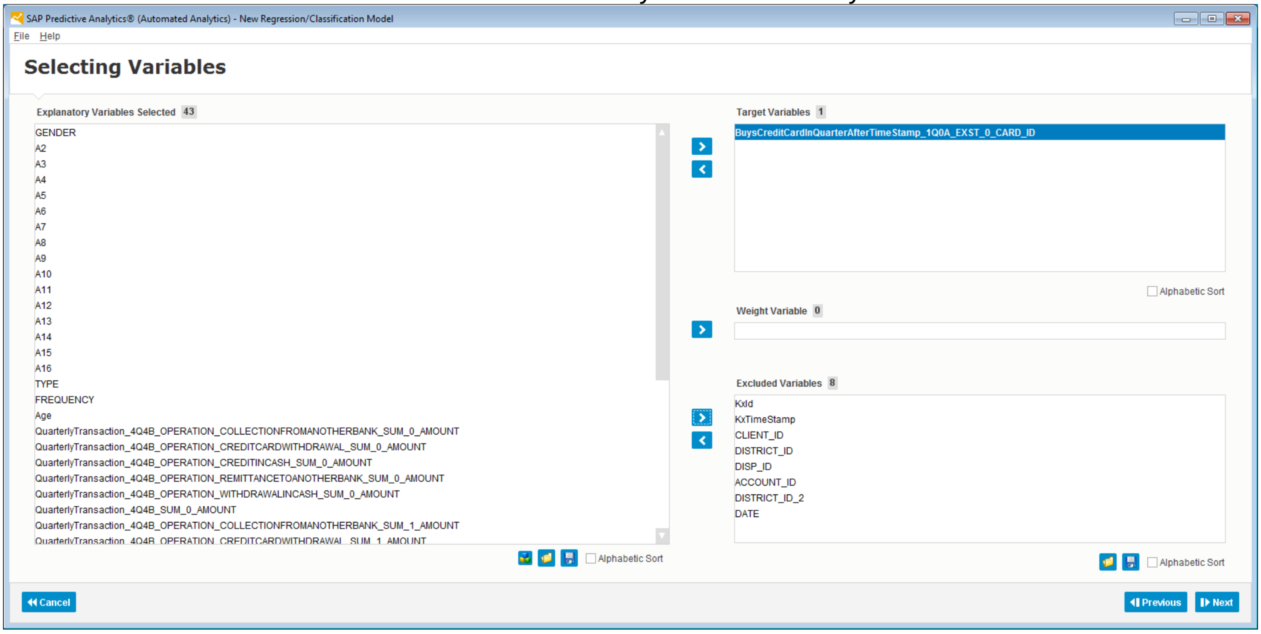
Data Set: TemporaryStoreForUI

First Row Index: 1 Last Row Index: 100

	ID	ACCOUNT...	TYPE	DISTRICT_...	FREQUENCY	DATE	Age	QuarterlyT...	QuarterlyT...	QuarterlyT...	QuarterlyT...	Q
1	1	1	OWNER	18	MONTHLY	2011-03-24	27.644079...	11037			7356	2
2	2	2	OWNER	1	MONTHLY	2009-02-26	53.497604...	60708			21798	4
3	3	2	DISPONENT	1	MONTHLY	2009-02-26	57.82067	60708			21798	4
4	4	3	OWNER	5	MONTHLY	2013-07-07	41.675564...			30306		
5	5	3	DISPONENT	5	MONTHLY	2013-07-07	38.088979...			30306		
6	6	4	OWNER	12	MONTHLY	2012-02-21	78.869266...	16659			10089	6
7	7	5	OWNER	15	MONTHLY	2013-05-30	69.524981...	15051				
8	8	6	OWNER	51	MONTHLY	2010-09-27	60.451745...	20007			11862	
9	9	7	OWNER	60	MONTHLY	2012-11-24	62.803559			67950	14640	4
10	10	8	OWNER	57	MONTHLY	2011-09-21	55.263517...			61425	27972	3
11	11	8	DISPONENT	57	MONTHLY	2011-09-21	47.953456...			61425	27972	3
12	12	9	OWNER	70	MONTHLY	2009-01-27	17.453797...			71089		3
13	13	10	OWNER	54	MONTHLY	2012-08-28	24.186173			53058	21099	2
14	14	11	OWNER	76	MONTHLY	2011-10-10	56.120465...	10482			6396	2
15	15	12	OWNER	21	MONTHLY	2013-04-15	79.937028...	17814			6887	5
16	16	12	DISPONENT	21	MONTHLY	2013-04-15	79.441478...	17814			6887	5
17	17	13	OWNER	76	MONTHLY	2013-08-17	63.811087...	13606		1000		
18	18	13	DISPONENT	76	MONTHLY	2013-08-17	67.334701...	13606		1000		
19	19	14	OWNER	47	MONTHLY	2012-11-27	55.603011...			44274	10887	3
20	20	15	OWNER	70	MONTHLY	2009-10-02	19.583846...			51195	9216	3
21	21	16	OWNER	12	MONTHLY	2013-09-23	45.765912...			22641		

Close this window with “OK”. Hit “Next” and save the model when prompted.

Now create a new classification model. You should only have to manually exclude the “DATE” column.


SAP Predictive Analytics® (Automated Analytics) - New Regression/Classification Model
File
Help
Alphabetic Sort
Alphabetic Sort
Alphabetic Sort
Cancel
Previous
Next

Selecting Variables

Explanatory Variables Selected 43

GENDER
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
TYPE
FREQUENCY
Age
QuarterlyTransaction_4Q4B_OPERATION_COLLECTIONFROMANOTHERBANK_SUM_0_AMOUNT
QuarterlyTransaction_4Q4B_OPERATION_CREDITCARDWITHDRAWAL_SUM_0_AMOUNT
QuarterlyTransaction_4Q4B_OPERATION_CREDITINCASH_SUM_0_AMOUNT
QuarterlyTransaction_4Q4B_OPERATION_REMITTANCETOANOTHERBANK_SUM_0_AMOUNT
QuarterlyTransaction_4Q4B_OPERATION_WITHDRAWALINCASH_SUM_0_AMOUNT
QuarterlyTransaction_4Q4B_SUM_0_AMOUNT
QuarterlyTransaction_4Q4B_OPERATION_COLLECTIONFROMANOTHERBANK_SUM_1_AMOUNT
QuarterlyTransaction_4Q4B_OPERATION_CREDITCARDWITHDRAWAL_SUM_1_AMOUNT

Target Variables 1

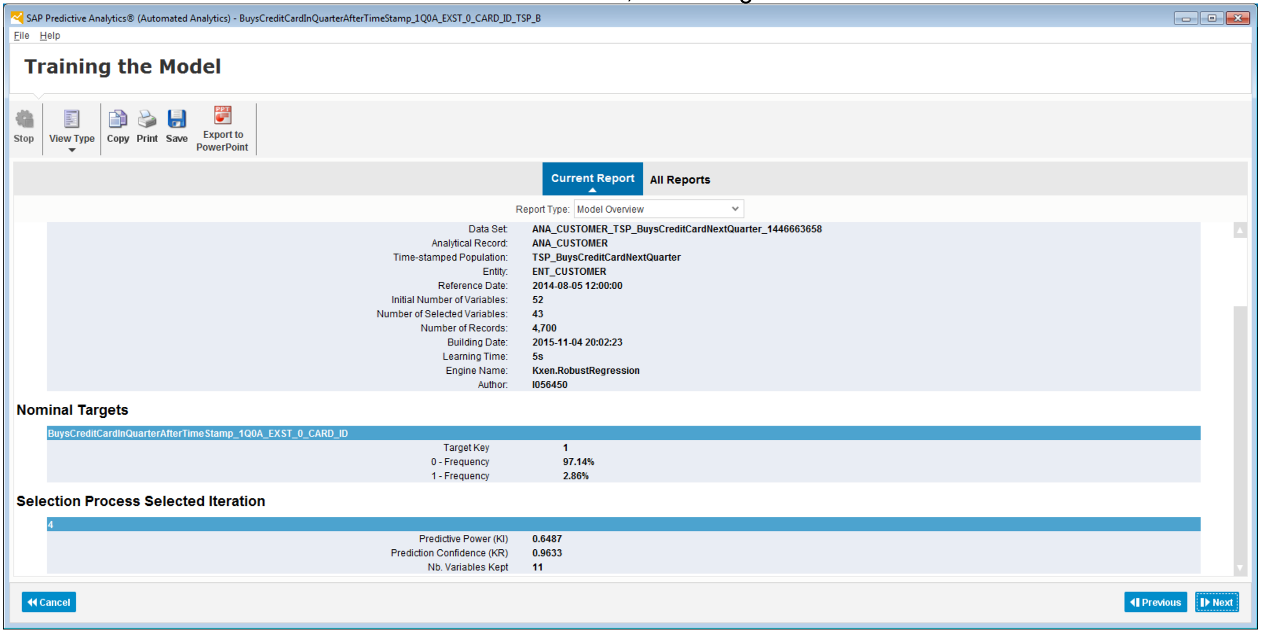
BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID

Weight Variable 0

Excluded Variables 8

KxId
KxTimeStamp
CLIENT_ID
DISTRICT_ID
DISP_ID
ACCOUNT_ID
DISTRICT_ID_2
DATE

The model has been greatly improved further! The Predictive Power incrased from 0.0.53 to 0.64 and the Prediction Confidence is now above the desired 0.95, indicating a robust model.


SAP Predictive Analytics® (Automated Analytics) - BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID_TSP_B
File
Help
Stop
View Type
Copy
Print
Save
Export to PowerPoint
Current Report
All Reports
Cancel
Previous
Next

Training the Model

Report Type: Model Overview

Data Set:	ANA_CUSTOMER_TSP_BuysCreditCardNextQuarter_1446663658
Analytical Record:	ANA_CUSTOMER
Time-stamped Population:	TSP_BuysCreditCardNextQuarter
Entity:	ENT_CUSTOMER
Reference Date:	2014-08-05 12:00:00
Initial Number of Variables:	52
Number of Selected Variables:	43
Number of Records:	4,700
Building Date:	2015-11-04 20:02:23
Learning Time:	5s
Engine Name:	Kxen.RobustRegression
Author:	I056450

Nominal Targets

BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID

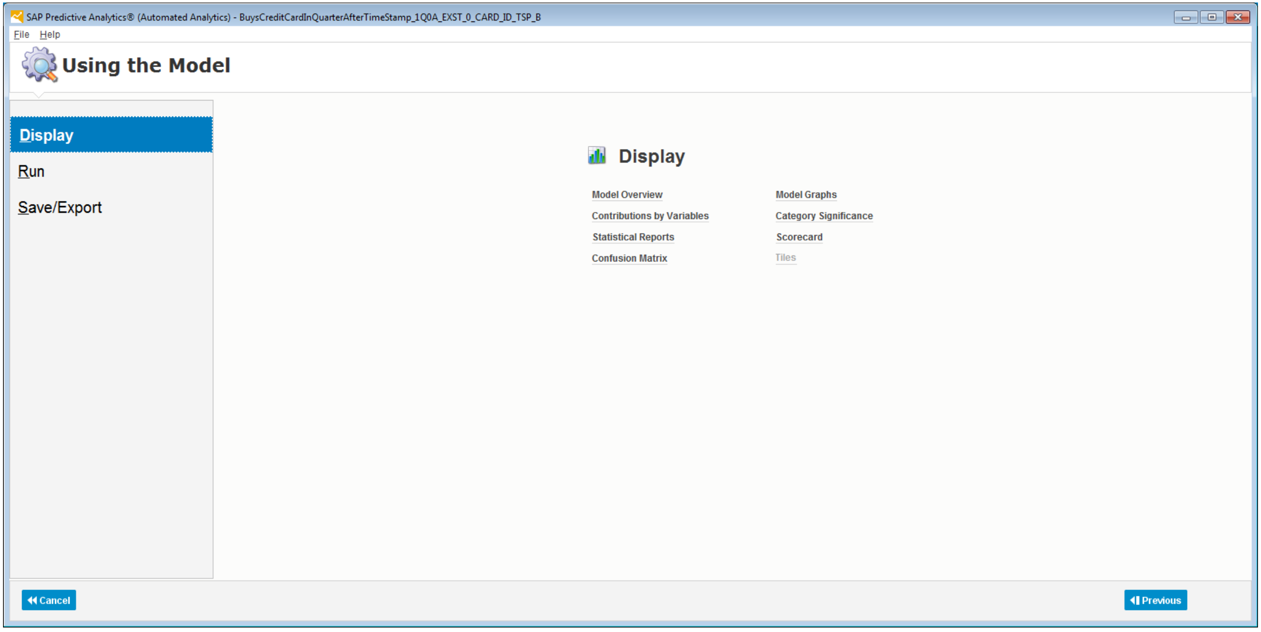
Target Key	1
0 - Frequency	97.14%
1 - Frequency	2.86%

Selection Process Selected Iteration

4

Predictive Power (KI)	0.6487
Prediction Confidence (KR)	0.9633
Nb. Variables Kept	11

Now we want to apply the trained model on new data. Which customer is likely to buy a credit card in the next quarter? Click “Next” to continue.



A short look into the “”Contribution Variables” shows the selected variables and their importance.

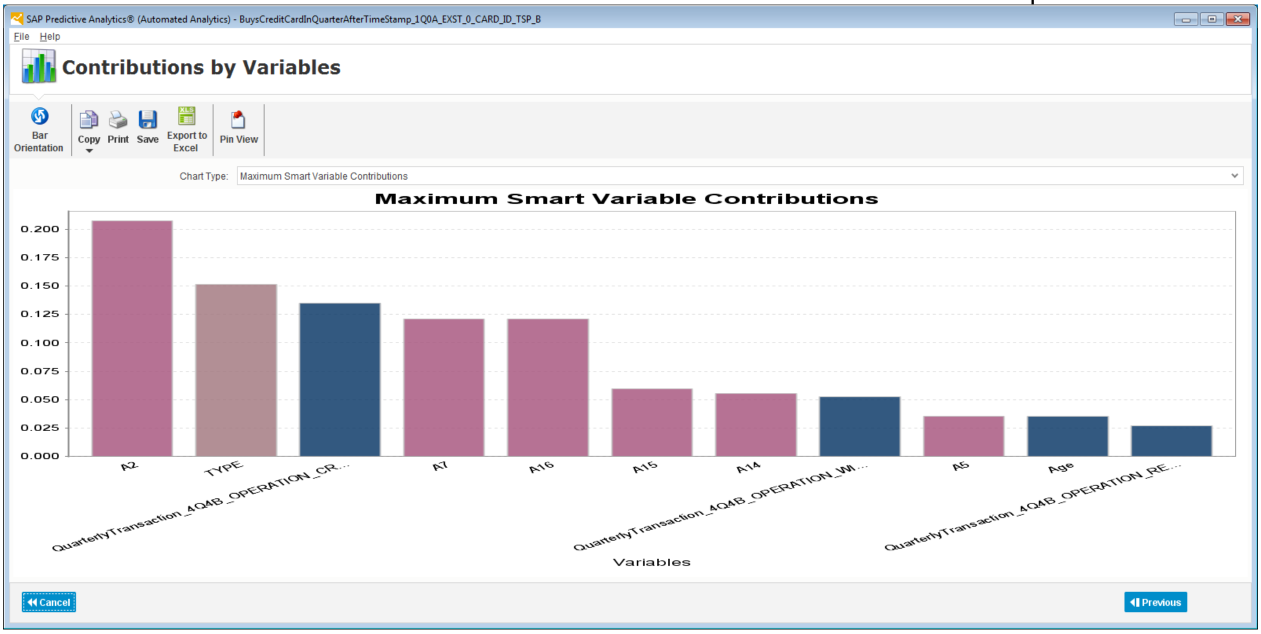
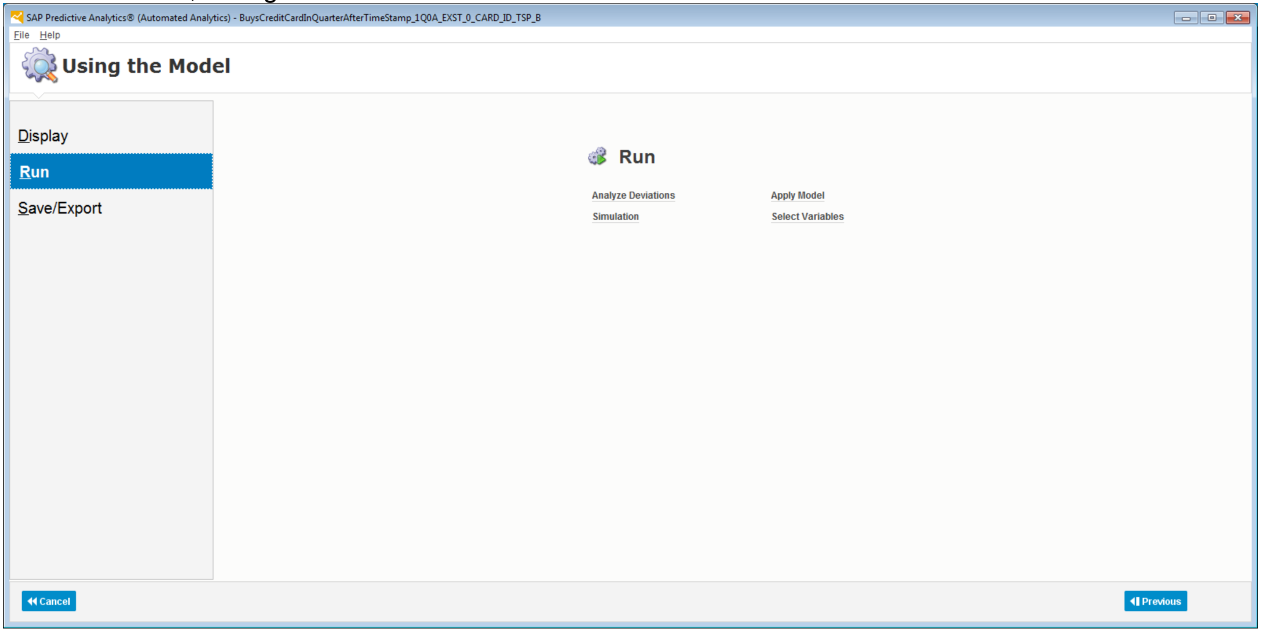

SAP Predictive Analytics® (Automated Analytics) - BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID_TSP_B
File Help
Contributions by Variables
Bar Orientation
Copy
Print
Save
Export to Excel
Pin View
Maximum Smart Variable Contributions
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
A2
TYPE
QuarterlyTransaction_4Q4B_OPERATION_CR...
A7
A16
A15
A14
QuarterlyTransaction_4Q4B_OPERATION_WI...
A5
Age
QuarterlyTransaction_4Q4B_OPERATION_RE...
Variables
Cancel
Previous

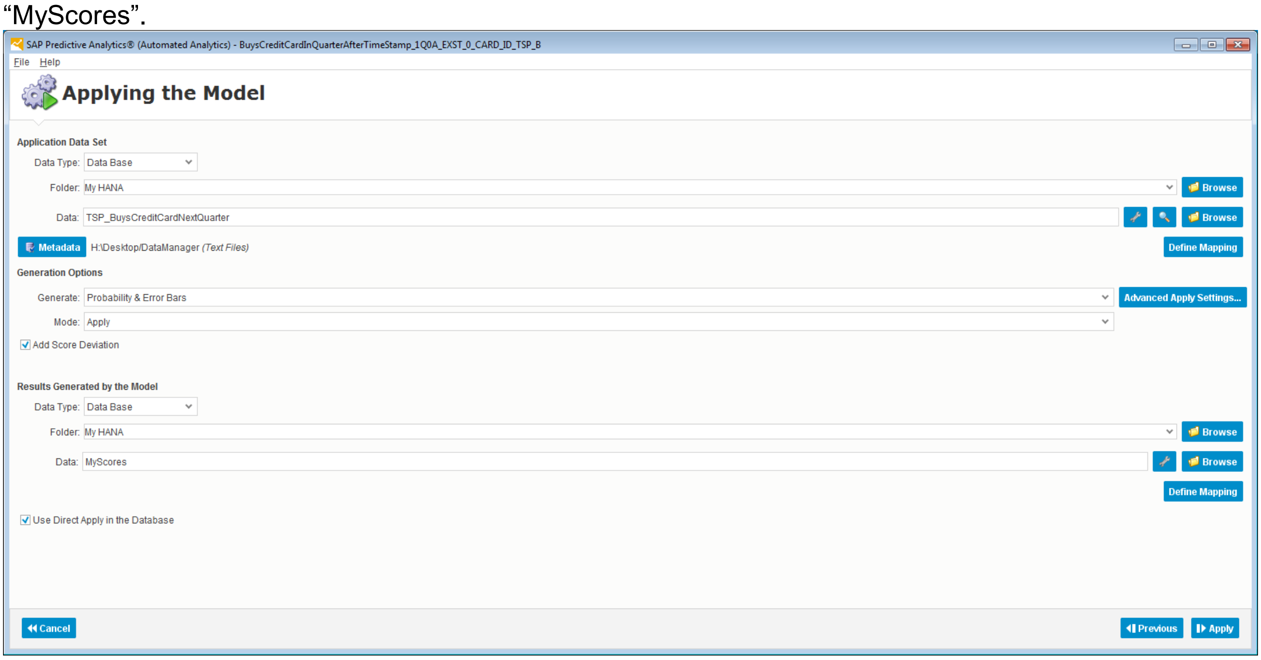
Chart Type: Maximum Smart Variable Contributions

Click “Previous”, then go into “Run” section.



SAP Predictive Analytics® (Automated Analytics) - BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID_TSP_B

Click “Apply Model” and set the following options. “Data” is the time-stamped population “TSP_BuysCreditCardNextQuarter”. Generate “Probability & Error Bars”. Write the results in a table called “MyScores”.



SAP Predictive Analytics® (Automated Analytics) - BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID_TSP_B

Applying the Model

Application Data Set

Data Type: Data Base

Folder: My HANA

Data: TSP_BuysCreditCardNextQuarter

H:\Desktop/DataManager *(Text Files)*

Generation Options

Generate: Probability & Error Bars

Mode: Apply

☑ Add Score Deviation

Results Generated by the Model

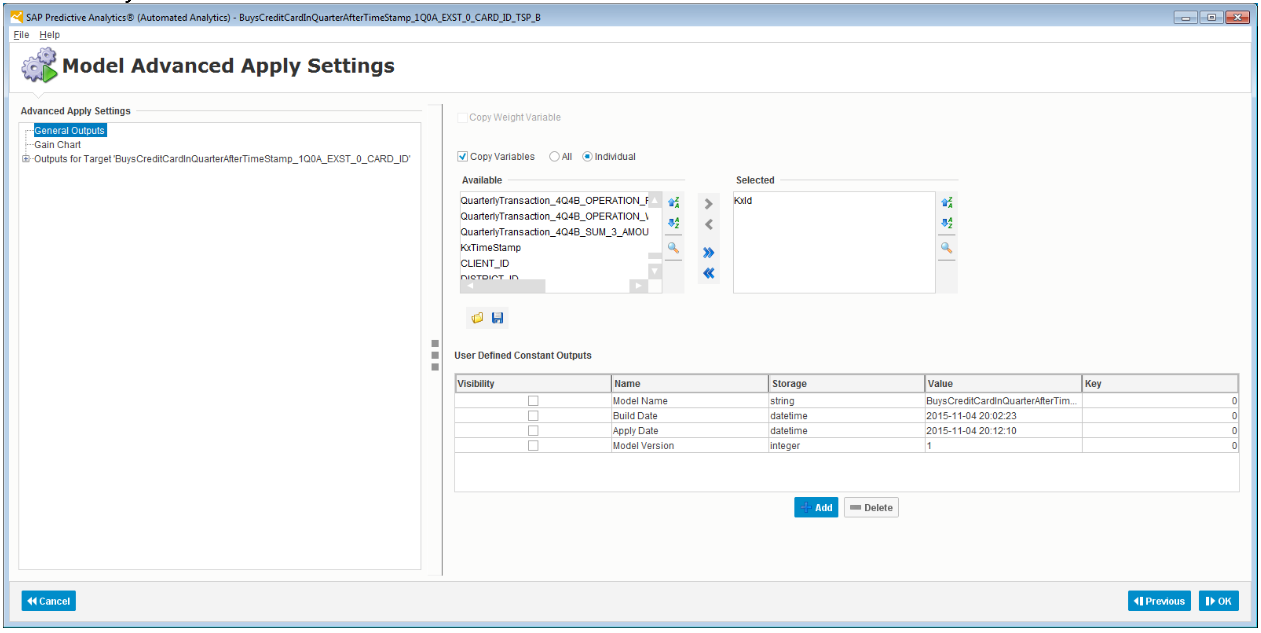
Data Type: Data Base

Folder: My HANA

Data: MyScores

☑ Use Direct Apply in the Database

Before proceeding, go into “Advanced Apply Settings…” and copy the “KxId” column into the output so you can identify the scores with the related clients.


SAP Predictive Analytics® (Automated Analytics) - BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID_TSP_B
File Help

Model Advanced Apply Settings

Advanced Apply Settings

- General Outputs
- Gain Chart
- Outputs for Target 'BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID'

□ Copy Weight Variable

☑ Copy Variables ○ All ● Individual

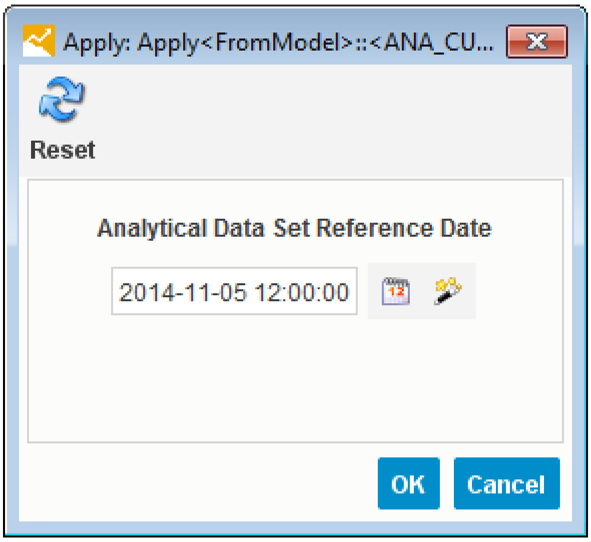
Available:
QuarterlyTransaction_4Q4B_OPERATION_F
QuarterlyTransaction_4Q4B_OPERATION_\
QuarterlyTransaction_4Q4B_SUM_3_AMOU
KxTimeStamp
CLIENT_ID
DISTRICT_ID

Selected:
KxId

User Defined Constant Outputs

Visibility	Name	Storage	Value	Key
□	Model Name	string	BuysCreditCardInQuarterAfterTim...	0
□	Build Date	datetime	2015-11-04 20:02:23	0
□	Apply Date	datetime	2015-11-04 20:12:10	0
□	Model Version	integer	1	0

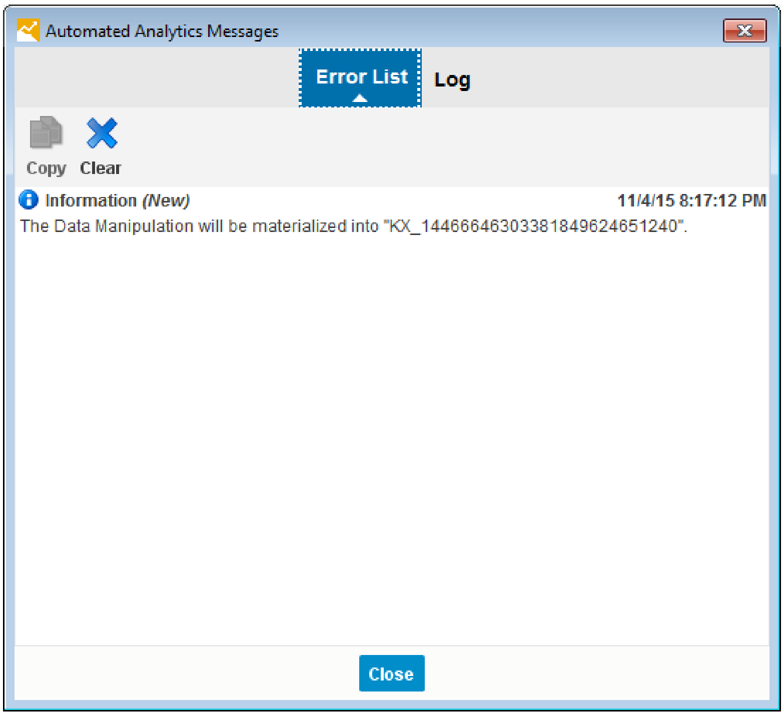
Click “OK” and “Apply”. Now enter a time-stamp that is 3 months after the previously used time-stamps. So pick 5th November 2014. The model learned from the past and now tries to predict a future it has not yet seen.



Analytical Data Set Reference Date

2014-11-05 12:00:00

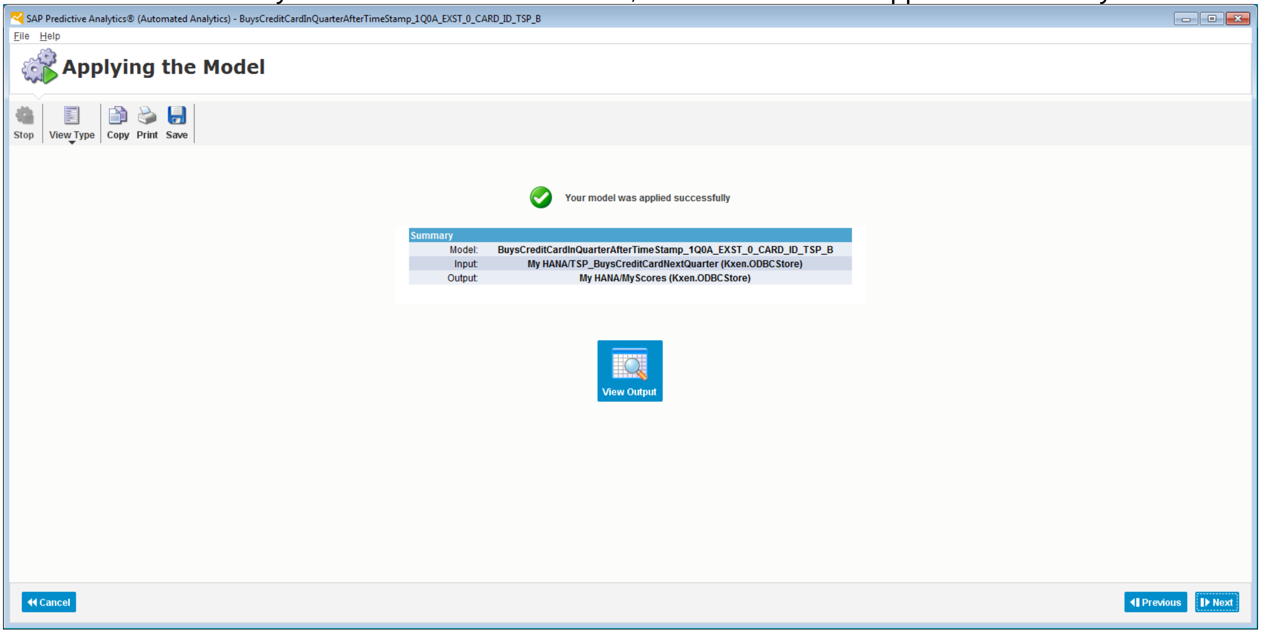
Click “OK”. You should get a confirmation that the data manipulations are materialized temporarily in a SAP HANA table.



Information *(New)* 11/4/15 8:17:12 PM

The Data Manipulation will be materialized into "KX_14466646303381849624651240".

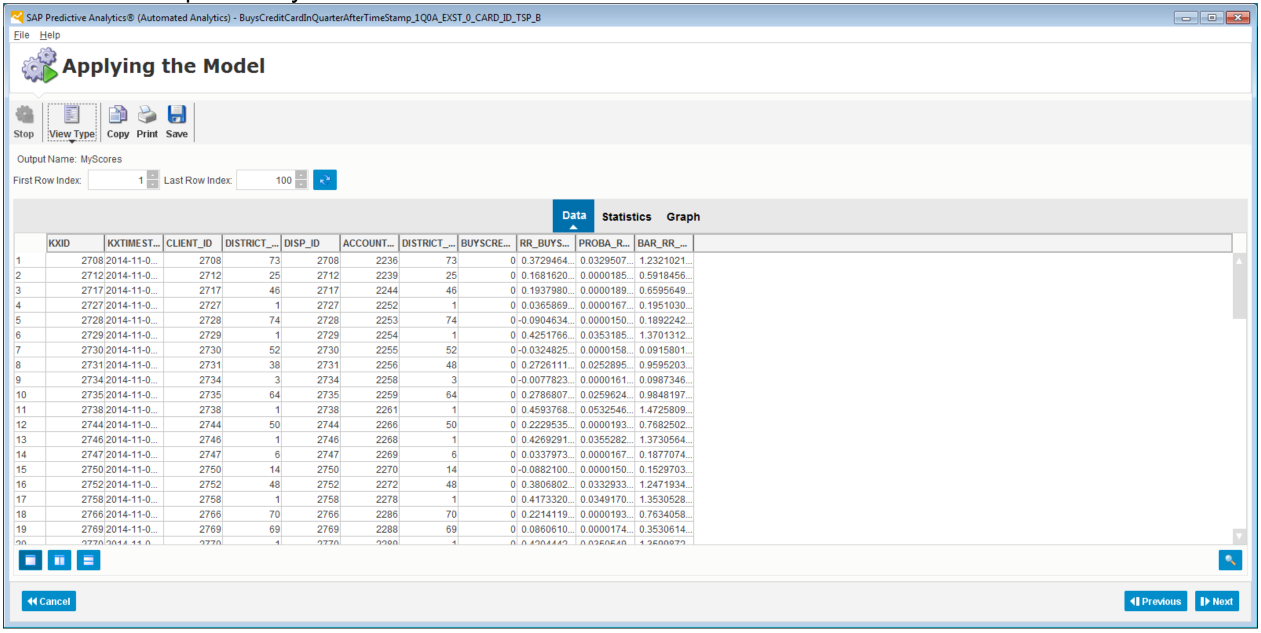
Close this window and you should see a confirmation, that the model was applied successfully.


SAP Predictive Analytics® (Automated Analytics) - BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID_TSP_B
File
Help
Applying the Model
Stop
View Type
Copy
Print
Save
View Output
Cancel
Previous
Next

✓ Your model was applied successfully

Summary	
Model:	BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID_TSP_B
Input:	My HANA/TSP_BuysCreditCardNextQuarter (Kxen.ODBCStore)
Output:	My HANA/MyScores (Kxen.ODBCStore)

Click “View Output” and you see the first rows of the results.


SAP Predictive Analytics® (Automated Analytics) - BuysCreditCardInQuarterAfterTimeStamp_1Q0A_EXST_0_CARD_ID_TSP_B
File Help
Applying the Model
Stop
View Type
Copy
Print
Save
Data
Statistics
Graph
Cancel
Previous
Next

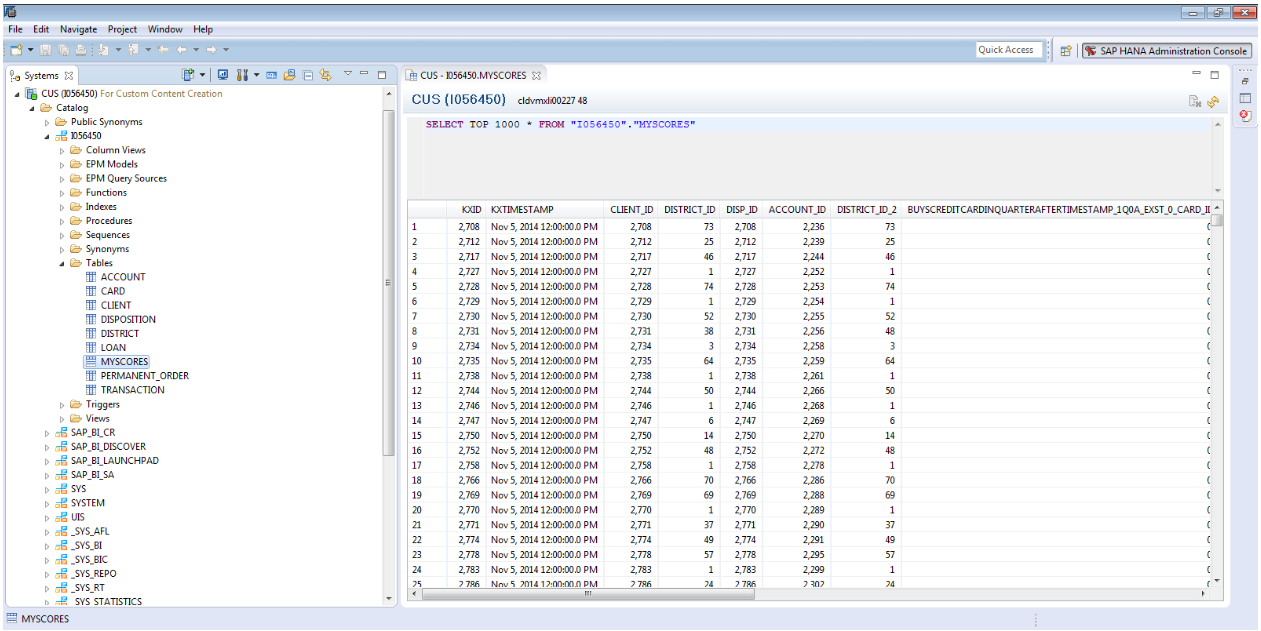
Output Name: MyScores

First Row Index: 1 Last Row Index: 100

	KXID	KXTIMEST...	CLIENT_ID	DISTRICT_...	DISP_ID	ACCOUNT...	DISTRICT_...	BUYSCRE...	RR_BUYS...	PROBA_R...	BAR_RR_...
1	2708	2014-11-0...	2708	73	2708	2236	73	0	0.3729464...	0.0329507...	1.2321021...
2	2712	2014-11-0...	2712	25	2712	2239	25	0	0.1681620...	0.0000185...	0.5918456...
3	2717	2014-11-0...	2717	46	2717	2244	46	0	0.1937980...	0.0000189...	0.6595649...
4	2727	2014-11-0...	2727	1	2727	2252	1	0	0.0365869...	0.0000167...	0.1951030...
5	2728	2014-11-0...	2728	74	2728	2253	74	0	-0.0904634...	0.0000150...	0.1892242...
6	2729	2014-11-0...	2729	1	2729	2254	1	0	0.4251766...	0.0353185...	1.3701312...
7	2730	2014-11-0...	2730	52	2730	2255	52	0	-0.0324825...	0.0000158...	0.0915801...
8	2731	2014-11-0...	2731	38	2731	2256	48	0	0.2726111...	0.0252895...	0.9595203...
9	2734	2014-11-0...	2734	3	2734	2258	3	0	-0.0077823...	0.0000161...	0.0987346...
10	2735	2014-11-0...	2735	64	2735	2259	64	0	0.2786807...	0.0259624...	0.9848197...
11	2738	2014-11-0...	2738	1	2738	2261	1	0	0.4593768...	0.0532546...	1.4725809...
12	2744	2014-11-0...	2744	50	2744	2266	50	0	0.2229535...	0.0000193...	0.7682502...
13	2746	2014-11-0...	2746	1	2746	2268	1	0	0.4269291...	0.0355282...	1.3730564...
14	2747	2014-11-0...	2747	6	2747	2269	6	0	0.0337973...	0.0000167...	0.1877074...
15	2750	2014-11-0...	2750	14	2750	2270	14	0	-0.0882100...	0.0000150...	0.1529703...
16	2752	2014-11-0...	2752	48	2752	2272	48	0	0.3806802...	0.0332933...	1.2471934...
17	2758	2014-11-0...	2758	1	2758	2278	1	0	0.4173320...	0.0349170...	1.3530528...
18	2766	2014-11-0...	2766	70	2766	2286	70	0	0.2214119...	0.0000193...	0.7634058...
19	2769	2014-11-0...	2769	69	2769	2288	69	0	0.0860610...	0.0000174...	0.3530614...

The colum “KXID” identifies the client. The column starting with “PROBA_RR” holds the estimated probability that the client will sign up for a credit card in the three months after 5th November 2011. You can use this information, to optimize your Marketing campaign.

Since the scores are written into a SAP HANA table, you can also see the results directly in the SAP HANA Studio.



CUS (I056450) cldvmxli00227 48

```
SELECT TOP 1000 * FROM "I056450"."MYSCORES"
```

	KXID	KXTIMESTAMP	CLIENT_ID	DISTRICT_ID	DISP_ID	ACCOUNT_ID	DISTRICT_ID_2	BUYSCREDITCARDINQUARTERAFTERTIMESTAMP_1Q0A_EXST_0_CARD_II
1	2,708	Nov 5, 2014 12:00:00.0 PM	2,708	73	2,708	2,236	73	(
2	2,712	Nov 5, 2014 12:00:00.0 PM	2,712	25	2,712	2,239	25	(
3	2,717	Nov 5, 2014 12:00:00.0 PM	2,717	46	2,717	2,244	46	(
4	2,727	Nov 5, 2014 12:00:00.0 PM	2,727	1	2,727	2,252	1	(
5	2,728	Nov 5, 2014 12:00:00.0 PM	2,728	74	2,728	2,253	74	(
6	2,729	Nov 5, 2014 12:00:00.0 PM	2,729	1	2,729	2,254	1	(
7	2,730	Nov 5, 2014 12:00:00.0 PM	2,730	52	2,730	2,255	52	(
8	2,731	Nov 5, 2014 12:00:00.0 PM	2,731	38	2,731	2,256	48	(
9	2,734	Nov 5, 2014 12:00:00.0 PM	2,734	3	2,734	2,258	3	(
10	2,735	Nov 5, 2014 12:00:00.0 PM	2,735	64	2,735	2,259	64	(
11	2,738	Nov 5, 2014 12:00:00.0 PM	2,738	1	2,738	2,261	1	(
12	2,744	Nov 5, 2014 12:00:00.0 PM	2,744	50	2,744	2,266	50	(
13	2,746	Nov 5, 2014 12:00:00.0 PM	2,746	1	2,746	2,268	1	(
14	2,747	Nov 5, 2014 12:00:00.0 PM	2,747	6	2,747	2,269	6	(
15	2,750	Nov 5, 2014 12:00:00.0 PM	2,750	14	2,750	2,270	14	(
16	2,752	Nov 5, 2014 12:00:00.0 PM	2,752	48	2,752	2,272	48	(
17	2,758	Nov 5, 2014 12:00:00.0 PM	2,758	1	2,758	2,278	1	(
18	2,766	Nov 5, 2014 12:00:00.0 PM	2,766	70	2,766	2,286	70	(
19	2,769	Nov 5, 2014 12:00:00.0 PM	2,769	69	2,769	2,288	69	(
20	2,770	Nov 5, 2014 12:00:00.0 PM	2,770	1	2,770	2,289	1	(
21	2,771	Nov 5, 2014 12:00:00.0 PM	2,771	37	2,771	2,290	37	(
22	2,774	Nov 5, 2014 12:00:00.0 PM	2,774	49	2,774	2,291	49	(
23	2,778	Nov 5, 2014 12:00:00.0 PM	2,778	57	2,778	2,295	57	(
24	2,783	Nov 5, 2014 12:00:00.0 PM	2,783	1	2,783	2,299	1	(
25	2,786	Nov 5, 2014 12:00:00.0 PM	2,786	24	2,786	2,302	24	(

Well done! You have used the Data Manager to improve a classification model created with “Automated Analytics”. The additional insight from the newly created variables based around the understanding of time has provided a much clearer picture, resulting in a stronger and more robust model.

HINTS AND TIPPS

Some basic points, which you might find helpful when working with Data Manager

- Data Manager needs to connect to a database with SQL. It cannot be used directly on flat files.
- It is preferred to use capital letters for all table names and column names, ie use CLIENT_ID instead of ClientID.
- The entity table must not contain duplicates. So you cannot base the entity on a transaction table, which will have duplicates.
- When working with date columns, typically the date itself should not be included in the model. Instead derive further columns from it, ie a duration from the time-stamp to the date variable. This is the case when calculating the age or the duration of how long an account has been active for instance.
- Only the target variable must use information from after the timestamp
- It is best practice to have 1000 positive cases to train a model on. Less cases can be ok, just try to get the Predictive Confidence (KR) above 0.95. This value can often be improved through additional variables, which impact the model positively.

APPENDIX

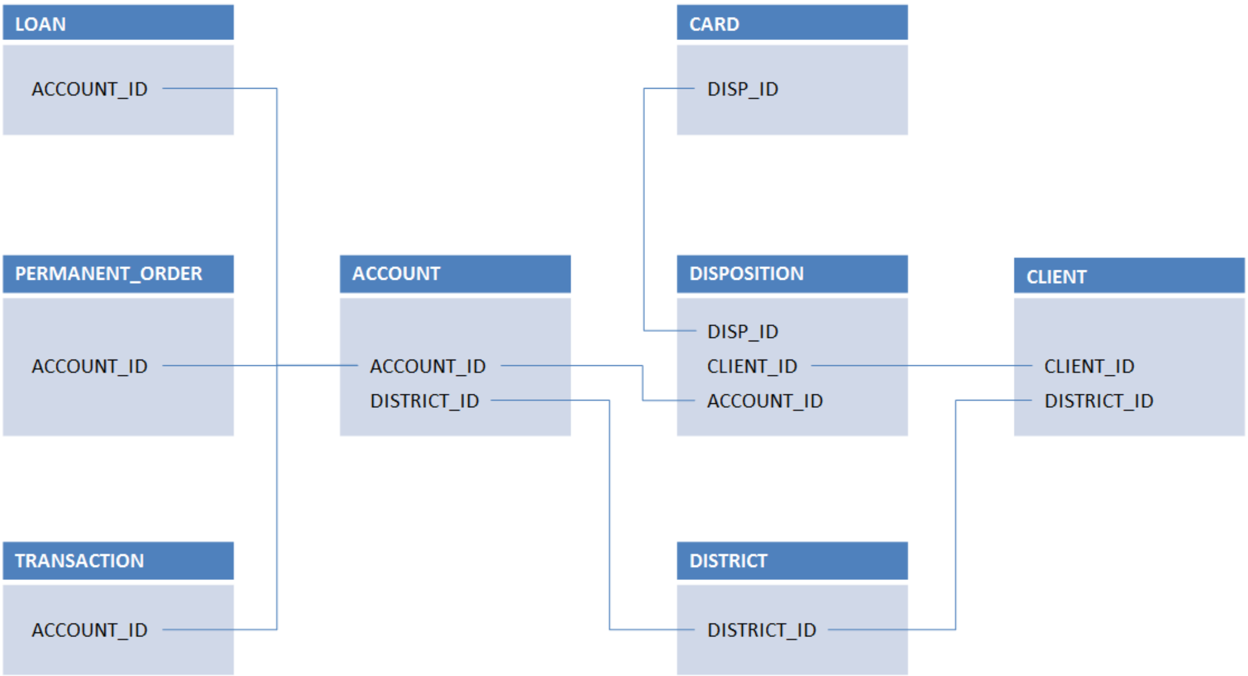
Data Description

This chapter describes the data we are working with. Most information in this chapter is taken from the PKDD ’99 website[8].

The following tables are available:

Table	Numer of Rows	
ACCOUNT	4500	static characteristics of an account
CLIENT	5369	characteristics of a client
DISPOSITION	5369	relates together a client with an account i.e. this relation describes the rights of clients to operate accounts
PERMANENT_ORDER	6471	describes characteristics of a payment order
TRANSACTION	1056320	describes one transaction on an account
LOAN	682	describes a loan granted for a given account
CARD	892	describes a credit card issued to an account
DISTRICT	77	describes demographic characteristics of a district

The database model is as follows:



[8] http://lisp.vse.cz/pkdd99

The tables contain the following columns.

ACCOUNT

Column	Description	Content
ACCOUNT_ID	Account identifier	
DISTRICT_ID	Branch's district identifier	
FREQUENCY	frequency of issuance of statements	'AFTER TRANSACTION', 'MONTHLY', 'WEEKLY'
DATE	Date of account opening	

CLIENT

Column	Description	Content
CLIENT_ID	Client identifier	
DISTRICT_ID	Branch's district identifier	
GENDER	Gender	'FEMALE' , 'MALE'
DATE_BIRTH	Date of birth	

DISPOSITION

Column	Description	Content
DISP_ID	Disposition identifier	
CLIENT_ID	Client identifier	
ACCOUNT_ID	Account identifier	
TYPE	Type of disposition	'DISPONENT', 'OWNER'

PERMANENT_ORDER

Column	Description	Content
ORDER_ID	Order identifier	
ACCOUNT_ID	Account identifier	
BANK_TO	Bank of the recipient	‘AB’, ‘CD’, ‘EF’, ‘GH’, ‘IJ’, ‘KL’, ‘MN’, ‘OP’, ‘QR’, ‘ST’, ‘UV’, ‘WX’, ‘YZ’
ACCOUNT_TO	Account of the recipient	
AMOUNT	Debited amount	
K_SYMBOL	Characterization of the payment	‘HOUSEHOLD PAYMENT’, ‘INSURANCE PAYMENT’, ‘LEASING’, ‘LOAN PAYMENT’, <Null>

TRANSACTION

Column	Description	Content
TRANS_ID	Transaction identifier	
ACCOUNT_ID	Account identifier	
TYPE	Transaction type	‘CREDIT’, ‘VYBER’, ‘WITHDRAWAL’
OPERATION	Mode of transaction	‘COLLECTION FROM ANOTHER BANK’, ‘CREDIT CARD WITHDRAWAL’, ‘CREDIT IN CASH’, ‘REMITTANCE TO ANOTHER BANK’, ‘WITHDRAWAL IN CASH’, <Null>
AMOUNT	Amount of money	
BALANCE	Balance after transaction	
K_SYMBOL	Characterization of the transaction	‘INTEREST CREDITED’, ‘PAYMENT FOR STATEMENT’, ‘HOUSEHOLD’, ‘OLD-AGE PENSION’, ‘INSURANCE PAYMENT’, ‘LOAN PAYMENT’, ‘SANCTION INTEREST IF NEGATIVE BALANCE’, <Null>
BANK	Bank of the partner	‘AB’, ‘CD’, ‘EF’, ‘GH’, ‘IJ’, ‘KL’, ‘MN’, ‘OP’, ‘QR’, ‘ST’, ‘UV’, ‘WX’, ‘YZ’, <Null>
ACCOUNT	Account of the partner	
DATE	Date of the transaction	

LOAN

Column	Description	Content
LOAN_ID	Loan identifier	
ACCOUNT_ID	Account identifier	
AMOUNT	Amount of money	
DURATION	Duration of the loan in months	
PAYMENTS	Monthly payments	
STATUS	Status of paying off the loan	‘A’: contract finished, no problems ‘B’: contract finished, loan not paid ‘C’: running contract, OK so far ‘D’: running contract, client in debt
DATE	Date when the loan was granted	

CARD

Column	Description	Content
CARD_ID	Credit card identifier	
DISP_ID	Disposition identifier	
TYPE	Type of card	‘classic’, ‘gold’, ‘junior’
ISSUED	Date when the card was issued	

DISTRICT

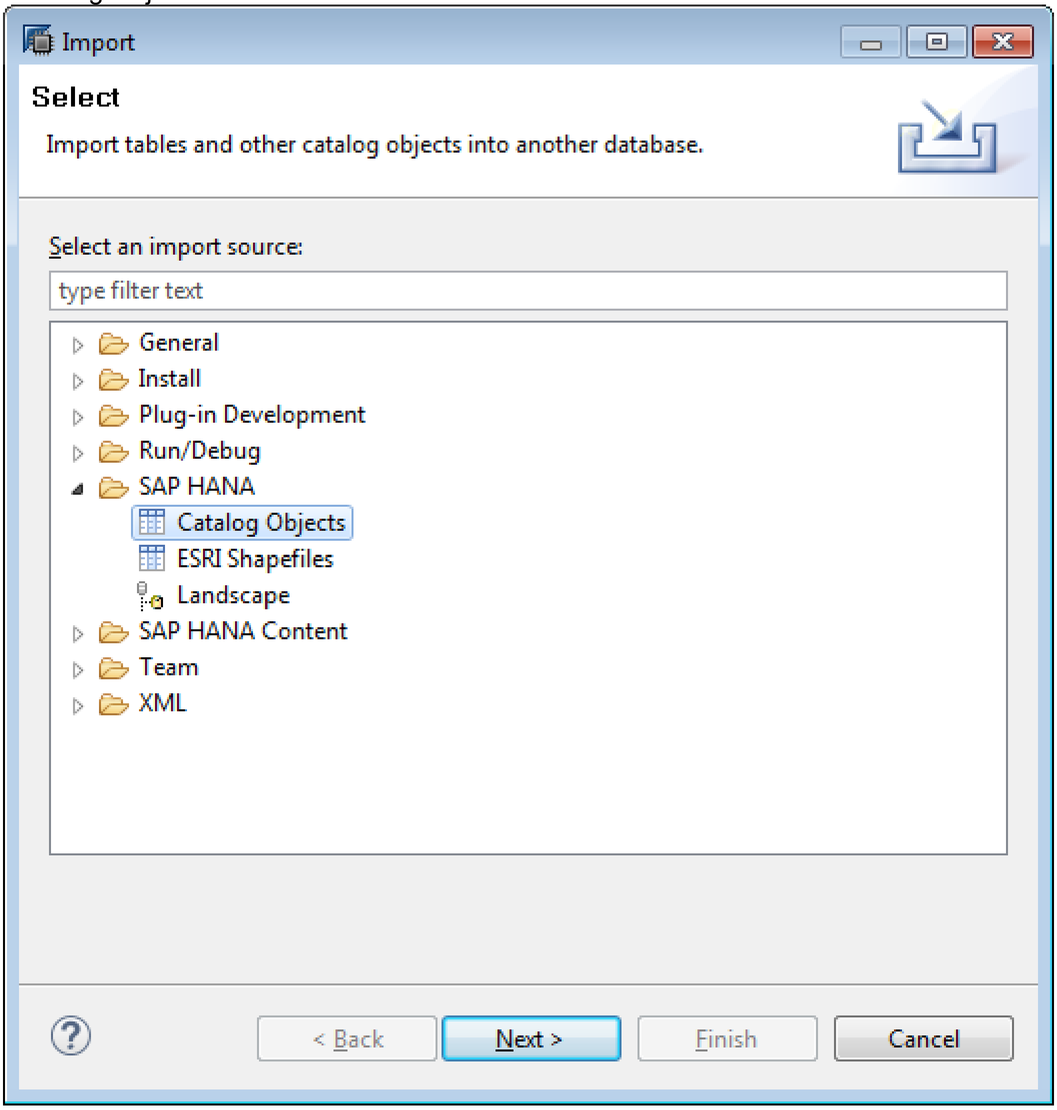
Column	Description	Content
DISTRICT_ID	Branch’s district identifier	
A2	District name	‘Benesov’, ‘Beroun’, ‘Blansko’, …
A3	District region	‘Prague’, ‘central Bohemia’, ‘east Bohemia’, …
A4	Number of inhabitants	
A5	Number of municipalities with inhabitants < 499	
A6	Number of municipalities with inhabitants 500-1999	
A7	Number of municipalities with inhabitants 2000-9999	
A8	Number of municipalities with inhabitants >10000	
A9	Number of cities	
A10	Ratio of urban inhabitants	
A11	Average salary	
A12	Unemployment rate previous year	
A13	Unemployment rate most recent year	
A14	Number of entrepreneurs per 1000 inhabitants	
A15	Number of committed crimes previous year	
A16	Number of committed crimes most recent year	

Data Load
You have two options to load the data into SAP HANA. The easiest would be to import the whole content with a few clicks and you have the content exactly as described in this document as option 1 (Import SAP HANA Content). This upload includes a number of transformations made on the original data, for instance a translation to English values and an update of the dates to recent values.

Alternatively, you can also manually import and transform the data as described in option 2 (Manually Import and Transform).

Option 1: Import SAP HANA Content
You can download the SAP HANA tables from GitHub[9]. Extract the zip file.

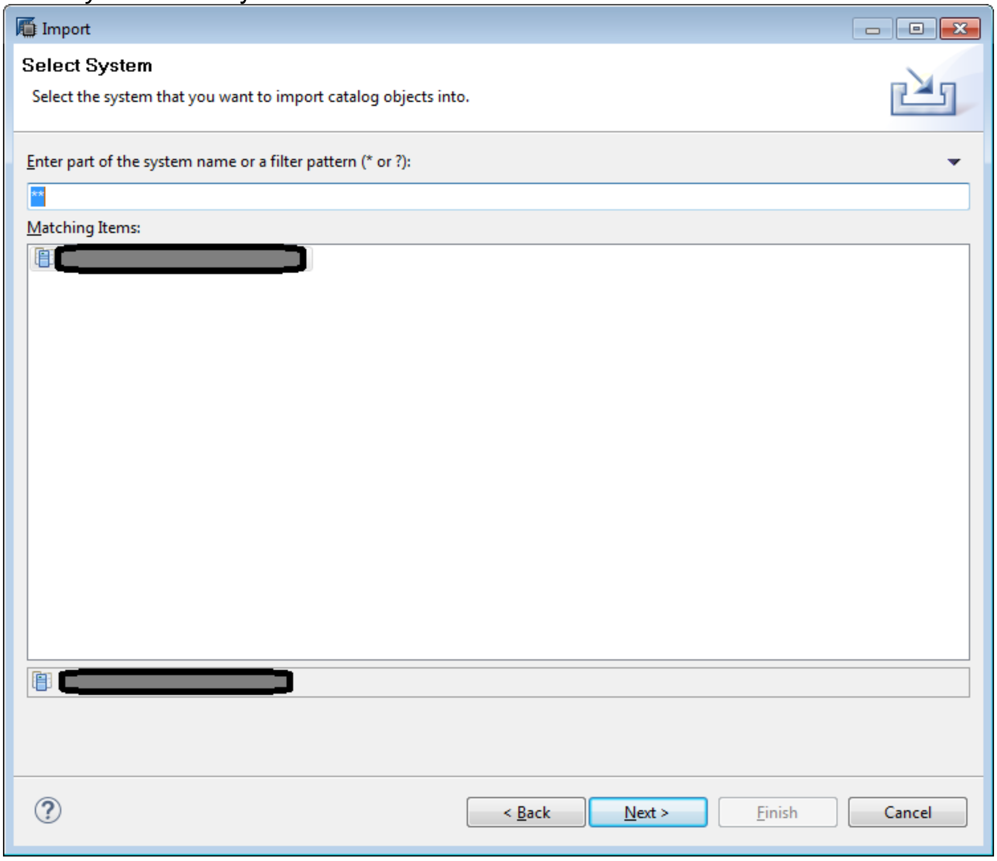
Now import the content into SAP HANA. In HANA Studio, select “File” → “Import” → “SAP HANA” → “Catalog Objects”.



Hit “Next”.

[9] Data for Tutorial: https://github.com/AndreasForster/Predictive/raw/master/DataManagerTutorialData.zip

Select your HANA system and click “Next”.



Import

Select System

Select the system that you want to import catalog objects into.

Enter part of the system name or a filter pattern (* or ?):

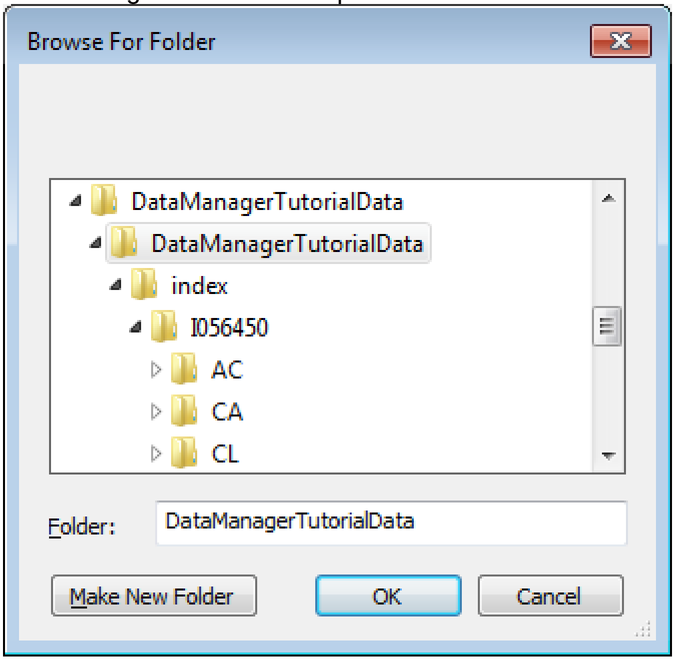
**

Matching Items:

[redacted]

[redacted]

On the next screen select “Import Catalog Objects from current client”. Select the folder you have extracted DataManagerTutorialData.zip into.



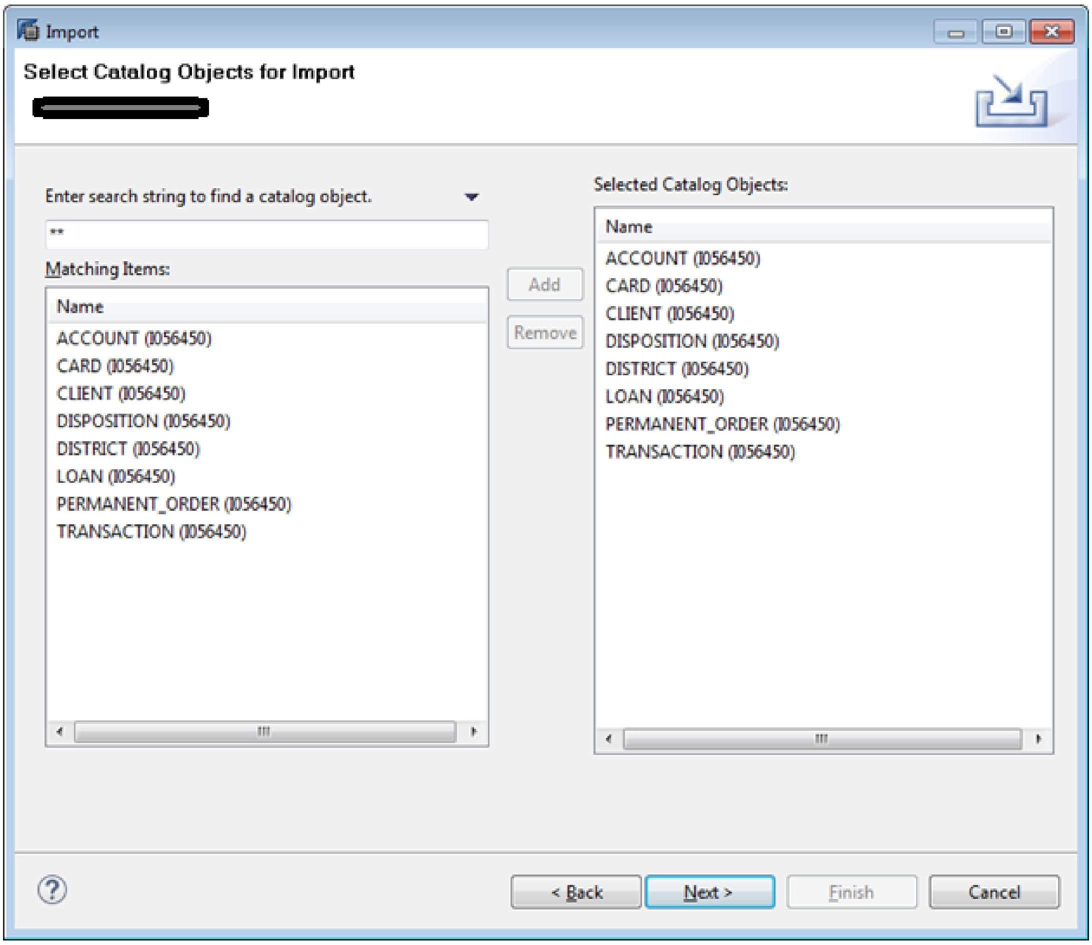
Browse For Folder

- DataManagerTutorialData
  - DataManagerTutorialData
    - index
      - I056450
        - AC
        - CA
        - CL

Folder: DataManagerTutorialData

Click “Next”.

Select all tables



Select Catalog Objects for Import

[redacted]

Enter search string to find a catalog object.

**

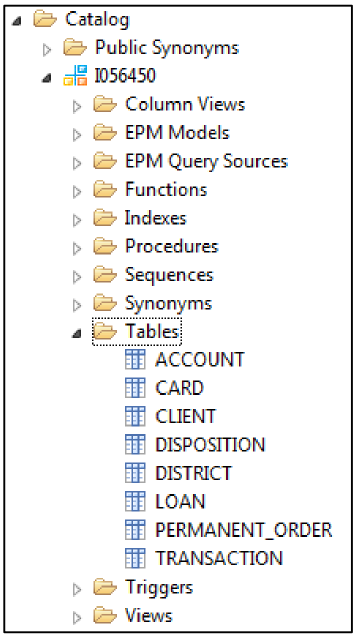
Matching Items:

Name
ACCOUNT (I056450)
CARD (I056450)
CLIENT (I056450)
DISPOSITION (I056450)
DISTRICT (I056450)
LOAN (I056450)
PERMANENT_ORDER (I056450)
TRANSACTION (I056450)

Selected Catalog Objects:

Name
ACCOUNT (I056450)
CARD (I056450)
CLIENT (I056450)
DISPOSITION (I056450)
DISTRICT (I056450)
LOAN (I056450)
PERMANENT_ORDER (I056450)
TRANSACTION (I056450)

Click “Next” and “Finish”. The tables should now be loaded.



- Catalog
  - Public Synonyms
  - I056450
    - Column Views
    - EPM Models
    - EPM Query Sources
    - Functions
    - Indexes
    - Procedures
    - Sequences
    - Synonyms
    - Tables
      - ACCOUNT
      - CARD
      - CLIENT
      - DISPOSITION
      - DISTRICT
      - LOAN
      - PERMANENT_ORDER
      - TRANSACTION
    - Triggers
    - Views

Option 2: Manually Import and Transform

Download original data and Upload into SAP HANA

In order to download and transform the original data, follow these steps. However, I strongly recommend to simply upload the prepared package into SAP HANA whenever possible (see above for Option 1).

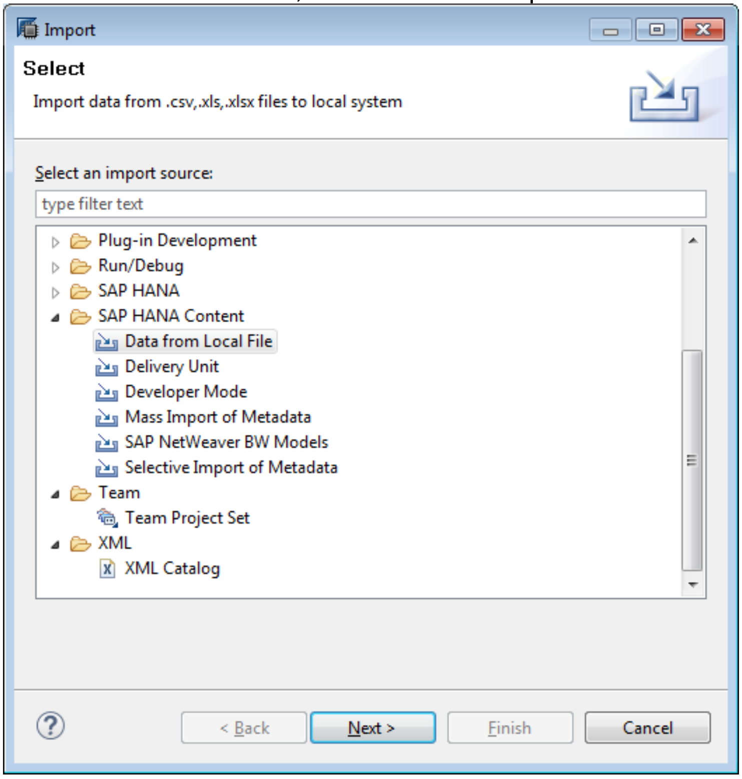
Step 1:
Download the raw data from the “Past ECML/PKDD Discovery Challenges” 10. From the PKD ’99 challenge select “Data” → “Original data download” → “Download the Financial Data”

Step 2:
Extract the data:berka.zip file and rename all file extensions from .asc to .csv.

Step 3:
One row in the district.csv file contains a ‘?’ in two columns to indicate missing data. Remove these question-marks so that the column is automatically identified as numerical.

Step 4:
Load each file as new table into SAP HANA. This can be done for instance with the SAP HANA Studio. Carry out the following steps for each table:

In the SAP HANA Studio, select “File” → “Import” → “SAP HANA Content” → “Data from Local File”.



Import

Select

Import data from .csv,.xls,.xlsx files to local system

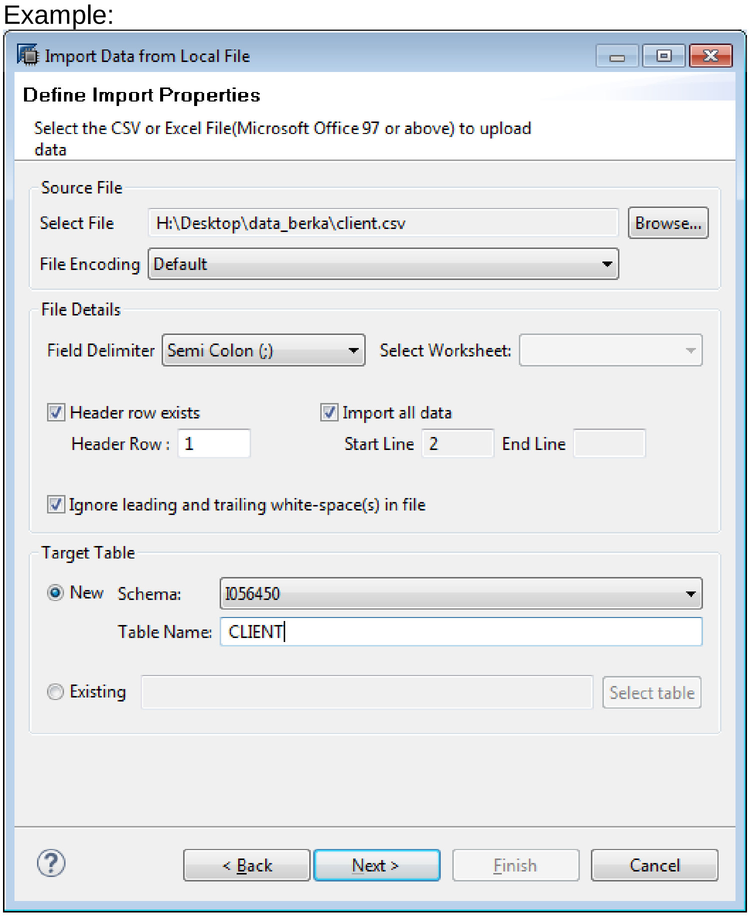
Select an import source:

[10] Past Challenges, http://lisp.vse.cz/challenge/PAST/index2.htm

Click “Next” → Select the Target System (most likely you will only see one) → Now specify the table to load.

- Select the .csv file
- Set “Field Delimiter” to “Semi Colon”
- Tick “Header row exists”
- Tick “Import all data”
- Select the database Schema from the dropdown
- As “Table Name” enter the file name (without file extension) in upper case, ie CLIENT

Example:



Import Data from Local File

Define Import Properties

Select the CSV or Excel File(Microsoft Office 97 or above) to upload data

Source File

Select File: H:\Desktop\data_berka\client.csv

File Encoding: Default

File Details

Field Delimiter: Semi Colon (;)   Select Worksheet:

☑ Header row exists   Header Row : 1

☑ Import all data   Start Line 2   End Line

☑ Ignore leading and trailing white-space(s) in file

Target Table

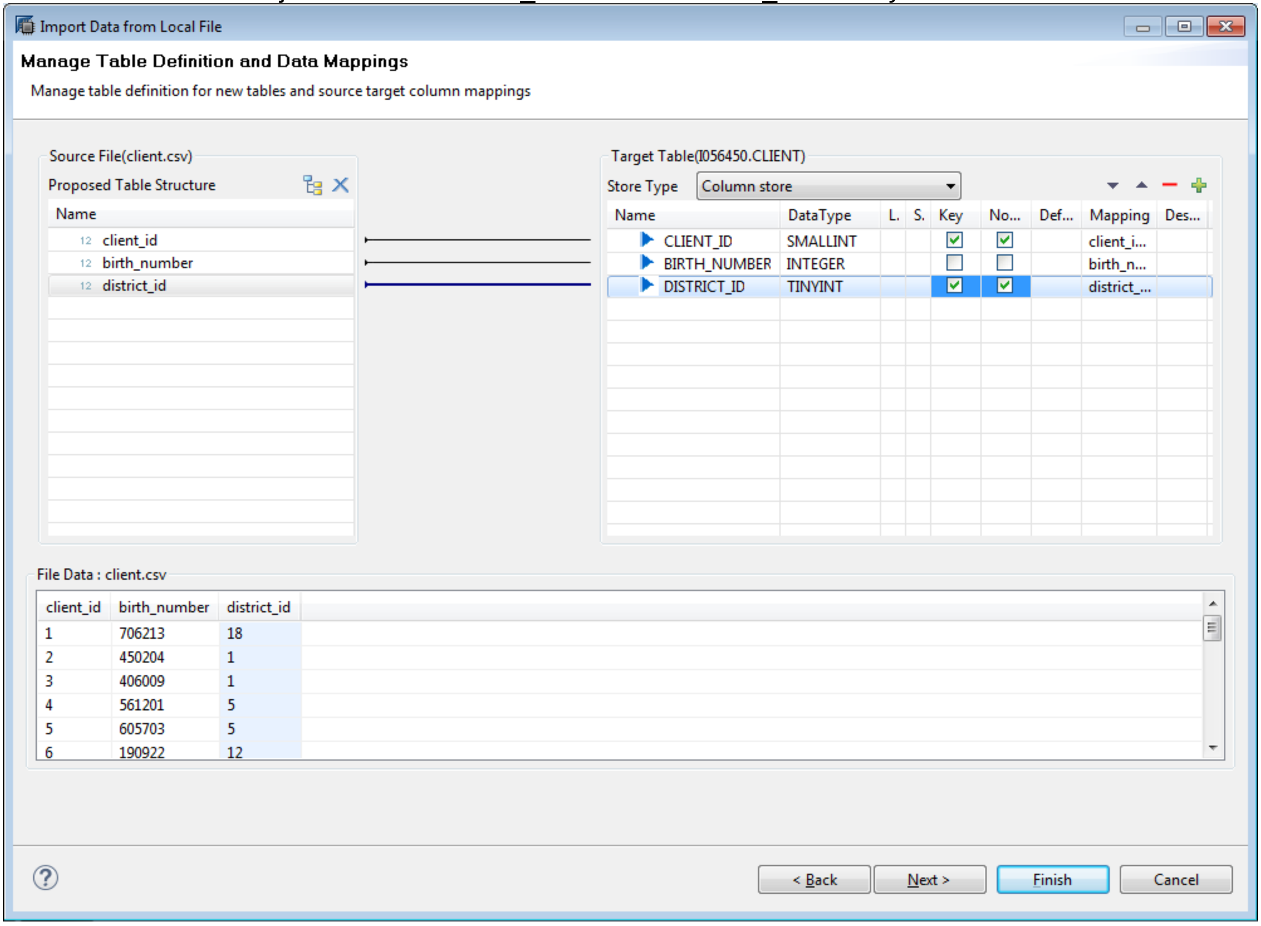
● New   Schema: I056450

Table Name: CLIENT

○ Existing

Click “Next” and change the column names from lower case to upper case. If indicated in the table below, you may have to set additional settings, ie a Key or different data types.

For the “CLIENT” table just set the “CLIENT_ID” and “DISTRICT_ID” as Keys.



Manage Table Definition and Data Mappings

Manage table definition for new tables and source target column mappings

Source File(client.csv)

Proposed Table Structure

Name
client_id
birth_number
district_id

Target Table(I056450.CLIENT)

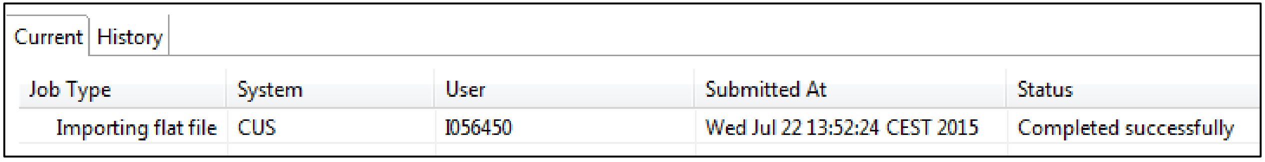
Store Type: Column store

Name	DataType	L.	S.	Key	No...	Def...	Mapping	Des...
CLIENT_ID	SMALLINT			☑	☑		client_i...	
BIRTH_NUMBER	INTEGER			□	□		birth_n...	
DISTRICT_ID	TINYINT			☑	☑		district_...	

File Data : client.csv

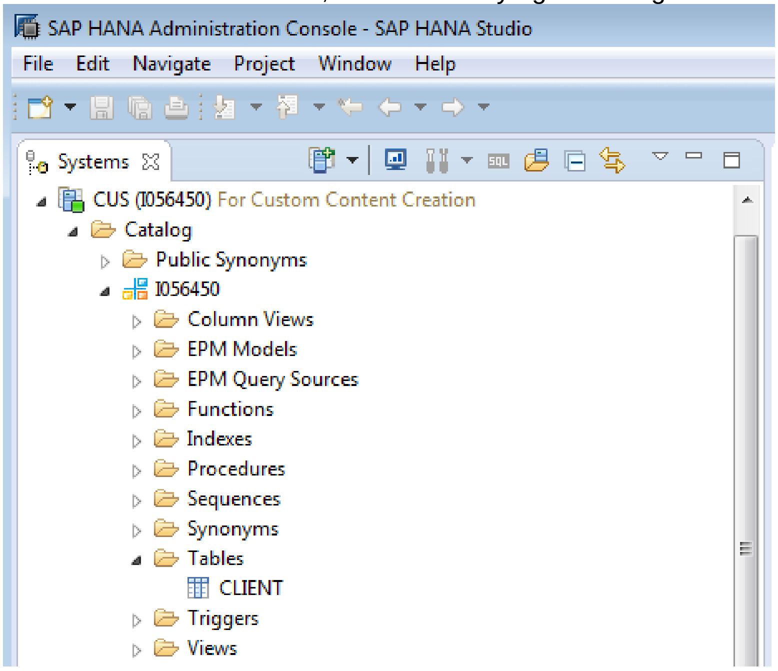
client_id	birth_number	district_id
1	706213	18
2	450204	1
3	406009	1
4	561201	5
5	605703	5
6	190922	12

Click “Finish” and you should see a success status in the job log.



Job Type	System	User	Submitted At	Status
Importing flat file	CUS	I056450	Wed Jul 22 13:52:24 CEST 2015	Completed successfully

The table has been created and the data has been uploaded. You can see the table in the Systems browser. You can now view the data, for instance by right-clicking on the table to open the data preview.

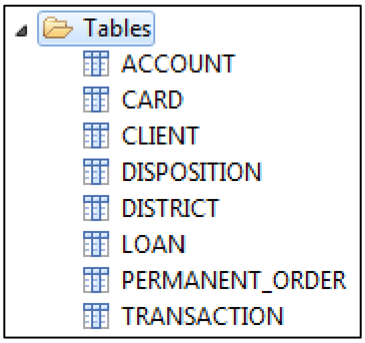

SAP HANA Administration Console - SAP HANA Studio
File Edit Navigate Project Window Help
Systems

- CUS (I056450) For Custom Content Creation
  - Catalog
    - Public Synonyms
    - I056450
      - Column Views
      - EPM Models
      - EPM Query Sources
      - Functions
      - Indexes
      - Procedures
      - Sequences
      - Synonyms
      - Tables
        - CLIENT
      - Triggers
      - Views

These are the specific table settings you have to apply, in addition to changing the table and column names from lower case to upper case.

File Name	Table Name	Key Columns	Additional
account.csv	ACCOUNT	ACCOUNT_ID, DISTRICT_ID	
card.csv	CARD	CARD_ID, DISP_ID	
client.csv	CLIENT	CLIENT_ID, DISTRICT_ID	
disp.csv	DISPOSITION	ACCOUNT_ID, CLIENT_ID, DISP_ID	
dstrict.csv	DISTRICT	DISTRICT_ID	Set name of A1 to DISTRICT_ID
loan.csv	LOAN	ACCOUNT_ID, LOAN_ID	
order.csv	PERMANENT_ORDER	ACCOUNT_ID, ORDER_ID	Set type of ORDER_ID to INTEGER
trans.csv	TRANSACTION	ACCOUNT_ID, TRANS_ID	Set length of K_SYMBOL to 17

You should now see all tables in SAP HANA.

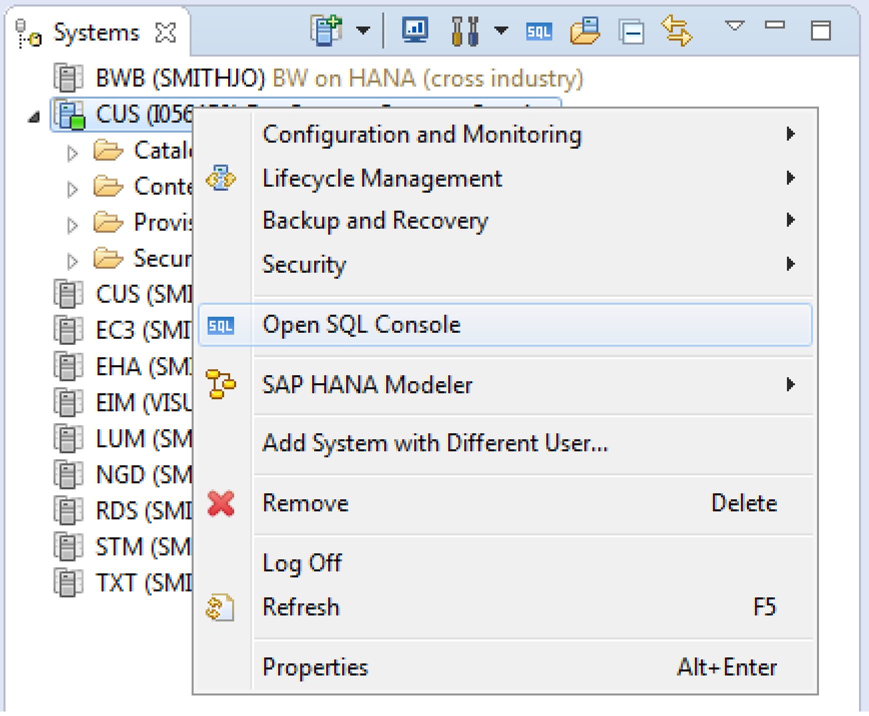


Tables
- ACCOUNT
- CARD
- CLIENT
- DISPOSITION
- DISTRICT
- LOAN
- PERMANENT_ORDER
- TRANSACTION

Data Transformation

Some of the data needs some minor transformation. The columns with dates for instance need to be transformed from numeric type to Date type.

Open the “SQL Console” in the SAP HANA Studio to execute the following SQL code, using your schema name instead of “I056450”. Execution can be started with the green arrow on the top right hand side.



```
/*
Table: ACCOUNT
Turn "DATE" column into DATE format
*/
ALTER TABLE "I056450"."ACCOUNT" ADD ("DATE_TEMP" NVARCHAR(6));
UPDATE "I056450"."ACCOUNT"  SET "DATE_TEMP" = TO_NVARCHAR("DATE");
ALTER TABLE "I056450"."ACCOUNT" DROP ("DATE");
ALTER TABLE "I056450"."ACCOUNT" ADD ("DATE" DATE);
UPDATE "I056450"."ACCOUNT"  SET "DATE" = TO_DATE("DATE_TEMP", 'YYMMDD');
ALTER TABLE "I056450"."ACCOUNT" DROP ("DATE_TEMP");
```

```
/*
Table: CARD
Turn "ISSUED" column into DATE format
*/
ALTER TABLE "I056450"."CARD" ADD ("ISSUED_TEMP" NVARCHAR(15));
UPDATE "I056450"."CARD"  SET "ISSUED_TEMP" = "ISSUED";
ALTER TABLE "I056450"."CARD" DROP ("ISSUED");
ALTER TABLE "I056450"."CARD" ADD ("ISSUED" DATE);
UPDATE "I056450"."CARD"  SET "ISSUED" =
      TO_DATE(LEFT(TO_NVARCHAR("ISSUED_TEMP"), 6), 'YYMMDD');
ALTER TABLE "I056450"."CARD" DROP ("ISSUED_TEMP");


/*
Table: CLIENT
Add "GENDER" column
*/
ALTER TABLE "I056450"."CLIENT" ADD ("GENDER" NVARCHAR(6));
UPDATE "I056450"."CLIENT"  SET "GENDER" = 'MALE'
      WHERE MOD("BIRTH_NUMBER", 10000) < 2000;
UPDATE "I056450"."CLIENT"  SET "GENDER" = 'FEMALE'
      WHERE MOD("BIRTH_NUMBER", 10000) > 2000;


/*
Table: CLIENT
Add "DATE_BIRTH" column
Drop "BIRTH_DATE" column
*/
ALTER TABLE "I056450"."CLIENT" ADD ("DATE_BIRTH" DATE);
UPDATE "I056450"."CLIENT"  SET "DATE_BIRTH" =
       TO_DATE(TO_NVARCHAR("BIRTH_NUMBER" - mod("BIRTH_NUMBER" , 10000)
       + mod("BIRTH_NUMBER" , 5000)+19000000), 'YYYYMMDD');
ALTER TABLE "I056450"."CLIENT" DROP ("BIRTH_NUMBER");


/*
Table: LOAN
Turn "DATE" column into DATE format
*/
ALTER TABLE "I056450"."LOAN" ADD ("DATE_TEMP" NVARCHAR(6));
UPDATE "I056450"."LOAN"  SET "DATE_TEMP" = "DATE";
ALTER TABLE "I056450"."LOAN" DROP ("DATE");
ALTER TABLE "I056450"."LOAN" ADD ("DATE" DATE);
UPDATE "I056450"."LOAN"  SET "DATE" =
      TO_DATE(LEFT(TO_NVARCHAR("DATE_TEMP"), 6), 'YYMMDD');
ALTER TABLE "I056450"."LOAN" DROP ("DATE_TEMP");


/*
Table: TRANSACTION
Turn "DATE" column into DATE format
*/
ALTER TABLE "I056450"."TRANSACTION" ADD ("DATE_TEMP" NVARCHAR(6));
UPDATE "I056450"."TRANSACTION"  SET "DATE_TEMP" = TO_NVARCHAR("DATE");
ALTER TABLE "I056450"."TRANSACTION" DROP ("DATE");
ALTER TABLE "I056450"."TRANSACTION" ADD ("DATE" DATE);
UPDATE "I056450"."TRANSACTION"  SET "DATE" = TO_DATE("DATE_TEMP", 'YYMMDD');
ALTER TABLE "I056450"."TRANSACTION" DROP ("DATE_TEMP");
```

Data Translation

Finally let’s translate all Czech data content into English. Execute the following SQL statements, having replaced “I056450” with your own schema name.

```
/*
Table: ACCOUNT
Translate "FREQUENCY" content into english
*/
UPDATE "I056450"."ACCOUNT" SET "FREQUENCY" =
      (
    CASE
      WHEN ("FREQUENCY" = 'POPLATEK MESICNE') THEN 'MONTHLY'
        WHEN ("FREQUENCY" = 'POPLATEK TYDNE') THEN 'WEEKLY'
        WHEN ("FREQUENCY" = 'POPLATEK PO OBRATU') THEN 'AFTER TRANSACTION'
        ELSE "FREQUENCY"
      END
    );


/*
Table: PERMANENT_ORDER
Translate "K_SYMBOL" content into english
*/
ALTER TABLE "I056450"."PERMANENT_ORDER" ALTER ("K_SYMBOL" NVARCHAR(20));
UPDATE "I056450"."PERMANENT_ORDER" SET "K_SYMBOL" =
      (
    CASE
      WHEN ("K_SYMBOL" = 'POJISTNE') THEN 'INSURANCE PAYMENT'
        WHEN ("K_SYMBOL" = 'SIPO') THEN 'HOUSEHOLD PAYMENT'
        WHEN ("K_SYMBOL" = 'LEASING') THEN 'LEASING'
        WHEN ("K_SYMBOL" = 'UVER') THEN 'LOAN PAYMENT'
        ELSE "K_SYMBOL"
    END
    );


/*
Table: TRANSACTION
Translate "K_SYMBOL" content into english
*/
ALTER TABLE "I056450"."TRANSACTION" ALTER ("K_SYMBOL" NVARCHAR(40));
UPDATE "I056450"."TRANSACTION" SET "K_SYMBOL" =
      (
    CASE
      WHEN ("K_SYMBOL" = 'POJISTNE') THEN 'INSURANCE PAYMENT'
        WHEN ("K_SYMBOL" = 'SLUZBY') THEN 'PAYMENT FOR STATEMENT'
        WHEN ("K_SYMBOL" = 'UROK') THEN 'INTEREST CREDITED'
        WHEN ("K_SYMBOL" = 'SANKC. UROK') THEN 'SANCTION INTEREST IF NEGATIVE
BALANCE'
        WHEN ("K_SYMBOL" = 'SIPO') THEN 'HOUSEHOLD'
        WHEN ("K_SYMBOL" = 'DUCHOD') THEN 'OLD-AGE PENSION'
        WHEN ("K_SYMBOL" = 'UVER') THEN 'LOAN PAYMENT'
        ELSE "K_SYMBOL"
    END
    );
```

```
/*
Table: TRANSACTION
Translate "OPERATION" content into english
*/
ALTER TABLE "I056450"."TRANSACTION" ALTER ("OPERATION" NVARCHAR(30));
UPDATE "I056450"."TRANSACTION" SET "OPERATION" =
      (
    CASE
      WHEN ("OPERATION" = 'VYBER KARTOU') THEN 'CREDIT CARD WITHDRAWAL'
        WHEN ("OPERATION" = 'VKLAD') THEN 'CREDIT IN CASH'
        WHEN ("OPERATION" = 'PREVOD Z UCTU') THEN 'COLLECTION FROM ANOTHER BANK'
        WHEN ("OPERATION" = 'VYBER') THEN 'WITHDRAWAL IN CASH'
        WHEN ("OPERATION" = 'PREVOD NA UCET') THEN 'REMITTANCE TO ANOTHER BANK'
        ELSE "OPERATION"
    END
    );


/*
Table: TRANSACTION
Translate "TYPE" content into english
*/
ALTER TABLE "I056450"."TRANSACTION" ALTER ("TYPE" NVARCHAR(10));
UPDATE "I056450"."TRANSACTION" SET "TYPE" =
      (
    CASE
      WHEN ("TYPE" = 'PRIJEM') THEN 'CREDIT'
        WHEN ("TYPE" = 'VYDAJ') THEN 'WITHDRAWAL'
        WHEN ("TYPE" = 'VYBER') THEN 'WITHDRAWAL'
        ELSE "TYPE"
    END
    );
```

Update Dates

The dates used in the original data go up to the end of 1998. Add 16 years to these dates so that we work with more recent timeframes.

```
UPDATE "I056450"."ACCOUNT" SET "DATE" = ADD_YEARS("DATE", 16);
UPDATE "I056450"."CARD" SET "ISSUED" = ADD_YEARS("ISSUED", 16);
UPDATE "I056450"."CLIENT" SET "DATE_BIRTH" = ADD_YEARS("DATE_BIRTH", 16);
UPDATE "I056450"."LOAN" SET "DATE" = ADD_YEARS("DATE", 16);
UPDATE "I056450"."TRANSACTION" SET "DATE" = ADD_YEARS("DATE", 16);
```