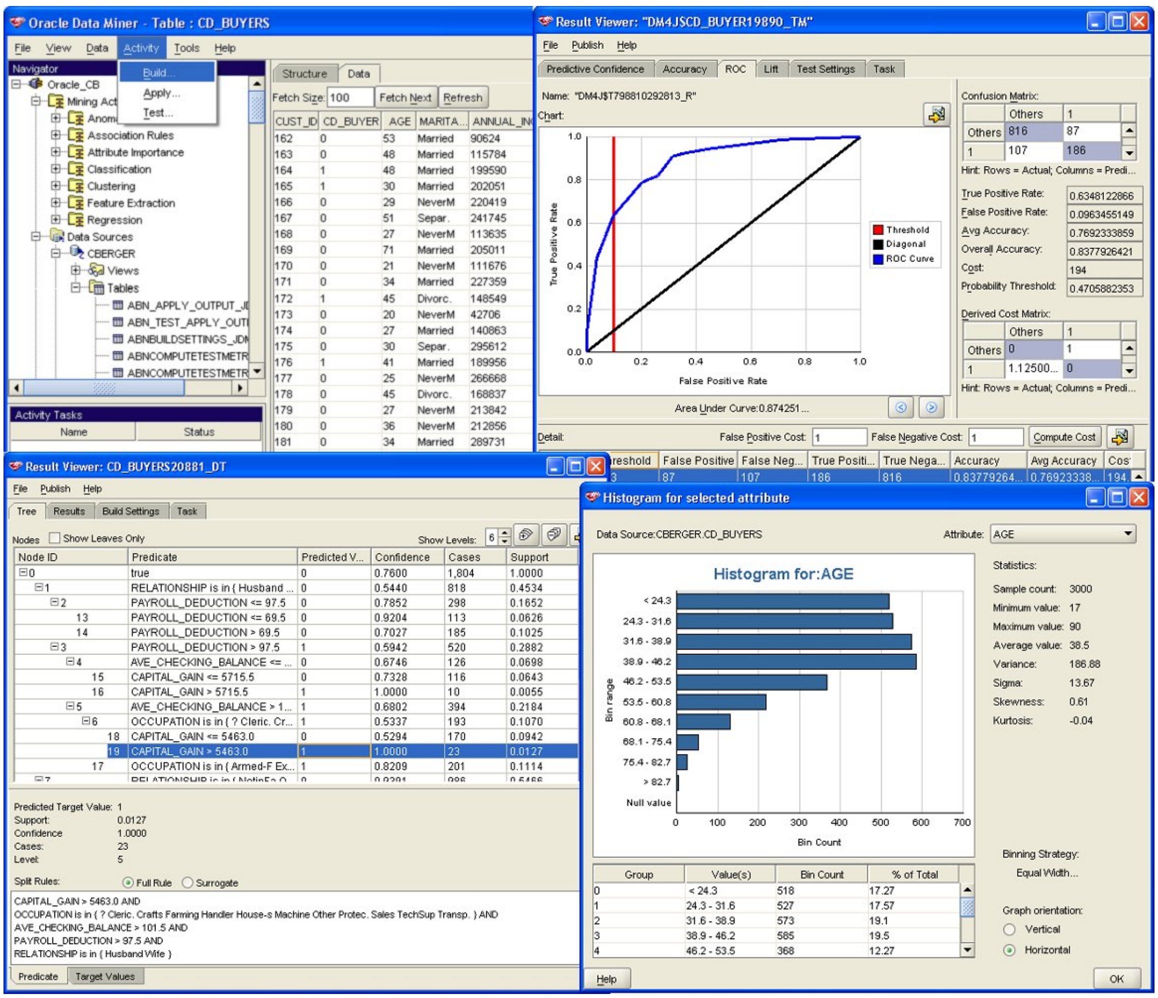
Oracle® Data Mining Tutorial

for

Oracle Data Mining 10*g* Release 2

Oracle Data Mining 11*g* Release 1



January 2008

Copyright © 2008, Oracle. All rights reserved.

Primary Author: Robert Haberstroh

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Table of Contents

Chapter 1 – A Primer on Oracle Data Mining

About this Tutorial

This tutorial was created using Oracle Data Miner 10.0.2.3; it can also be used with more recent releases of Oracle Data Miner.

Oracle Data Miner 10.2.0.4 and Oracle Data Miner 11.1 use the same graphical user interface as Oracle Data Miner 10.2.0.3, with minor changes to some screens.

Different versions of Oracle Data Miner require different versions of Oracle Data Mining:

- Oracle Data Miner 10.2.0.3 and 10.2.0.4 require Oracle Data Mining 10.2. You cannot connect to an Oracle 11*g* database with these versions of Data Miner.

- Oracle Data Miner 11.1 requires Oracle Data Mining 11.1. This is the only version of Oracle Data Miner that works with Oracle 11*g*. You cannot connect to Oracle 10.2 with this version of Data Miner.

Oracle Data Miner 10.2.0.4 provides bug fixes for Oracle Data Miner 10.2.0.3.

Oracle Data Miner 11.1 is the graphical user interface for Oracle Data Mining 11*g*, Release 1 (11.1). For more information about Oracle Data Miner 11.1, see Appendix D.

This tutorial does not explain all features of Oracle Data Miner 11.1; in particular, it does not explain Generalized Linear Models.

Data Mining Solutions

Oracle Data Mining (ODM) can provide solutions to a wide variety of business problems, all centered around gaining insight into the future activities of individuals:

Problem: A retailer wants to increase revenues by identifying all potentially high-value customers in order to offer incentives to them. The retailer also wants guidance in store layout by determining the products most likely to be purchased together.

Solution: An ODM Classification model is built in order to find the customers who are more than 75% likely to spend more than $1000 in the next year.

An ODM Association Rules model is built to analyze market baskets by store location so that product placement can be established on a store-by-store basis.

Problem: A government agency wants faster and more accurate methods of highlighting possible fraudulent activity for further investigation.

Solution: Create ODM Classification, Clustering, and Anomaly Detection models to flag "suspicious" cases.

Problem: A biochemical researcher must deal with thousands of attributes associated with an investigation of drug effectiveness.

Solution: Use ODM's Attribute Importance function to reduce the number of factors to a manageable subset of the attributes.

Problem: A mortgage company wants to increase revenue by reducing the time required for loan approval.

Solution: An ODM Regression model can predict the likely value of a home, eliminating the requirement for an on-site inspection.

Mining with Oracle Data Mining

If you are facing a business problem similar to one of these, then Oracle Data Mining can assist you in developing a solution.

As you approach a data mining problem using ODM, you can be assured that your business domain knowledge and your knowledge of the available data are the most important factors in the process. Oracle Data Mining automates the mechanics of building, testing, and applying a model so that you can concentrate on the business aspects of the problem, not on the mathematical and statistical details – although this tutorial will give you some insight into the underlying operations.

Please refer to the document Oracle Data Mining Concepts, found at
http://www.oracle.com/pls/db102/portal.portal_db?selected=6
for a thorough overview of Oracle Data Mining 10.2; for information about ODM 11.1, see Appendix D of this manual.

The features of Oracle Data Mining are accessible through three different interfaces, each aimed a different type of user:

1) Oracle Data Mining Predictive Analytics (PA) is a package containing two programs – Predict and Explain – each requiring only that the input data

be in the correct format, and making no demands on the user regarding algorithm choices or parameter settings. This package is intended for the non-technical user, such as a marketing director, whose interest is in obtaining a quick and reliable ad hoc result.

Refer to Appendix C for more information on PA.

2) ODM includes both a Java and a PL/SQL Application Programming Interface (API), allowing a programmer to embed ODM functionality into an application such as a Call Center.

Refer to the document ODM Application Developer's Guide, found at http://www.oracle.com/pls/db102/portal.portal_db?selected=6 for more information on the APIs; for information about ODM 11.1 APIs, see the references in Appendix D of this manual.

3) ODM supports a graphical user interface, Oracle Data Miner (ODMr), for use by the business analyst who has a thorough understanding of the business as well as the data available for data mining solutions.

This Tutorial concentrates on the third type of user – the business analyst who will use ODMr to attack and solve business problems.

ODM Functionality

As shown in the introductory data mining examples, ODM is applicable in a variety of business, public sector, health care, and other environments. The common thread running through all data mining projects is the goal of analyzing individual behavior.

The term “behavior” has a loose interpretation, to include:

- The purchasing habits of a customer
- The vulnerability of an individual to a certain disease
- The likelihood that an item passing through an assembly line will be flawed
- The characteristics observed in an individual indicating membership in a particular segment of the population

Data Mining is sometimes called Knowledge Discovery – its goal is to provide actionable information, not found by other means, that can improve your business, whether that business is selling a product, determining what tax returns might be fraudulent, or improving the probability that an oil well will produce a profit.

It is worth noting that the goal is “improvement”, not infallible predictions. For example, suppose a marketing campaign results in a 2% positive response. If Oracle Data Mining can help focus the campaign on the people most likely to respond, resulting in a 3% response, then the business outcome is a 50% increase in revenue.

ODM creates a model of individual behavior, sometimes called a profile, by sifting through cases in which the desired behavior has been observed in the past, and determining a mathematical formula that defines the relationship between the observed characteristics and the behavior. This operation is called “building”, or “training”, a model, and the model is said to “learn” from the training data.

The characteristics indicating the behavior are encapsulated in the model with sufficient generality so that when a new case is presented to the model – even if the case is not exactly like any case seen in the training process - a prediction can be made with a certain confidence, or probability. For example, a person can be predicted to respond positively to a marketing campaign with 73% confidence. That is, the person “fits the profile” of a responder with probability 73%.

You do this all the time with your brain’s ability to make inferences from generalities: if you know that robins, eagles, and chickens are birds, then upon seeing a penguin for the first time you might observe the webbed feet, feathers, beak and something that may be a wing, and you might infer that this individual is likely to be in the “bird” class.

Data mining can be divided into two types of “Learning”, supervised and unsupervised.

<u>Supervised Learning</u> has the goal of predicting a value for a particular characteristic, or attribute that describes some behavior. For example:

S1 Purchasing Product X (Yes or No)
S2 Defaulting on a loan (Yes or No)
S3 Failing in the manufacturing process (Yes or No)
S4 Producing revenue (Low, Medium, High)
S5 Selling at a particular price (a specific amount of money)
S6 Differing from known cases (Yes or No)

The attribute being predicted is called the Target Attribute.

<u>Unsupervised Learning</u> has the goal of discovering relationships and patterns rather than of determining a particular value. That is, there is no target attribute. For Example:

U1 Determine distinct segments of a population and the attribute values indicating an individual's membership in a particular segment.
U2 Determine the five items most likely to be purchased at the same time as item X. (this type of problem is usually called Market Basket Analysis)

Oracle Data Mining provides functionality to solve each of the types of problems shown above.

Examples S1, S2, S3 illustrate Binary Classification – the model predicts one of two target values for each case (that is, places each case into one of two classes, thus the term Classification).

Example S4 illustrates Multiclass Classification – the model predicts one of several target values for each case.

Example S5 illustrates Regression – the model predicts a specific target value for each case from among (possibly) infinitely many values.

Example S6 illustrates One-class Classification, also known as Anomaly Detection – the model trains on data that is homogeneous, that is all cases are in one class, then determines if a new case is similar to the cases observed, or is somehow "abnormal" or "suspicious".

Example U1 illustrates Clustering – the model defines segments, or "clusters" of a population, then decides the likely cluster membership of each new case.

Example U2 illustrates Associations – the model determines which cases are likely to be found together.

Each ODM function will be discussed and explained in detail as the tutorial proceeds.

The Data Mining Process

The phases of solving a business problem using Oracle Data Mining are as follows:

- Problem Definition in Terms of Data Mining and Business Goals
- Data Acquisition and Preparation
- Building and Evaluation of Models
- Deployment

Problem Definition in Terms of Data Mining and Business Goals

The business problem must be well-defined and stated in terms of data mining functionality. For example, retail businesses, telephone companies, financial institutions, and other types of enterprises are interested in customer "churn" – that is, the act of a previously loyal customer in switching to a rival vendor.

The statement "I want to use data mining to solve my churn problem" is much too vague. From a business point of view, the reality is that it is much more difficult and costly to try to win a defected customer back than to prevent a disaffected customer from leaving; furthermore, you may not be interested in retaining a low-value customer. Thus, from a data mining point of view, the problem is to predict which customers are likely to churn with high probability, and also to predict which of those are potentially high-value customers.

This requires clear definitions of "low-value" customer and of "churn". Both are business decisions, and may be difficult in some cases – a bank knows when a customer has closed a checking account, but how does a retailer know when a customer has switched loyalties? Perhaps this can be determined when purchases recorded by an affinity card decrease dramatically over time.

Suppose that these business definitions have been determined. Then we can state the problem as: "I need to construct a list of customers who are predicted to be most likely to churn and also are predicted to be likely high-value customers, and to offer an incentive to these customers to prevent churn". The definition of "most likely" will be left open until we see the results generated by Oracle Data Mining.

Data acquisition and Preparation

A general rule of thumb in data mining is to gather as much information as possible about each individual, then let the data mining operations indicate any filtering of the data that might be beneficial. In particular, you should not eliminate some attribute because you think that it might not be important – let ODM's algorithms make that decision. Moreover, since the goal is to build a profile of behavior that can be applied to any individual, you should eliminate specific identifiers such as name, street address, telephone number, etc. (however, attributes that indicate a general location without identifying a specific individual, such as Postal Code, may be helpful.)

Continuing with the churn example in the context of a bank, you may have a customer's personal demographics stored in one location (age, income, etc.), "business" demographics in another (a list of the customer's banking products, beginning/ending dates, etc), and transactions in another. You will need access to each of these locations.

After determining a business definition for “churn”, you will probably have to add a new column to each customer’s record indicating Churn (Yes/No). Also, you will want to create new columns giving aggregate and derived information (Years_as_Customer rather than Beginning_Date, Avg_num_transactions_per_month, etc.).

It is generally agreed that the data gathering and preparation phase consumes more than 50% of the time and effort of a data mining project.

Building and Evaluation of Models

The Activity Guides of Oracle Data Miner automate many of the difficult tasks during the building and testing of models. It’s difficult to know in advance which algorithms will best solve the business problem, so normally several models are created and tested.

No model is perfect, and the search for the best predictive model is not necessarily a question of determining the model with the highest accuracy, but rather a question of determining the types of errors that are tolerable in view of the business goals.

For example, a bank using a data mining model to predict credit risk in the loan application process wants to minimize the error of predicting “no risk” when in fact the applicant is likely to default, since that type of error is very costly to the bank. On the other hand, the bank will tolerate a certain number of errors that predict “high risk” when the opposite is true, as that is not very costly to the bank (although the bank loses some potential profit and the applicant may become a disgruntled customer at being denied the loan).

As the tutorial proceeds through the Mining Activities, there will be more discussion on determining the “best” model.

Deployment

Oracle Data Mining produces actionable results, but the results are not useful unless they can be placed into the correct hands quickly.

For instantaneous presentation of results, refer to the documents cited above on programmatic deployment using the Oracle Data Mining Java or PL/SQL API, or to the use of Predictive Analytics.

Continuing with the bank's churn problem, when an ODM predictive model is applied to the customer base for the purpose of creating a ranked list of those likely to churn, a table is created in the database and is populated with the Customer ID/Prediction/Probability details. Thus, the results are available using any of the usual methods of querying a database table.

In particular, the Oracle Data Miner user interface provides wizards for publishing the results either to an Excel spreadsheet or to Oracle Discoverer.

Chapter 2 - Data Exploration and Transformation

The data used in the data mining process usually has to be collected from various locations, and also some transformation of the data is usually required to prepare the data for data mining operations. The Mining Activity Guides will assist you in joining data from disparate sources into one view or table, and will also carry out transformations that are required by a particular algorithm; those transforms will be discussed in the context of the Guides. However, there are transforms that typically will be completed on a standalone basis using one of the Data Transformation wizards.

These include

- Recode
- Filter
- Derive field

and others.

Moreover, utilities are available for importing a text file into a table in the database, for displaying summary statistics and histograms, for creating a view, for creating a table from a view, for copying a table, and for dropping a table or view.

The examples below assume that the installation and configuration explained in Appendix A have been completed and that the sample views are available to the current user.

These sample views include:

MINING_DATA_BUILD_V
MINING_DATA_TEST_V
MINING_DATA_APPLY_V

and others, including the tables of the SH schema.

These tables describe the purchasing habits of customers in a pilot marketing campaign. They will be used to illustrate the business problems of identifying the most valuable customers as well as defining the product affinity that will help determine product placement in the stores.

Note on data format: Previous versions of Oracle Data Mining allowed two distinct data formats, Single Row per Record, in which all the information about an individual resides in a single row of the table/view, and Multiple row per Record (sometimes called “Transactional” format), in which information for a given individual may be found in several rows (for example if each row represents an item purchased). In ODM 10*g* Release 2 and ODM 11*g* Release 1, only Single Row per Record format is acceptable (except in the case of Association Rules); however, some language relating to the former distinction remains in some wizards. An example of the Single Row per Record format will be seen in the sample MINING_DATA_BUILD_V.

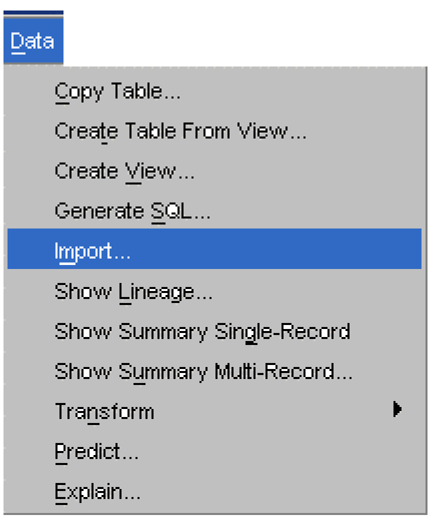
The database feature called Nested Column is used to accommodate the use case previously handled by Transactional format.

To begin, launch the Oracle Data Miner user interface as explained in the final two sections of Appendix A.

The Import Wizard

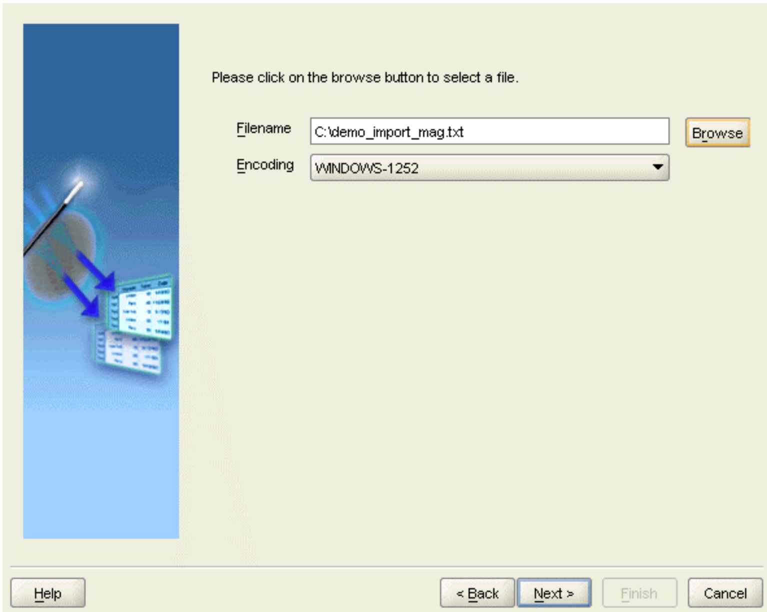
The text file demo_import_mag.txt is included in the Supplemental_Data file available with this tutorial. It consists of comma-separated customer data from a magazine subscription service, with attribute names in the first row. The Import wizard accepts information about the text file from the user and configures the SQLLDR command to create a table. You must identify the location of the SQLLDR executable in the Preferences worksheet. See Appendix B – Setting Preferences.

To import the text file into a table, select Import in the Data pulldown menu.

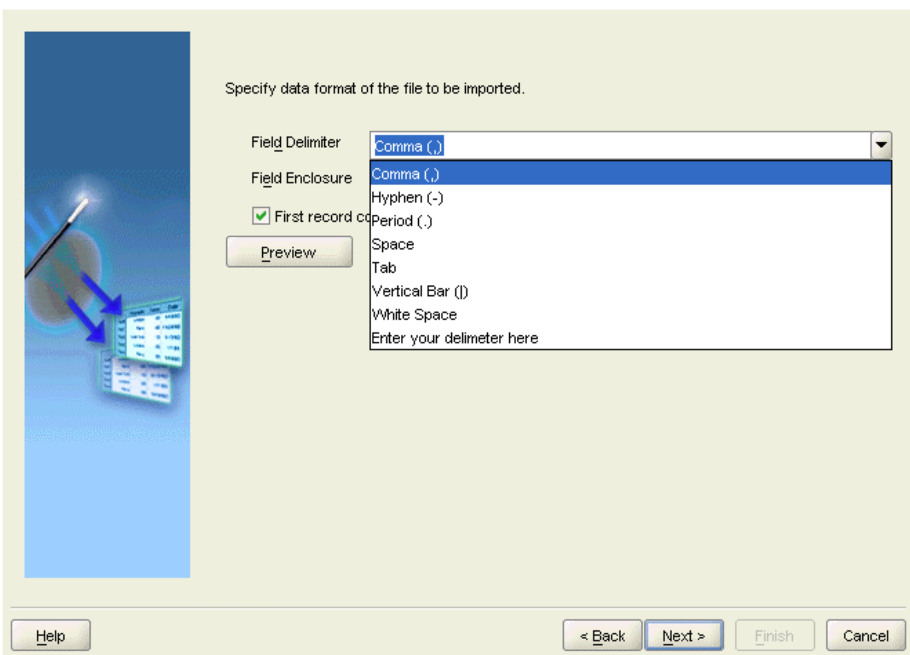


Click Next on the Welcome page to proceed.

Step 1: Click Browse to locate the text file to be imported



Step 2: Select the field (column) delimiter from the pulldown menu

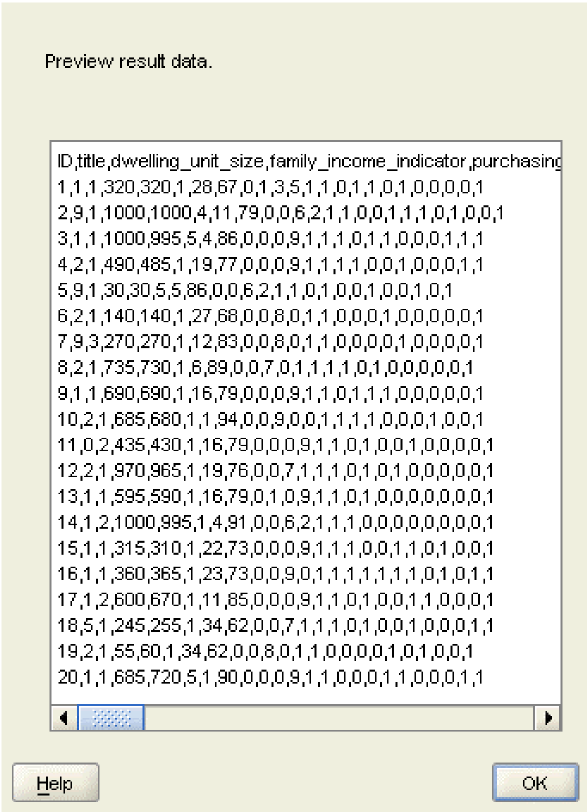


Any string field values containing the delimiter must be enclosed in either single or double quotes; if this is the case, specify the enclosures from the pull-down menu. In addition, certain other characters are unacceptable in a string for some purposes; an alternative to quoting the string is replacing the illegal characters prior to importing the file.

SQLLDR parameters such as termination criteria can be selected by clicking Advanced Settings.

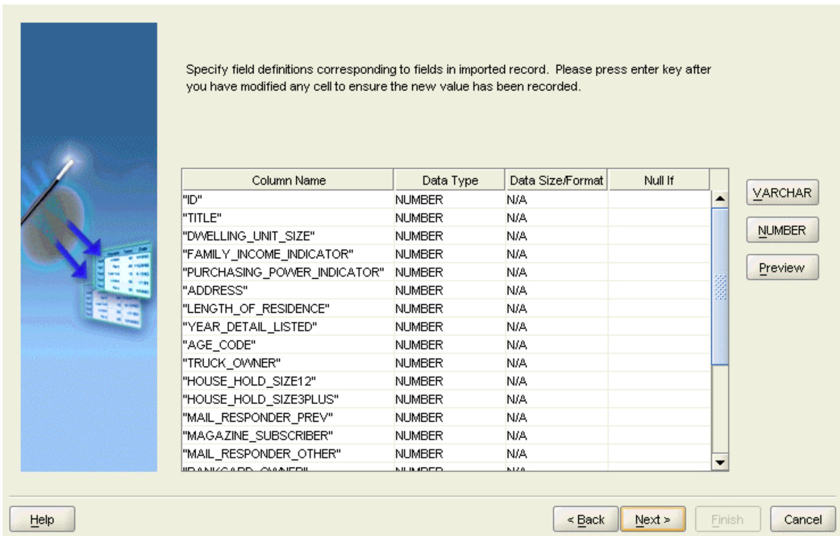
If the first row of the file contains the field names, click the appropriate checkbox.

To verify the format specifications, click Preview:

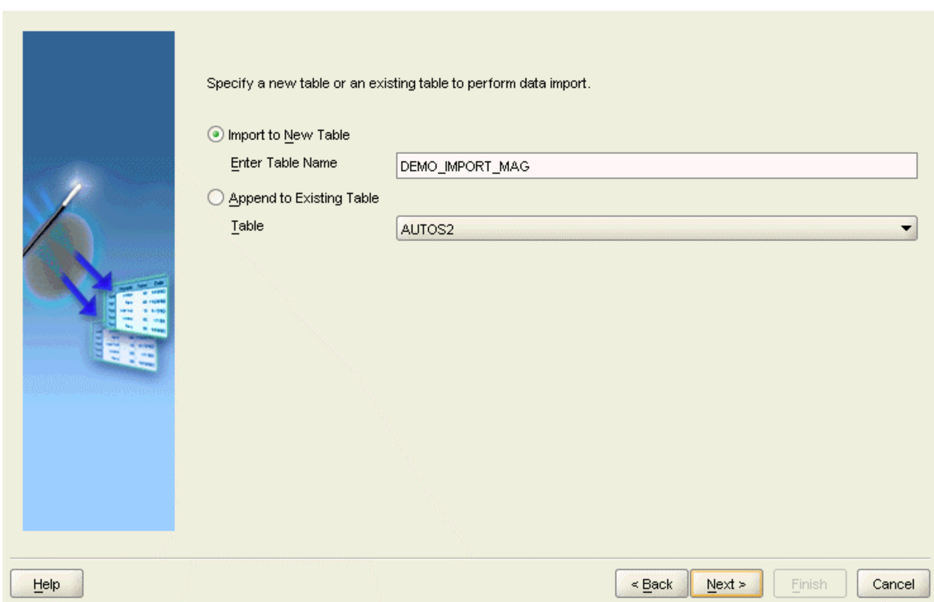


Step 3: Verify the attribute names and data types. If the first row of the text file does not contain field names, then dummy names are supplied and they may be modified in this step (don't forget to enclose the new column names in double quotes). The Data Type may also be modified.

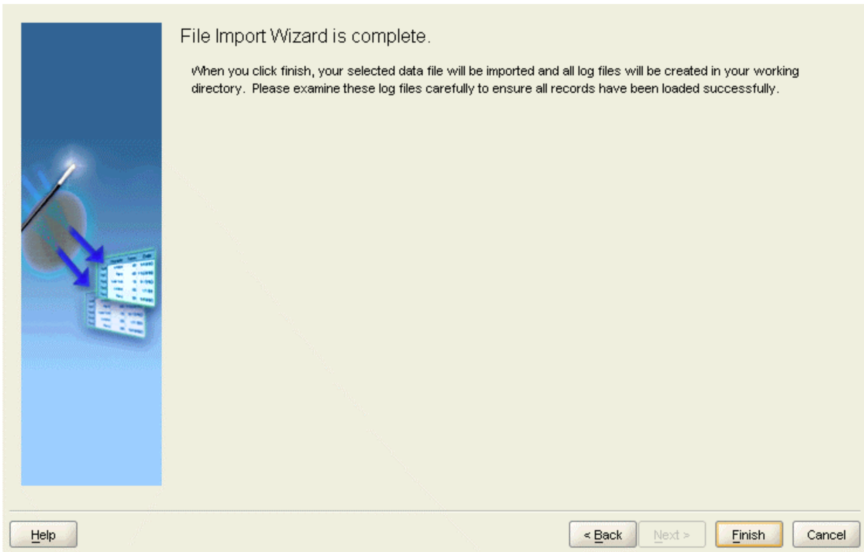
In the NULL IF column, you can specify a string that will be recoded to NULL if encountered, for example ? or UNKNOWN.



Step 4: Specify the name of the new table or the existing table in which the imported data will be inserted:



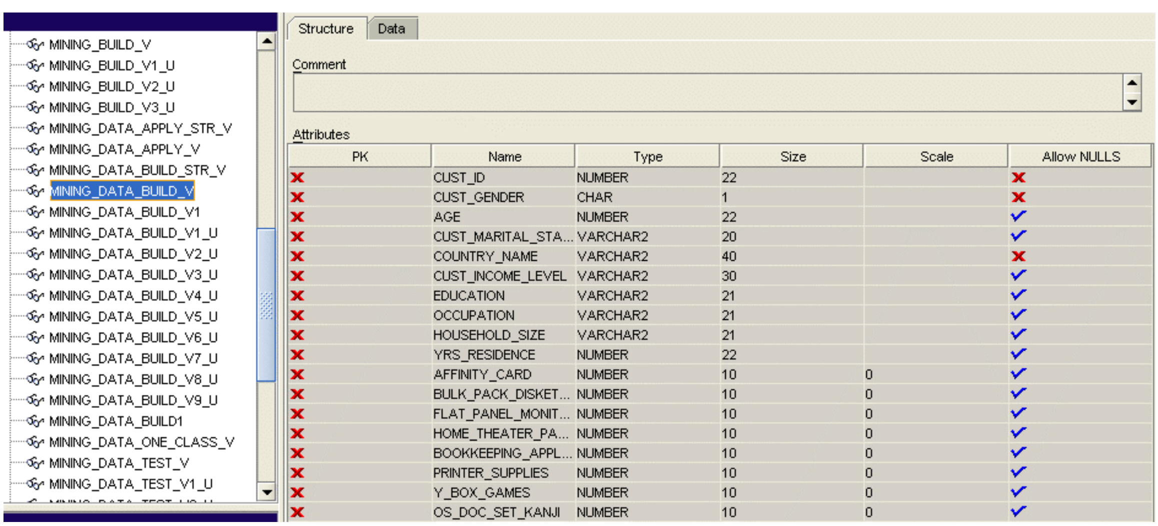
Click Finish to initiate the import operation.



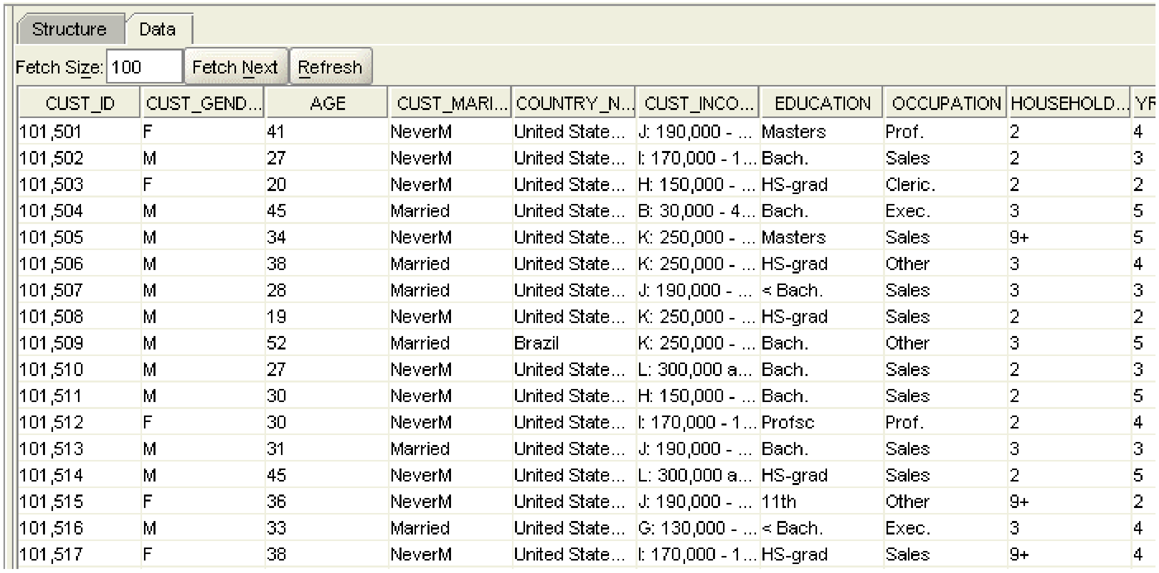
When completed, the Browser displays a sample from the table.

Data Viewer and Statistics

Left click on the name of a table or view to display the structure.



Click the Data tab to see a sample of the table/view contents.


Structure | Data

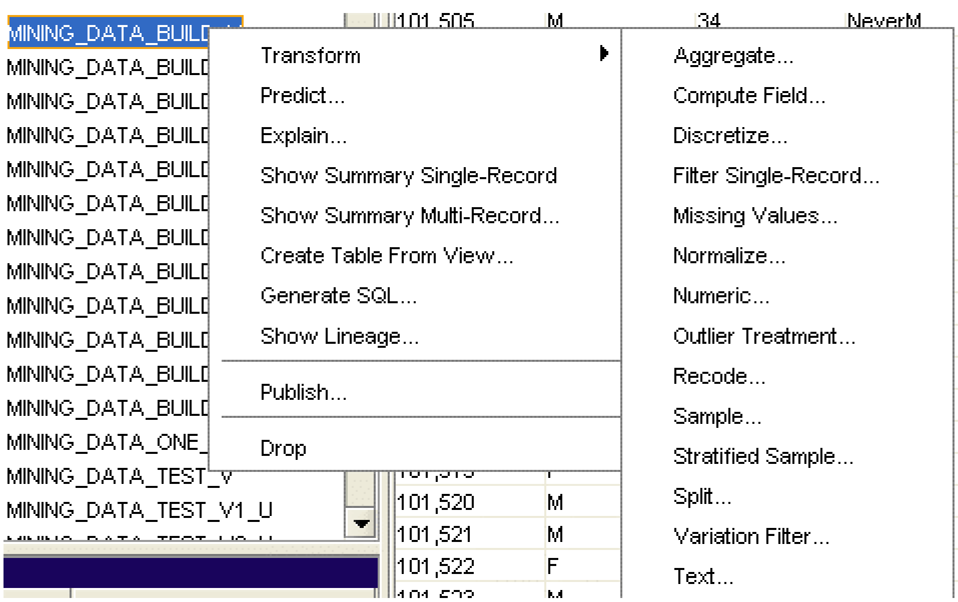
Fetch Size: 100 | Fetch Next | Refresh

CUST_ID	CUST_GEND...	AGE	CUST_MARI...	COUNTRY_N...	CUST_INCO...	EDUCATION	OCCUPATION	HOUSEHOLD...	YF
101,501	F	41	NeverM	United State...	J: 190,000 - ...	Masters	Prof.	2	4
101,502	M	27	NeverM	United State...	I: 170,000 - 1...	Bach.	Sales	2	3
101,503	F	20	NeverM	United State...	H: 150,000 - ...	HS-grad	Cleric.	2	2
101,504	M	45	Married	United State...	B: 30,000 - 4...	Bach.	Exec.	3	5
101,505	M	34	NeverM	United State...	K: 250,000 - ...	Masters	Sales	9+	5
101,506	M	38	Married	United State...	K: 250,000 - ...	HS-grad	Other	3	4
101,507	M	28	Married	United State...	J: 190,000 - ...	< Bach.	Sales	3	3
101,508	M	19	NeverM	United State...	K: 250,000 - ...	HS-grad	Sales	2	2
101,509	M	52	Married	Brazil	K: 250,000 - ...	Bach.	Other	3	5
101,510	M	27	NeverM	United State...	L: 300,000 a...	Bach.	Sales	2	3
101,511	M	30	NeverM	United State...	H: 150,000 - ...	Bach.	Sales	2	5
101,512	F	30	NeverM	United State...	I: 170,000 - 1...	Profsc	Prof.	2	4
101,513	M	31	Married	United State...	J: 190,000 - ...	Bach.	Sales	3	3
101,514	M	45	NeverM	United State...	L: 300,000 a...	HS-grad	Sales	2	5
101,515	F	36	NeverM	United State...	J: 190,000 - ...	11th	Other	9+	2
101,516	M	33	Married	United State...	G: 130,000 - ...	< Bach.	Exec.	3	4
101,517	F	38	NeverM	United State...	I: 170,000 - 1...	HS-grad	Sales	9+	4

The default number of records shown is 100; enter a different number in the Fetch Size window, then click Refresh to change the size of the display, or click Fetch Next to add to add more rows to the display.

Right-click the table/view name to expose a menu with more options.

Click Transform to expose another menu giving access to transformation wizards (some of which will be discussed in detail later).

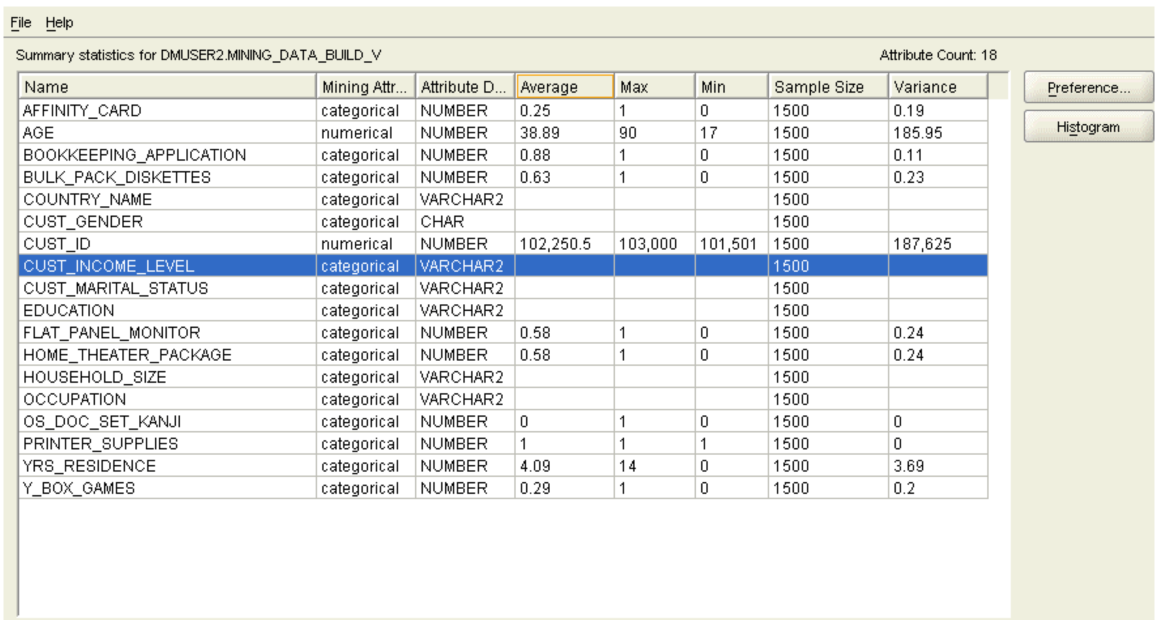


The two menu choices Generate SQL and Show Lineage appear only for views; they are not on the menu for tables.

Show Lineage displays the SQL code and identifies the underlying table(s) used to create the view, while Generate SQL allows you to save the SQL code into an executable script.

Create Table from View and Drop are self-explanatory, Predict and Explain are discussed in Appendix C, and Publish makes the table or view available to Oracle Discoverer. Publish will be discussed in Chapter 14: Deployment.

To see a statistical summary, click one of the two selections depending on the data format type. The following example uses Show Summary Single-Record.

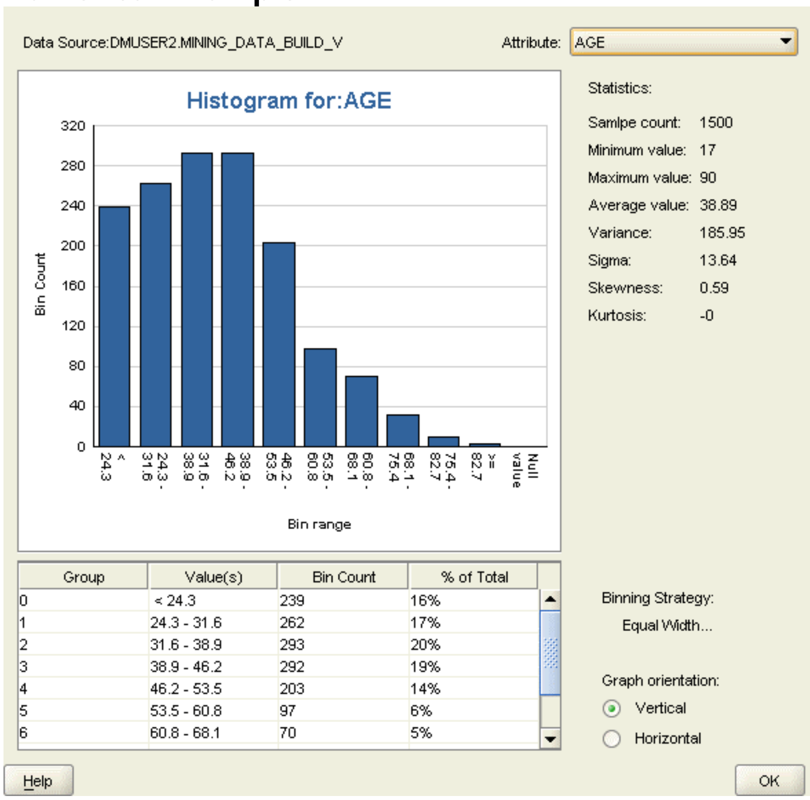


Name	Mining Attr...	Attribute D...	Average	Max	Min	Sample Size	Variance
AFFINITY_CARD	categorical	NUMBER	0.25	1	0	1500	0.19
AGE	numerical	NUMBER	38.89	90	17	1500	185.95
BOOKKEEPING_APPLICATION	categorical	NUMBER	0.88	1	0	1500	0.11
BULK_PACK_DISKETTES	categorical	NUMBER	0.63	1	0	1500	0.23
COUNTRY_NAME	categorical	VARCHAR2				1500	
CUST_GENDER	categorical	CHAR				1500	
CUST_ID	numerical	NUMBER	102,250.5	103,000	101,501	1500	187,625
CUST_INCOME_LEVEL	categorical	VARCHAR2				1500	
CUST_MARITAL_STATUS	categorical	VARCHAR2				1500	
EDUCATION	categorical	VARCHAR2				1500	
FLAT_PANEL_MONITOR	categorical	NUMBER	0.58	1	0	1500	0.24
HOME_THEATER_PACKAGE	categorical	NUMBER	0.58	1	0	1500	0.24
HOUSEHOLD_SIZE	categorical	VARCHAR2				1500	
OCCUPATION	categorical	VARCHAR2				1500	
OS_DOC_SET_KANJI	categorical	NUMBER	0	1	0	1500	0
PRINTER_SUPPLIES	categorical	NUMBER	1	1	1	1500	0
YRS_RESIDENCE	categorical	NUMBER	4.09	14	0	1500	3.69
Y_BOX_GAMES	categorical	NUMBER	0.29	1	0	1500	0.2

For each numerical attribute, Maximum and Minimum values, as well as average and variance, are shown. These statistics are calculated on a sample (1500 in this screen shot); the size of the sample can be changed by adjusting ODM Preferences as explained in Appendix B.

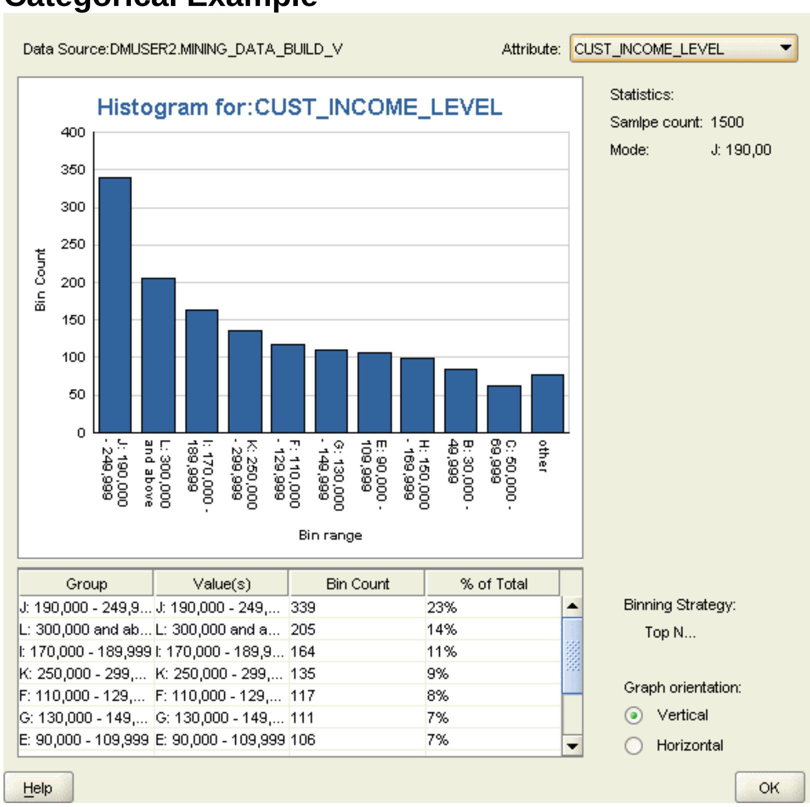
For any highlighted attribute, click Histogram to see a distribution of values. The values are divided into ranges, or bins.

Numerical Example



Group	Value(s)	Bin Count	% of Total
0	< 24.3	239	16%
1	24.3 - 31.6	262	17%
2	31.6 - 38.9	293	20%
3	38.9 - 46.2	292	19%
4	46.2 - 53.5	203	14%
5	53.5 - 60.8	97	6%
6	60.8 - 68.1	70	5%

Categorical Example



Group	Value(s)	Bin Count	% of Total
J: 190,000 - 249,9...	J: 190,000 - 249,...	339	23%
L: 300,000 and ab...	L: 300,000 and a...	205	14%
I: 170,000 - 189,999	I: 170,000 - 189,9...	164	11%
K: 250,000 - 299,...	K: 250,000 - 299,...	135	9%
F: 110,000 - 129,...	F: 110,000 - 129,...	117	8%
G: 130,000 - 149,...	G: 130,000 - 149,...	111	7%
E: 90,000 - 109,999	E: 90,000 - 109,999	106	7%

The default number of bins is 10; this number can be changed for a highlighted attribute by clicking Preference in the Summary window.

Numerical attributes are divided into bins of equal width between the minimum and maximum. The bins are displayed in ascending order of attribute values.

Categorical attributes are binned using the “Top N” method (N is the number of bins). The N values occurring most frequently have bins of their own; the remaining values are thrown into a bin labeled “Other”. The bins are displayed in descending order of bin size.

Transformations

You can right-click on the table/view name or pull down the Data menu to access the data transformation wizards. Many of the transforms are incorporated into the Mining Activity Guides; some have value as standalone operations. In each case the result is a view, unless the wizard allows a choice of table or view. Some examples follow:

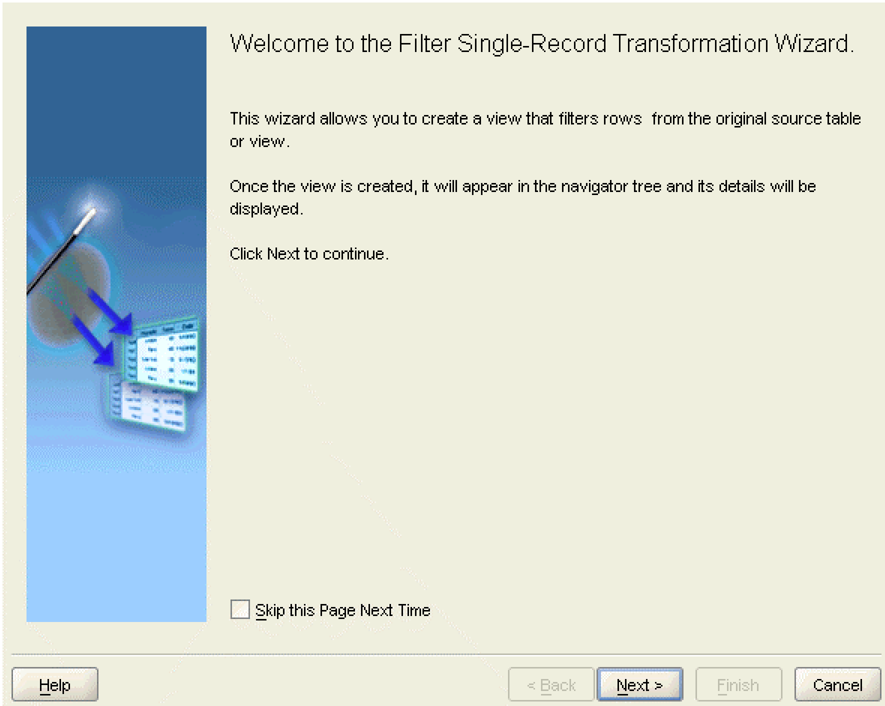
Filter Single-Record

Suppose we want to concentrate on our customers between the ages of 21 and 35. We can filter the data to include only those people.

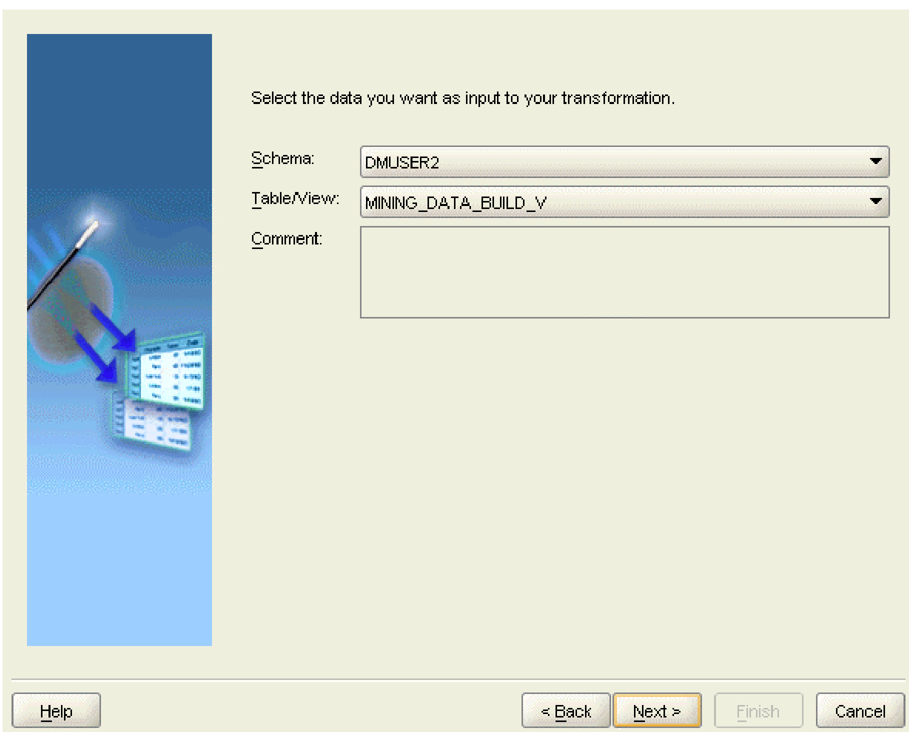
Oracle Data Miner provides a filtering transformation to define a subset of the data based upon attribute values.

Begin by highlighting Transformations on the Data pulldown menu and selecting Filter Single-Record (or right-click on the table/view name) to launch the wizard.

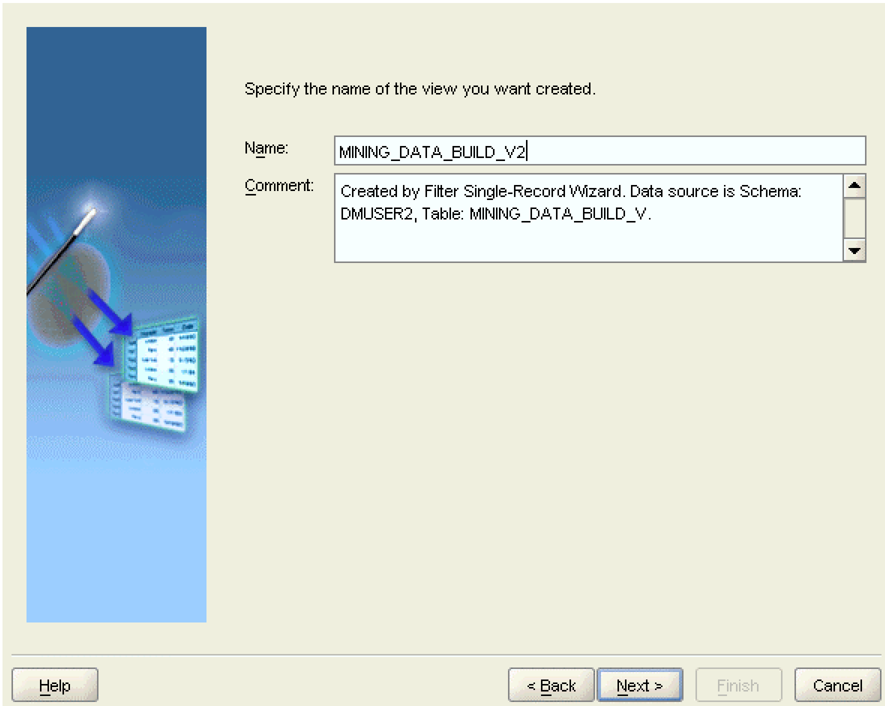
Click Next on the Welcome page.



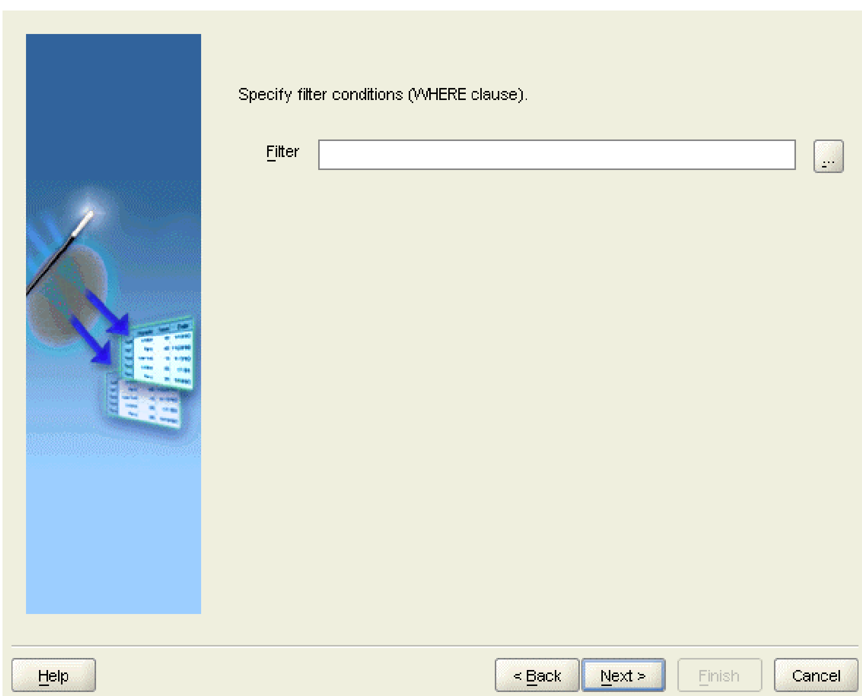
Identify the input data and click Next (if you accessed the wizard by right-clicking the table/view name, then the data is already known and this step is skipped).

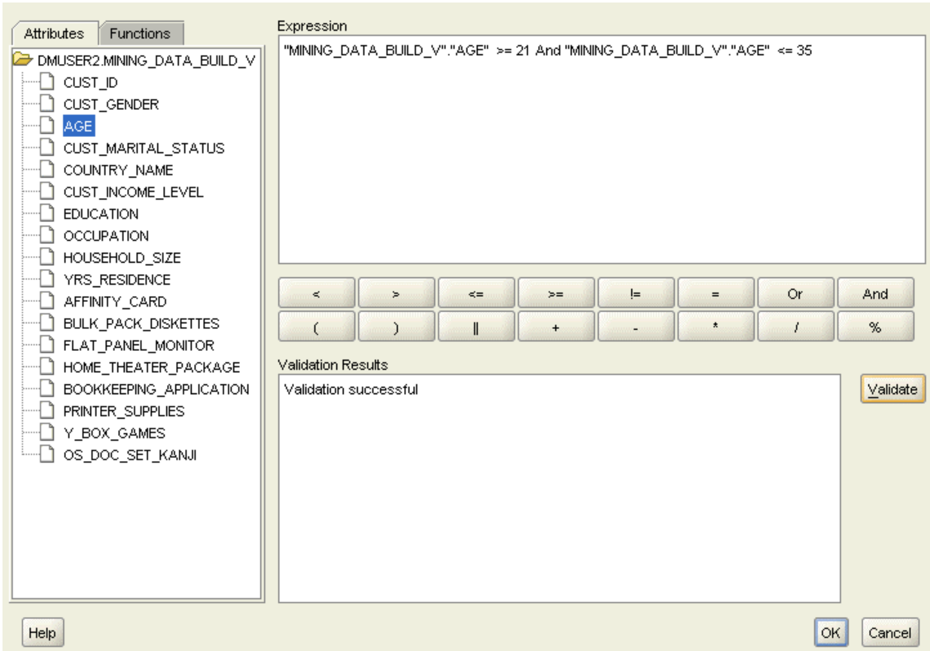


Enter a name for the resultant view and click Next.



Click the icon to the right of the Filter window to construct the filtering condition in a dialog box.



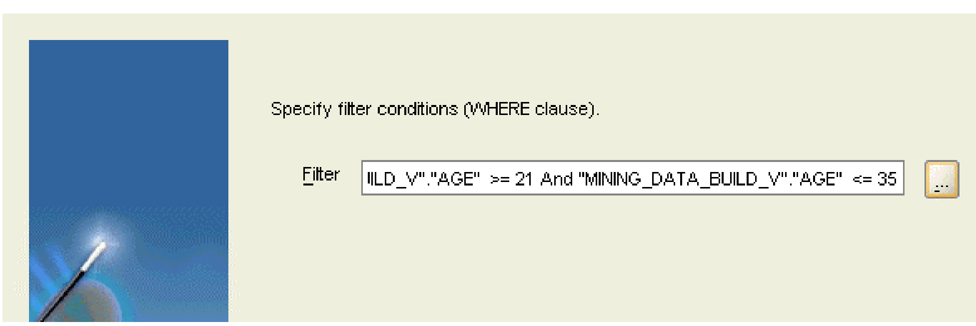


The Expression Editor allows easy construction of the “where clause” that will be inserted into the query to create the new view.

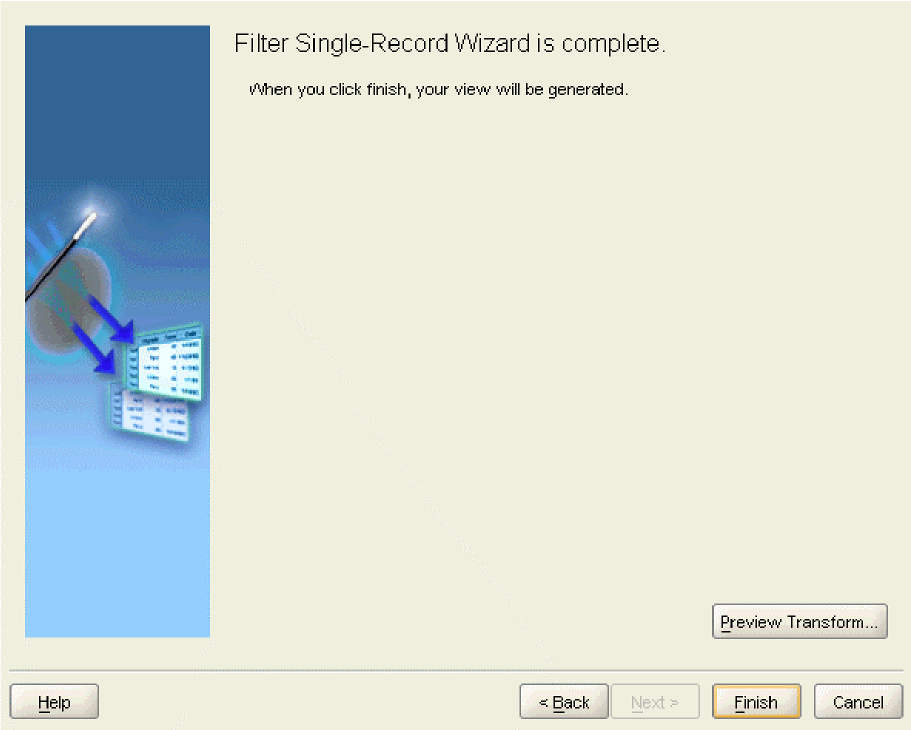
In this example, we want only those records representing individuals whose age is between 21 and 35 years. Double-click the attribute name AGE, click the “>=” button, and type “21” to construct the first part of the condition shown. Click AND to continue defining the full condition. Note that complex conditions can be constructed using the “And”, “Or”, and parentheses buttons.

Click the Validate button to check that the condition is satisfied by a subset of the source data.

When you dismiss the Expression Editor by clicking OK, the condition is displayed in the Filter window.



You may preview the results and then choose to generate a stored procedure by clicking Preview Transform on the Finish page. Click Finish to complete the transformation.



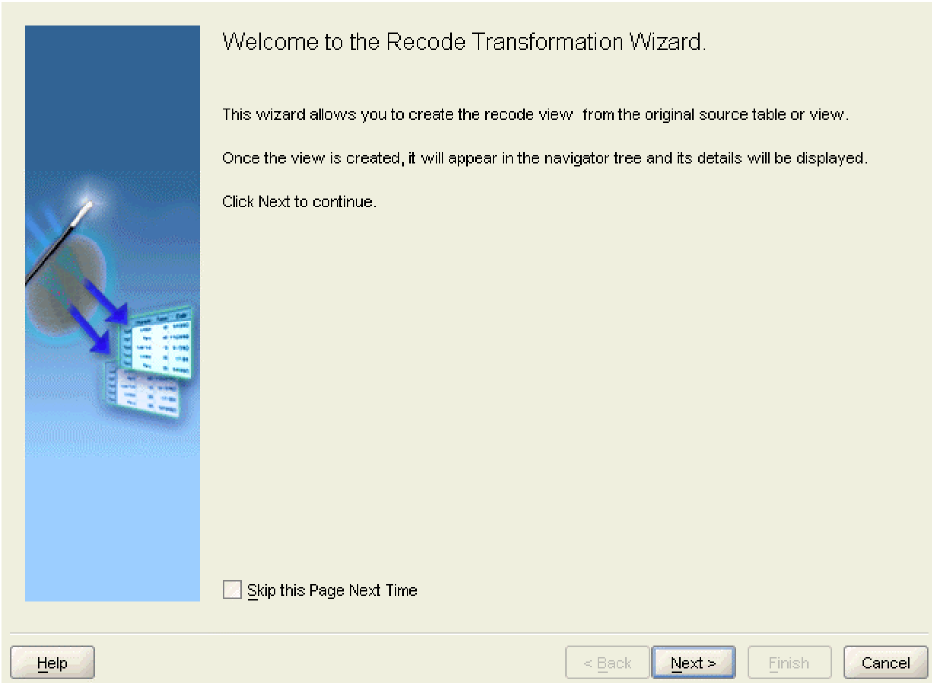
When the transformation is complete, a sample of the new data is displayed.

Recode

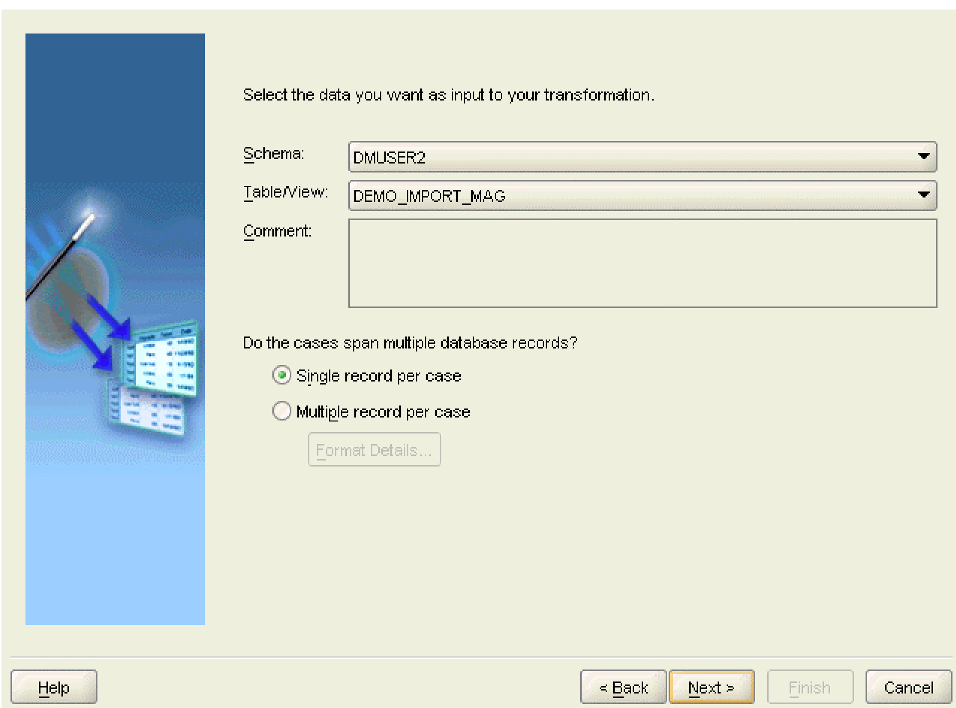
The Recode transformation allows specified attribute values to be replaced by new values. For example, suppose the Summarization Viewer reveals that the attribute LENGTH_OF_RESIDENCE has a numerical range from 1 to 34 in the table DEMO_IMPORT_MAG, just created in the Import example. In order to make the model build operation more efficient, you decide to consider only two classes of residence: LOW for residences of less than or equal to 10 years, and HIGH for residences of more than 10 years.

NOTE: The Recode transformation scans the entire dataset and compiles a list of distinct values for the attribute to be recoded, resulting in possible system resource problems if the attribute is numerical and continuous. In this case, the same outcome can be produced without difficulty by defining bins manually using the Discretization wizard.

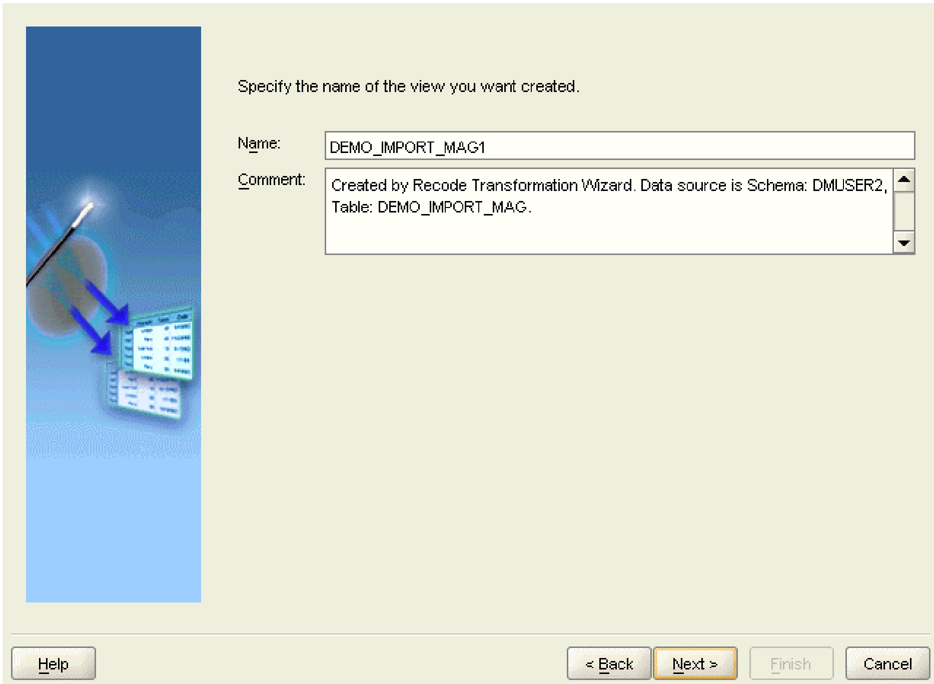
Begin by highlighting Transform on the Data pulldown menu and selecting Recode (or right-click on the table/view name) to launch the wizard.



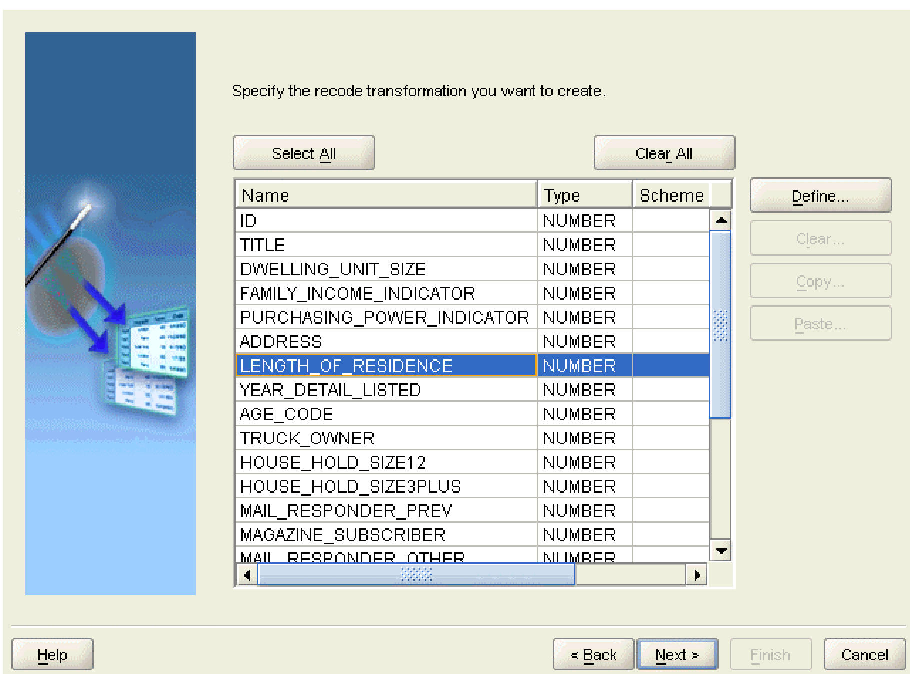
Select the table or view to be transformed and specify the format by clicking the appropriate radio button (if you accessed the wizard by right-clicking the table/view name, then the data is already known and this step is skipped).



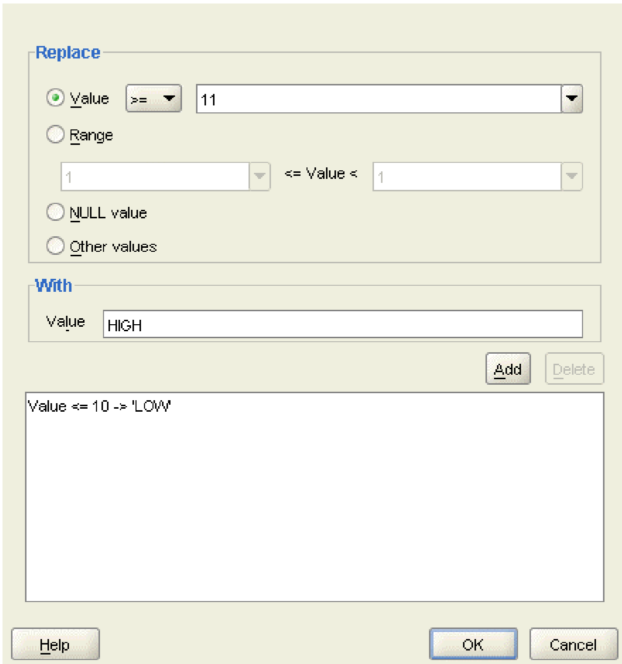
Enter a name for the resultant view.



Highlight the attribute to be recoded and click Define.

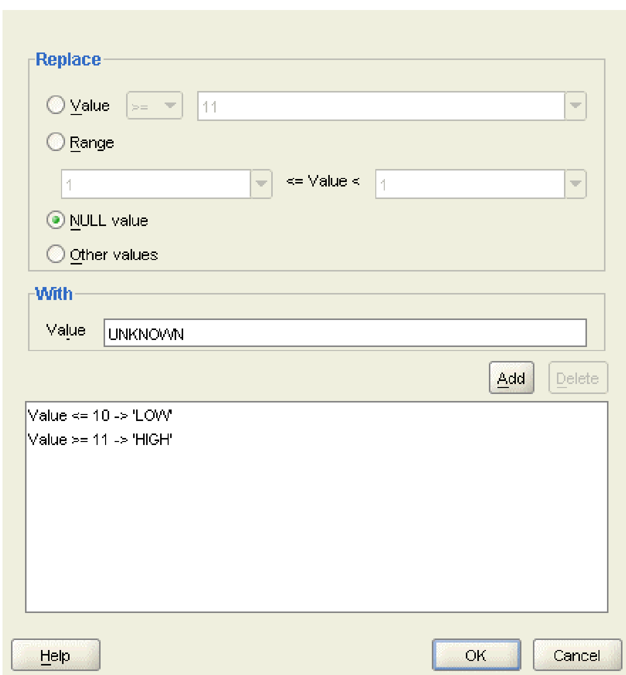


In the Recode dialog box, choose the condition on the attribute value and enter the new value in the With Value window; click Add to confirm. Repeat for each condition.

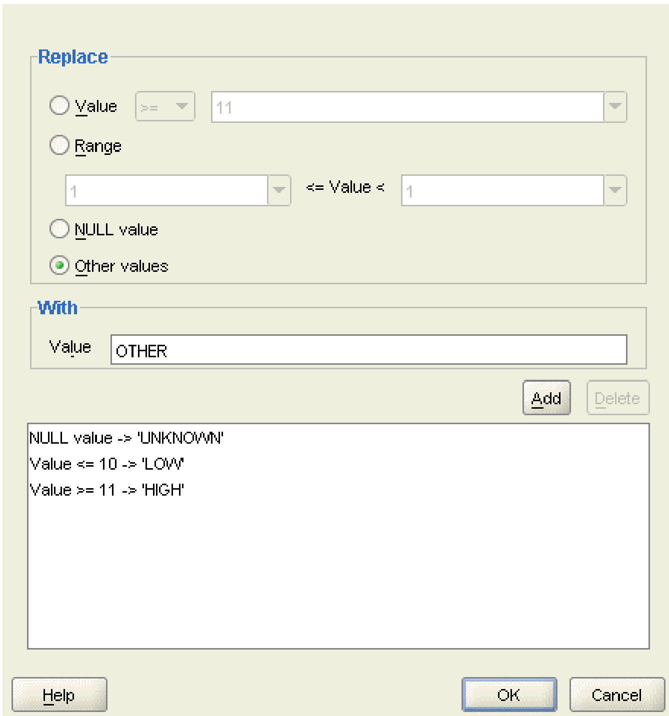


Warning: The wizard does not check the conditions for inconsistencies.

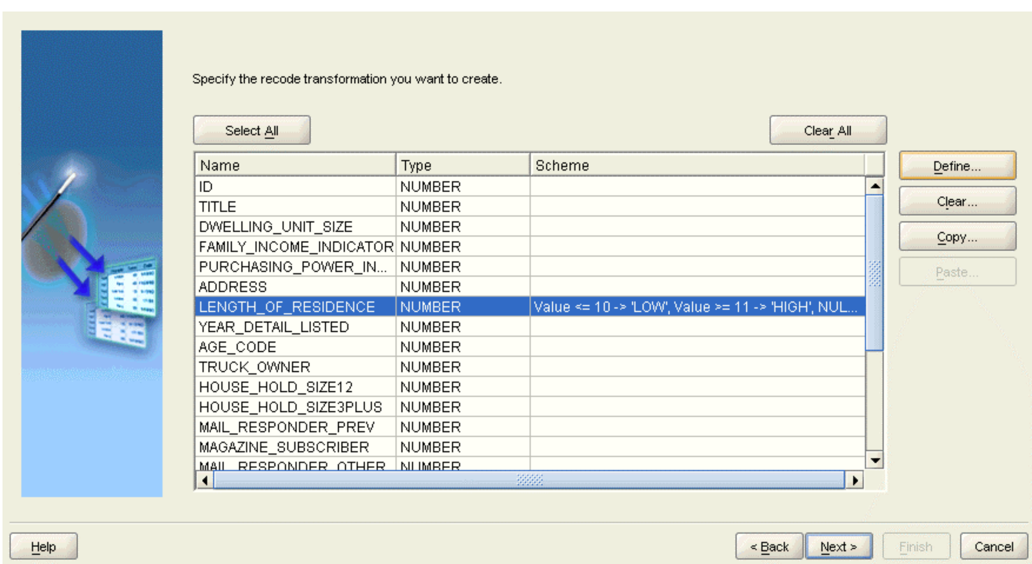
In the same dialog box, a missing values treatment can be defined. In this example, all null values for this attribute are recoded to ‘UNKNOWN’.



Also, a treatment for any value not included in the conditions may be defined; in this example, all such values are recoded to ‘OTHER’.

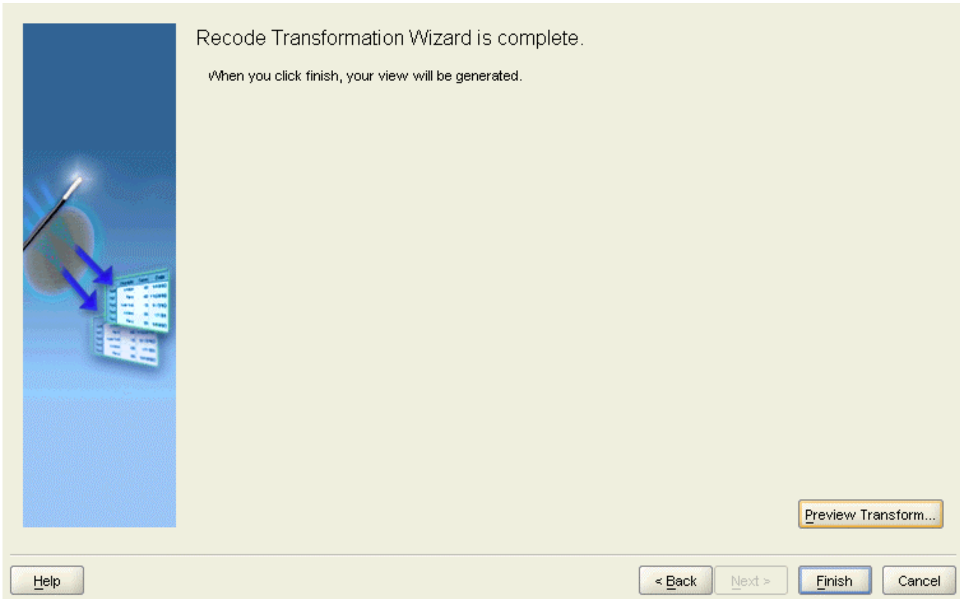


Click OK; the recode definitions are now displayed with the attributes. You may recode more than one attribute by highlighting another attribute and repeating the steps.

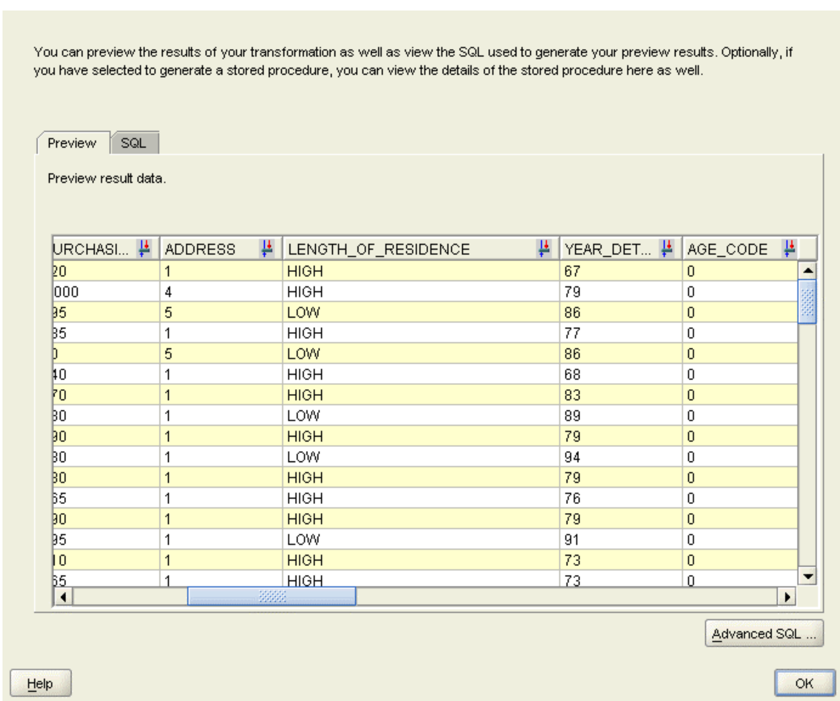


When done, click Next.

You may preview the results by clicking Preview Transform on the Finish page.

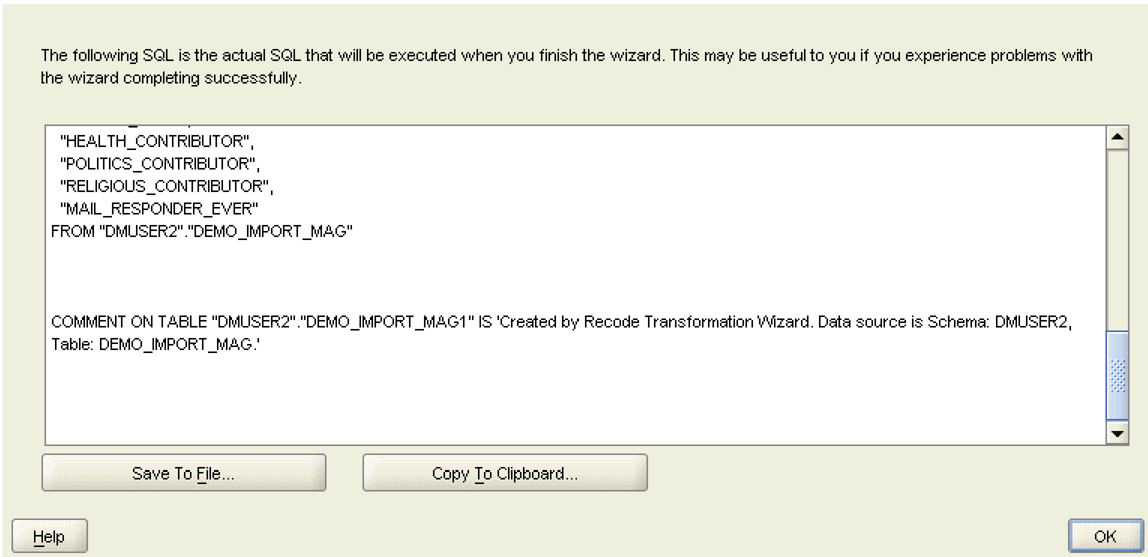


Note that the recoded attribute has assumed the defined data type; LENGTH_OF_RESIDENCE, previously numerical, is now of type VARCHAR2.



URCHASI...	ADDRESS	LENGTH_OF_RESIDENCE	YEAR_DET...	AGE_CODE
20	1	HIGH	67	0
000	4	HIGH	79	0
35	5	LOW	86	0
35	1	HIGH	77	0
0	5	LOW	86	0
40	1	HIGH	68	0
70	1	HIGH	83	0
30	1	LOW	89	0
30	1	HIGH	79	0
30	1	LOW	94	0
30	1	HIGH	79	0
65	1	HIGH	76	0
30	1	HIGH	79	0
35	1	LOW	91	0
10	1	HIGH	73	0
65	1	HIGH	73	0

On this same page, you can click the SQL tab to see the query used to display the preview. To save executable code for future use, you can click the Advanced SQL button to see and save the complete code that creates the transformed dataset.



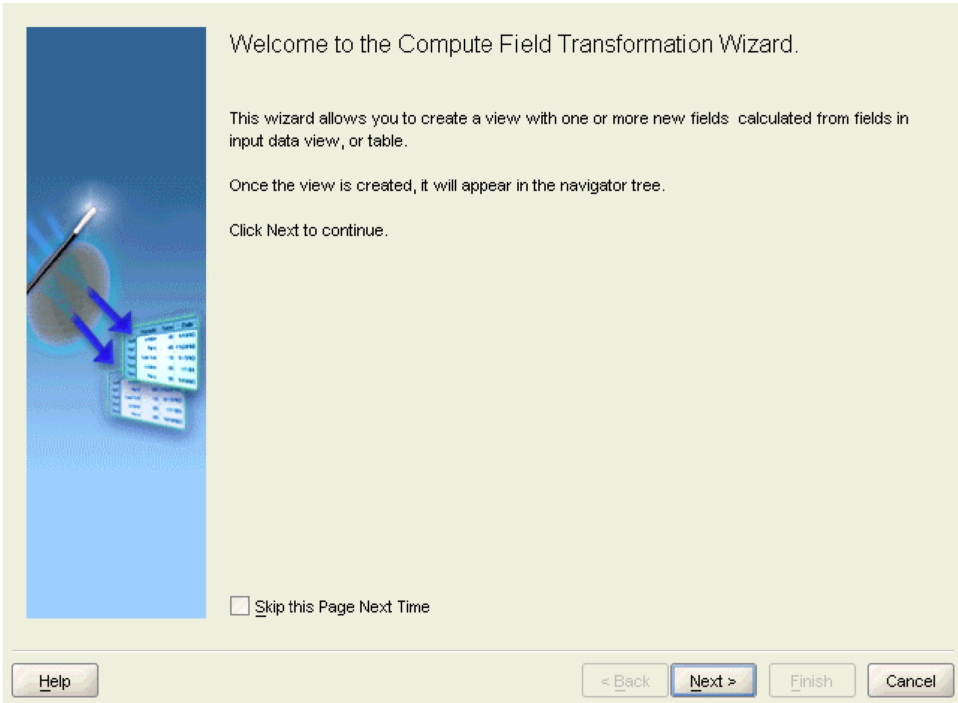
Click Finish to complete the transformation; a sample of the transformed data is displayed.

Compute Field

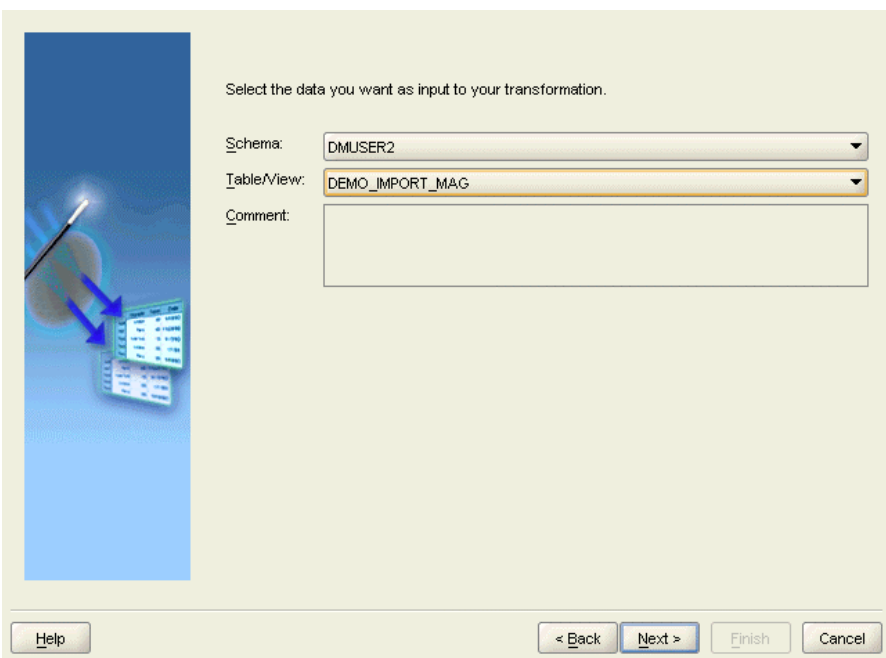
It is often necessary when preparing data for data mining to derive a new column from existing columns. For example, specific dates are usually not interesting, but the elapsed time in days between dates may be very important (calculated easily in the wizard as Date2 – Date1; the difference between two date types gives the number of days between the two dates in numerical format). Note also that the function SYSDATE represents the current date, so for example SYSDATE - DATE_OF_BIRTH gives AGE (in days).

The following example shows another viewpoint on Disposable Income as Fixed Expenses, calculated as (Income – Disposable Income).

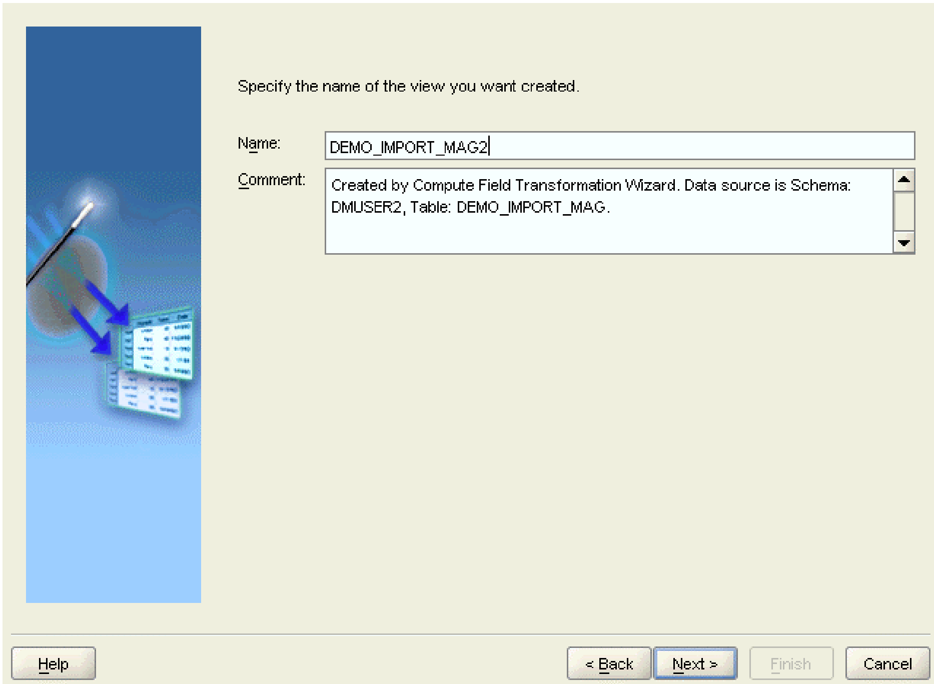
Begin by highlighting Transform on the Data pulldown menu and selecting Compute Field (or right-click on the table/view name) to launch the wizard.



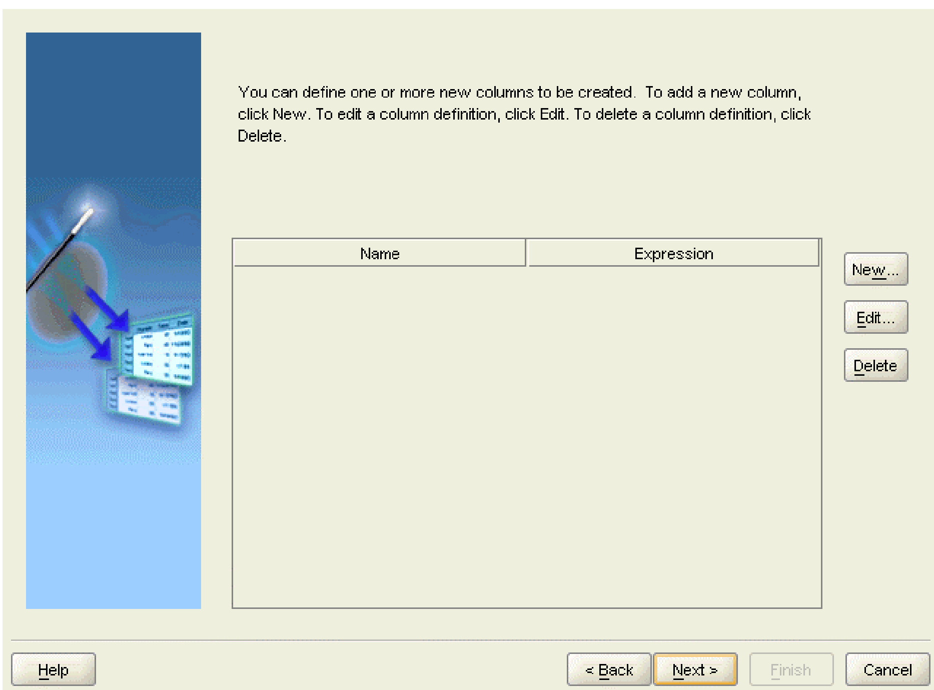
Select the table or view to be transformed (if you accessed the wizard by right-clicking the table/view name, then the data is already known and this step is skipped).



Enter the name of the view to be created.



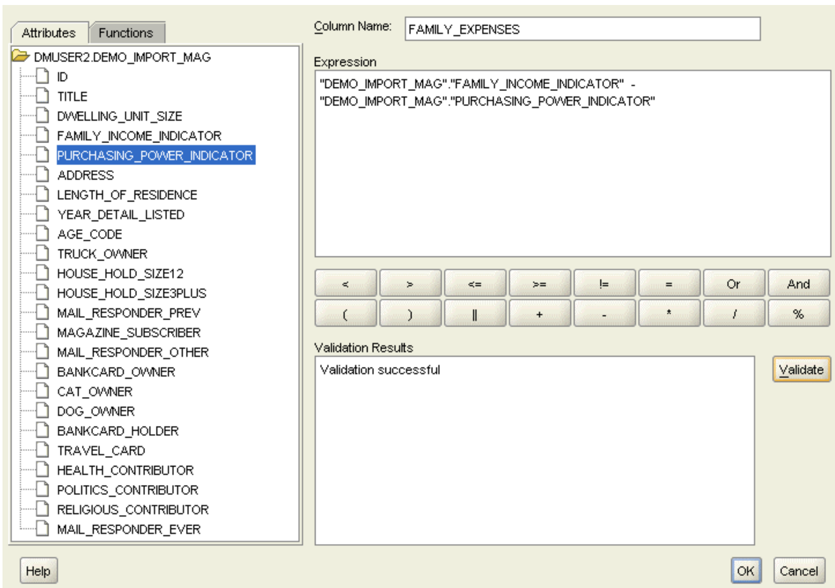
Click New to construct a definition of the new column.



In the Expression Editor, double-click on an attribute name to include it in the expression. Click on the appropriate buttons to include operators. Note that many SQL functions are available to be selected and included in the expression by clicking the Functions tab. Enter the new attribute name in the Column Name window.

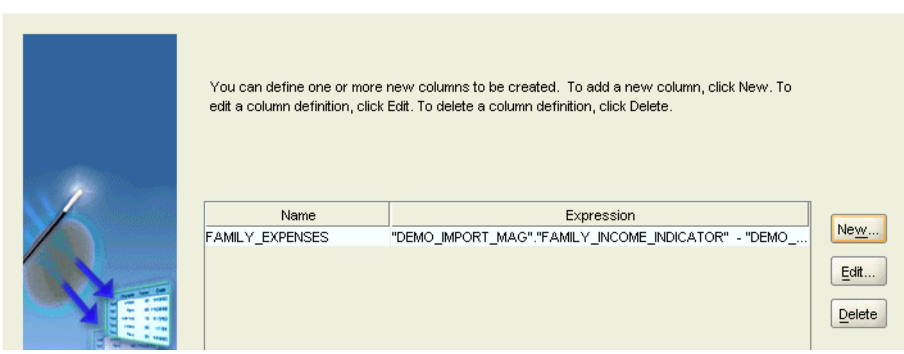
In this example, the new column FAMILY_EXPENSES is the difference of FAMILY_INCOME_INDICATOR and PURCHASING_POWER_INDICATOR.

You can check that the calculation is valid by clicking the Validate button.

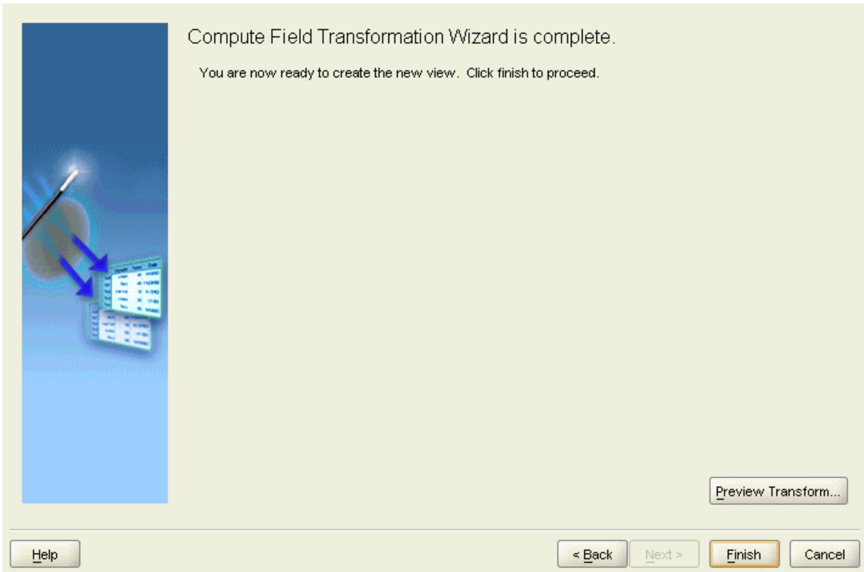


You may want to drop the columns FAMILY_INCOME_INDICATOR and PURCHASING_POWER_INDICATOR
after the result is created. This can be done by using the result as source in the Create View wizard and deselecting those columns (illustrated in the next section).

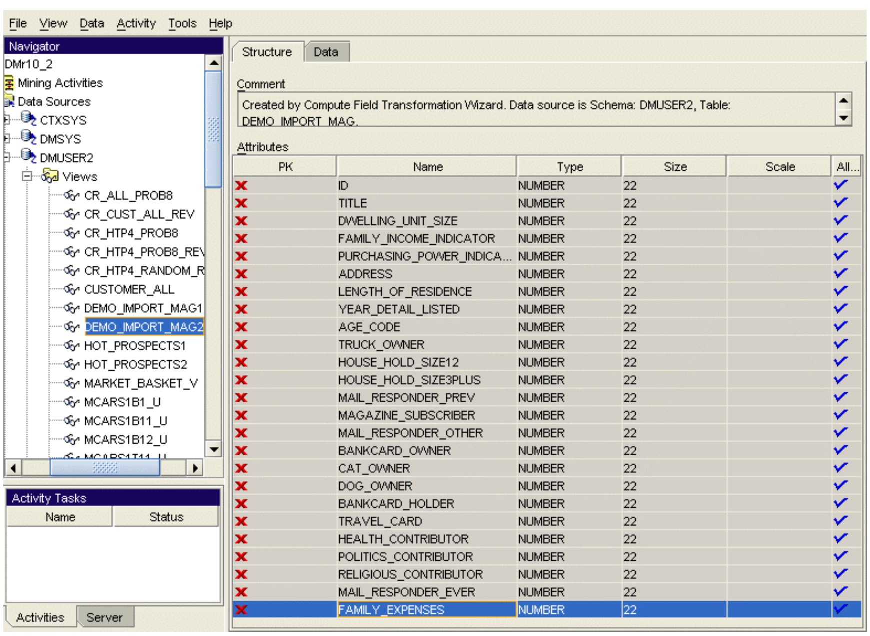
The column definition is displayed in the Define New Columns window; you may repeat the process to define other new columns in the same window.



You may preview the results and then choose to generate a stored procedure from the Finish page. Click Finish to complete the transformation.



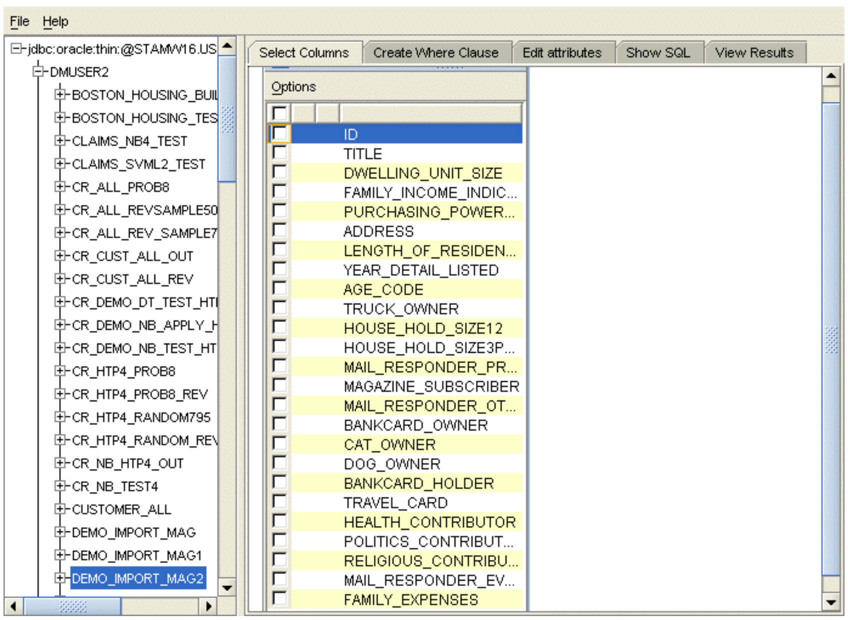
The view with the new column is displayed when the transformation is complete.



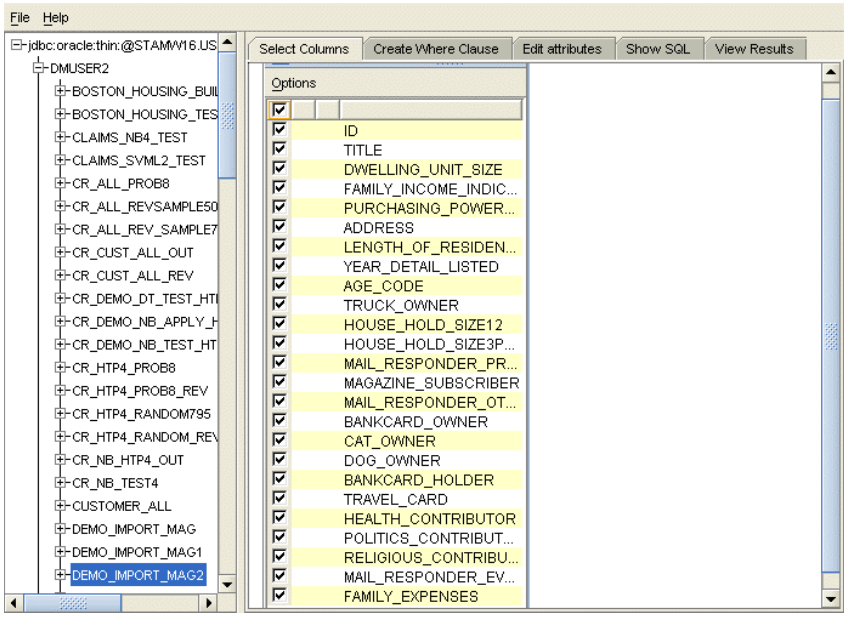
Create View Wizard

The Mining Activity Guides provide utilities for the combining of data from various sources, but there are times when the Create View wizard can be used independently of the Guides to adjust the data to be used as input to the data mining process. One example is the elimination of attributes (columns).

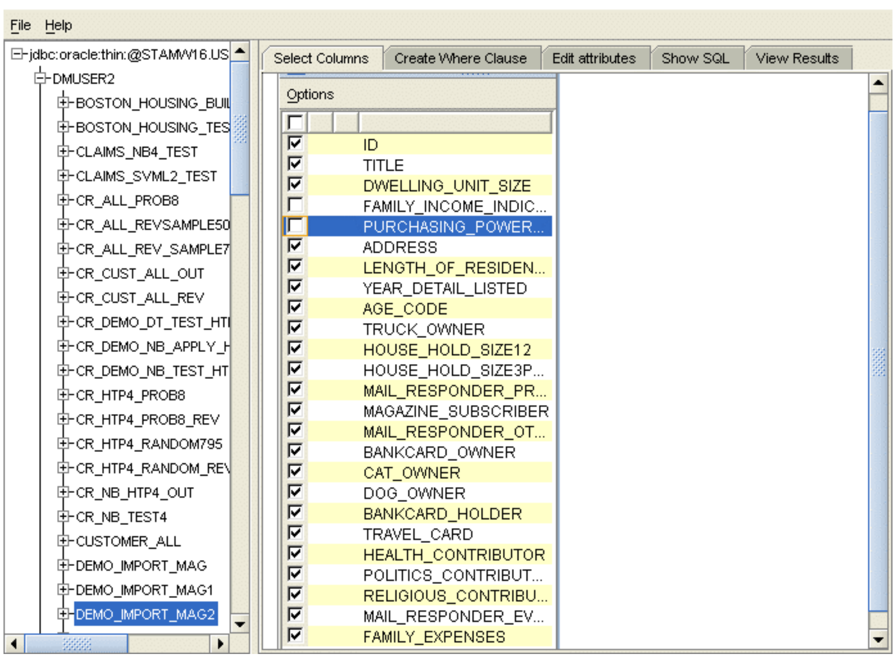
Begin by selecting Create View from the Data pulldown menu. Click the plus sign "+" next to the database connection to expand the tree listing the available schemas. Expand the schemas to identify tables and views to be used in creating the new view. Double-click the name DEMO_IMPORT_MAG2 (created in the previous section) to bring it into the work area.



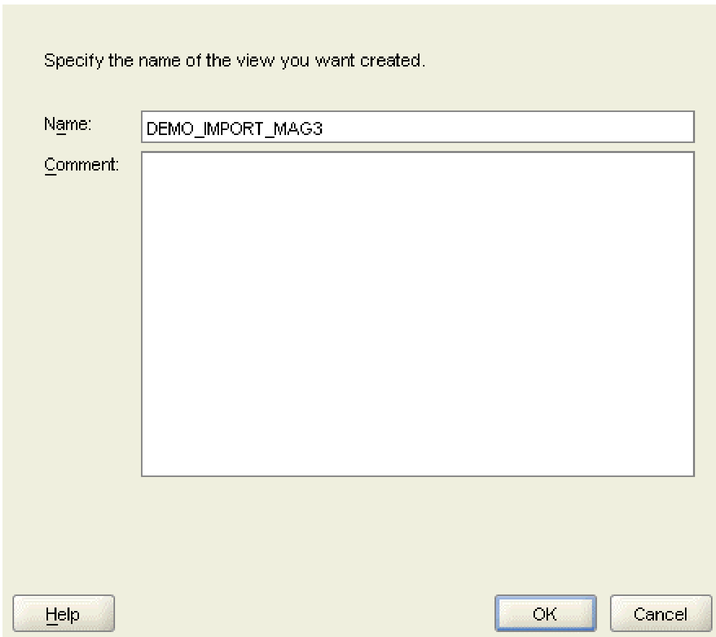
Click the checkbox next to an attribute name to toggle inclusion of that attribute in the new view; click the top checkbox to toggle all checkboxes.



Then click the checkboxes next to FAMILY_INCOME_INDICATOR and PURCHASING_POWER_INDICATOR to deselect those attributes.



Select Create View from the File pulldown menu and enter the name of the resultant view in the dialog box; then click OK.



When the view has been created, a sample of the data is displayed. Dismiss the Create View wizard by selecting Exit from the wizard's File pulldown menu.

Chapter 3 – Overview of Mining Activity Guides

When the data mining problem has been defined and the source data identified, there are two phases remaining in the data mining process: Build/Evaluate models, and deploy the results.

Oracle Data Miner contains activity guides for the purpose of carrying out these phases with the minimum of required intervention. Moreover, the implicit and explicit choices and settings used in the Build activity can be passed on seamlessly to the Apply or Test activities, so that many operations usually required are hidden or eliminated.

You can choose to let the algorithms and the activity guides optimize the settings internally; in that case, you need only identify the data (and target, if required), and specify the data mining algorithm. However, the expert who is familiar with the effects of parameter adjustments can choose to gain access to each of the parameters and can modify the operations manually.

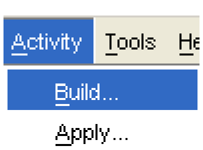
This chapter illustrates the appearance and steps presented in the Activity Guide wizards; the reasons behind the entries and choices will be explained in the discussions of individual algorithms.

The Build Activity

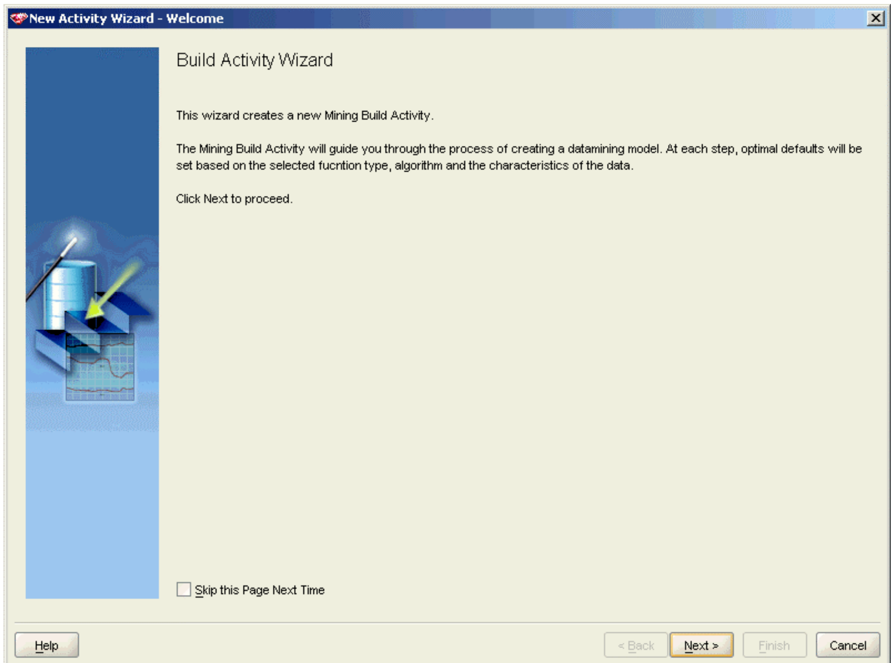
The Build Activity wizard allows you to:

- Identify supplemental data to add to the case table (the basic source data)
- Select the data mining functionality and algorithm
- Adjust the activity settings manually, rather than to accept automatic settings

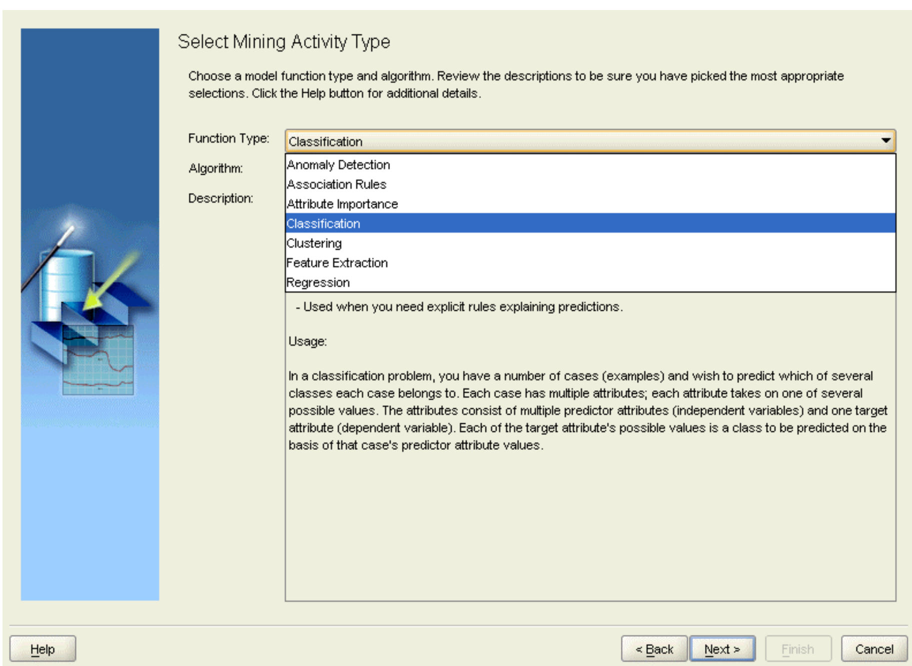
The Mining Activity Build wizard is launched from the Activity pull-down menu:



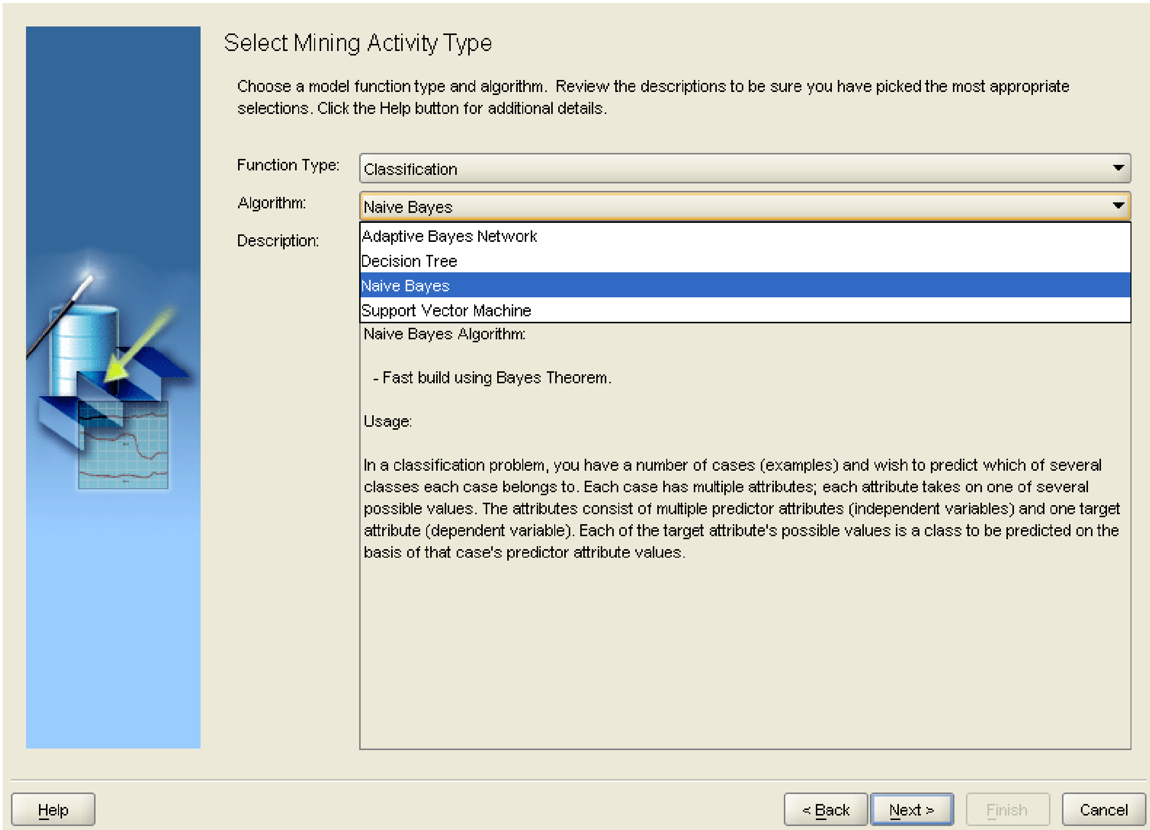
Select Build to activate the wizard and click Next on the Welcome page.



Choose the Function to use (this example uses Classification) and click Next:



Choose the algorithm to use (this example uses Naïve Bayes) and click Next:



Specifying Source Data

The next steps have to do with identifying the source data for the activity; in part these steps are dependent on the type of activity and the type of data available. Some typical steps are shown.

The Case Table or View

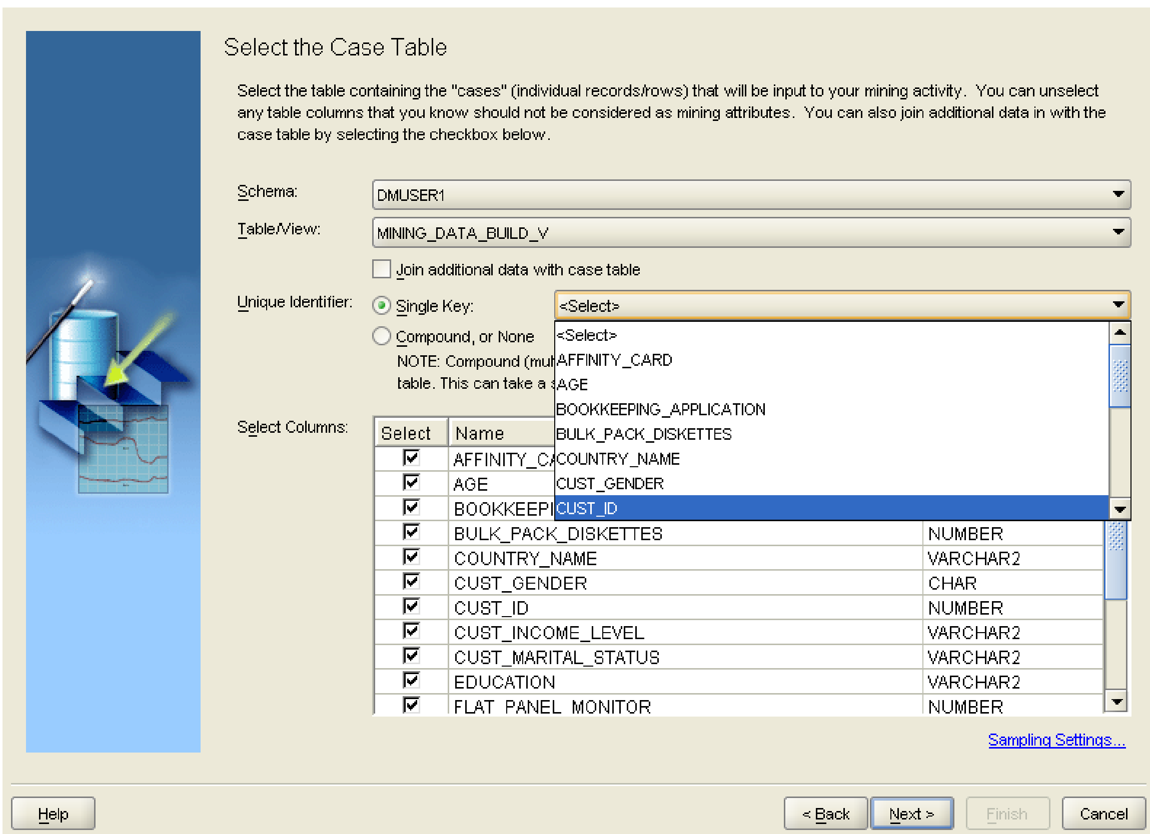
The “core” data has been identified (usually called the “case” table) and possibly transformed as discussed in Chapter 2. This example uses the view MINING_DATA_BUILD_V to build the model.

A later step in the wizard will employ heuristics to eliminate some attributes automatically; normally each attribute should remain selected in this step unless you know that it should be eliminated for some reason, such as a legal prohibition against using certain information in analysis.

The possibilities for gathering data are:

1. The case table or view contains all the data to be mined.
2. Other tables or views contain additional simple attributes of an individual, such as FIRST_NAME, LAST_NAME, etc.
3. Other tables or views contain complex attributes of an individual such as a list of products purchased or a list of telephone calls for a given period (sometimes called "transactional" data).
4. The data to be mined consists of transactional data only; in this case, the case table must be constructed from the transactional data, and might consist only of a column containing the unique identifiers for the individuals and a target column.

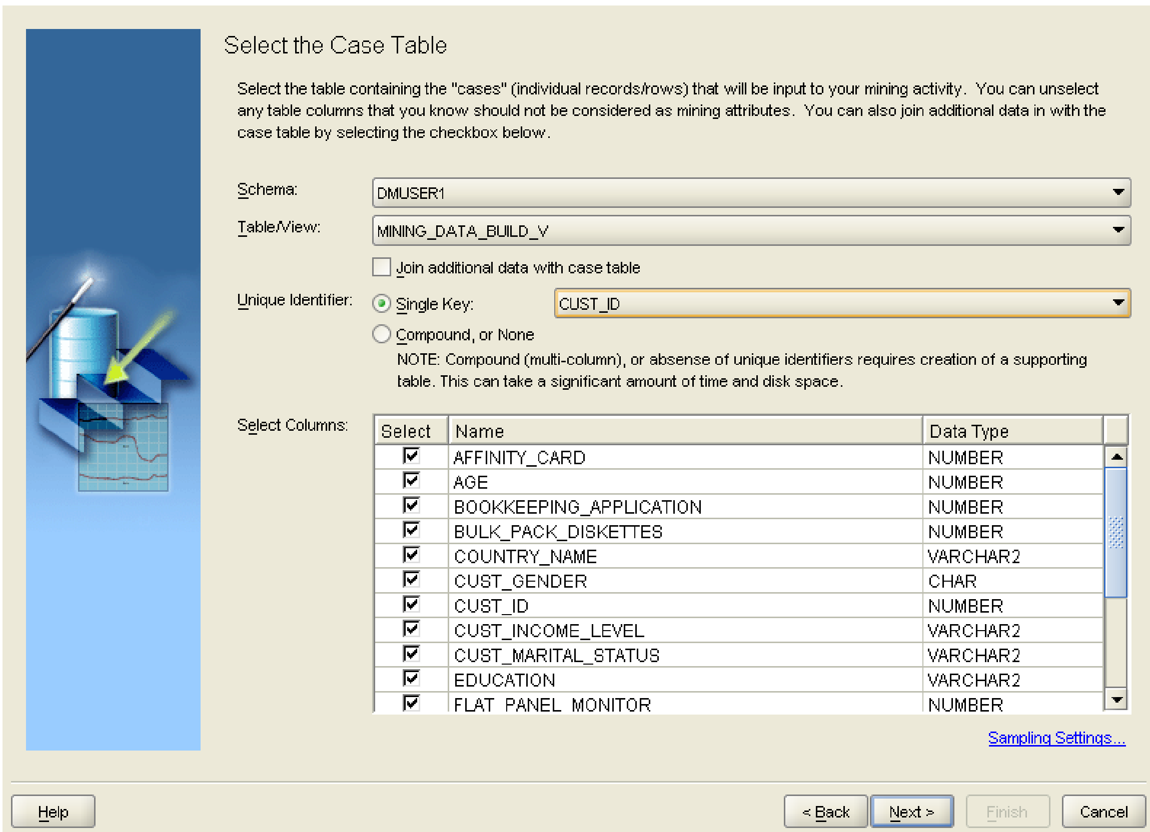
In each case, the unique identifier for each row must be selected from the pull-down menu as shown.



All statistics are based on a sample of the case table; the default is a random sample whose size, N cases, is determined by the Preference Settings (see Appendix B). If the data is very large, there may be performance and resource issues – you can click Sampling Settings to choose the first N rows rather than a random sample.

No Additional Data

Possibility 1 is the easiest; in Step 2 of the wizard, ensure that the box “Join additional data with case table” is not checked, and click Next to proceed directly to the Target selection step.

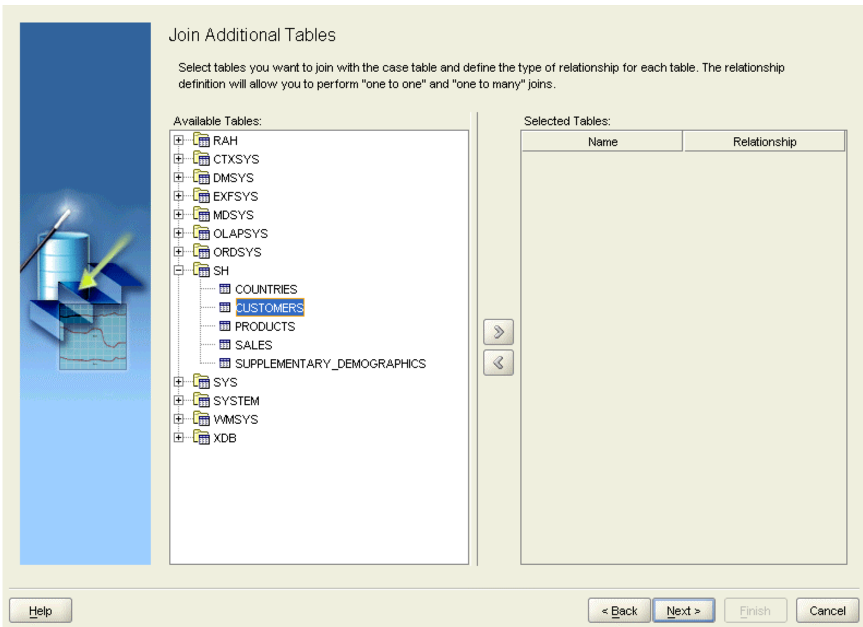


The other three possibilities require that you check the box; then clicking Next takes you to steps used to identify the additional data to include.

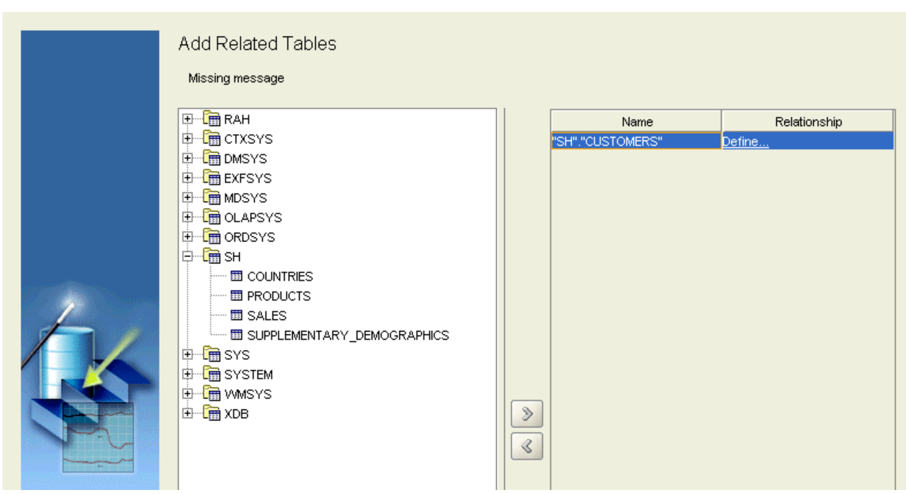
Simple Additional Data

Suppose that you wish to add customer_city from the table CUSTOMERS in the SH schema to each row of the base table (possibility 2).

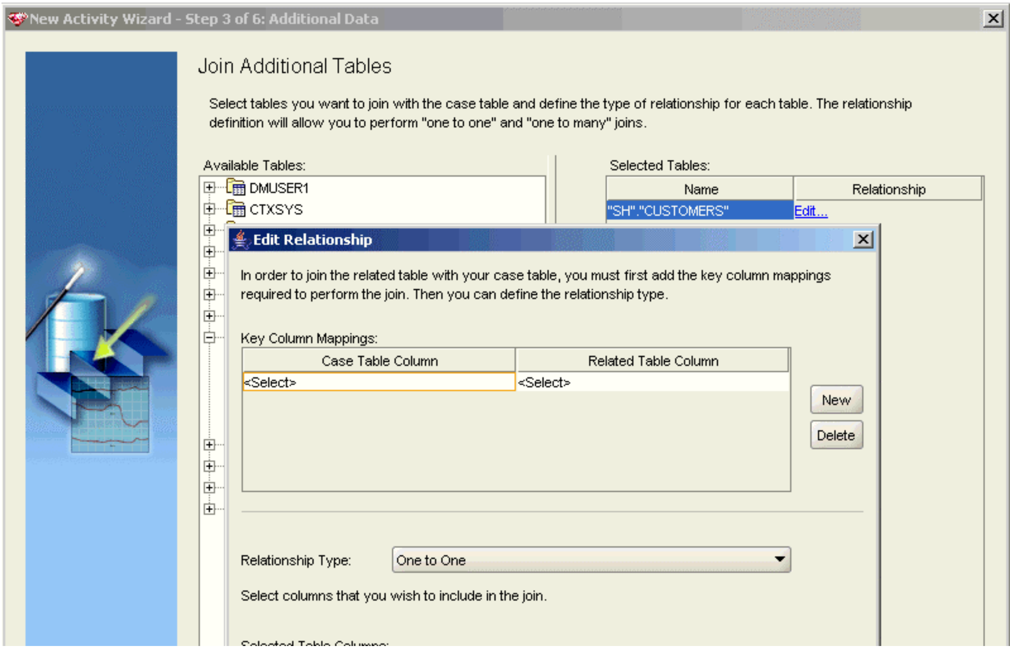
You will next see the page shown below in which you can expand the schema name to display the tables/views of that schema.



Highlight the table containing the desired attributes and click ">" to move the table into the right-hand frame.

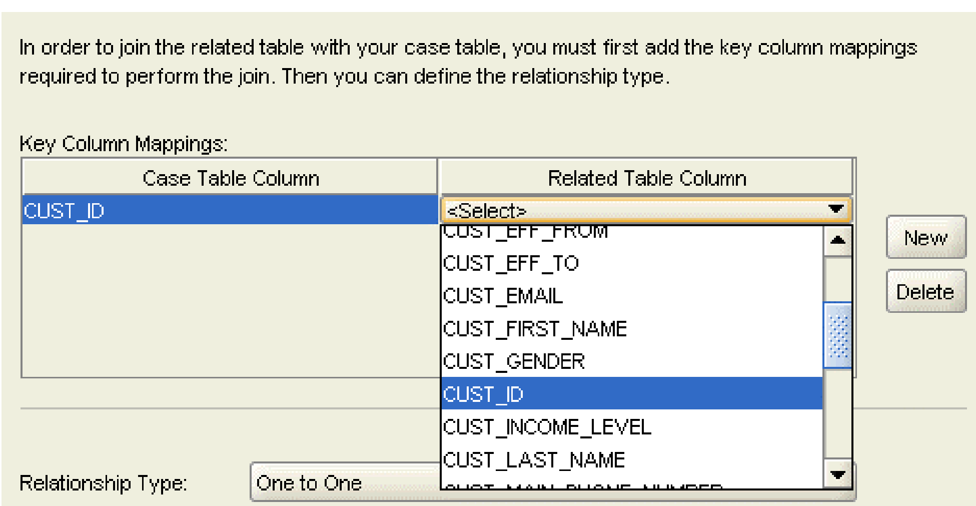


Click Define in the Relationship column to specify the matching identifiers in the two tables. A new window pops up.

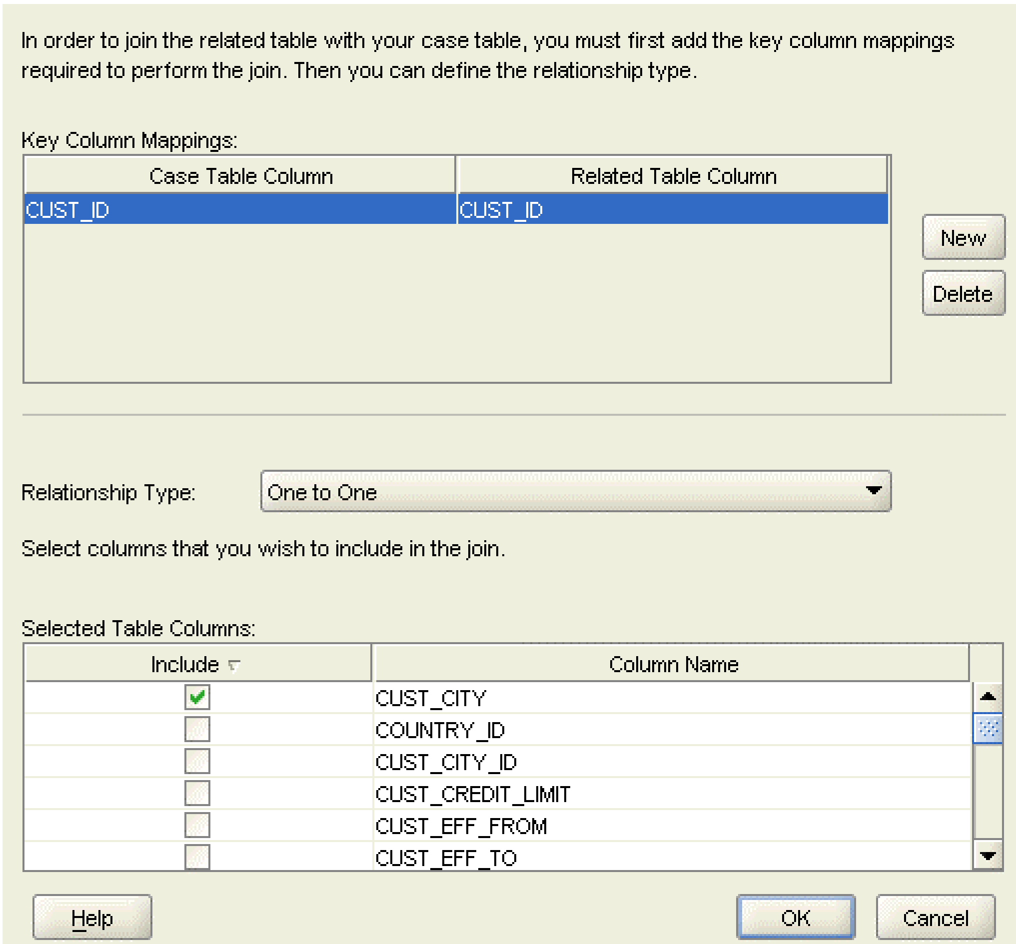


We know that CUST_ID in the Case table and CUST_ID in the related table use the same value to identify a customer uniquely, so click <select> in each column to choose the appropriate name from the list.

If more than one column is required to establish uniqueness, click New to add another column to the list.



This is a simple one-to-one relationship – one discrete piece of information for each individual in the related table is added in a new column in the case table, so in the Relationship Type pull-down menu, select One to One. Then click the appropriate checkboxe to include CUST_CITY as a new column in the input data. Then click OK to return to the Join Additional Tables screen, and if there are no other tables to join, click Next to proceed to input/target selection page.



Complex Additional Data

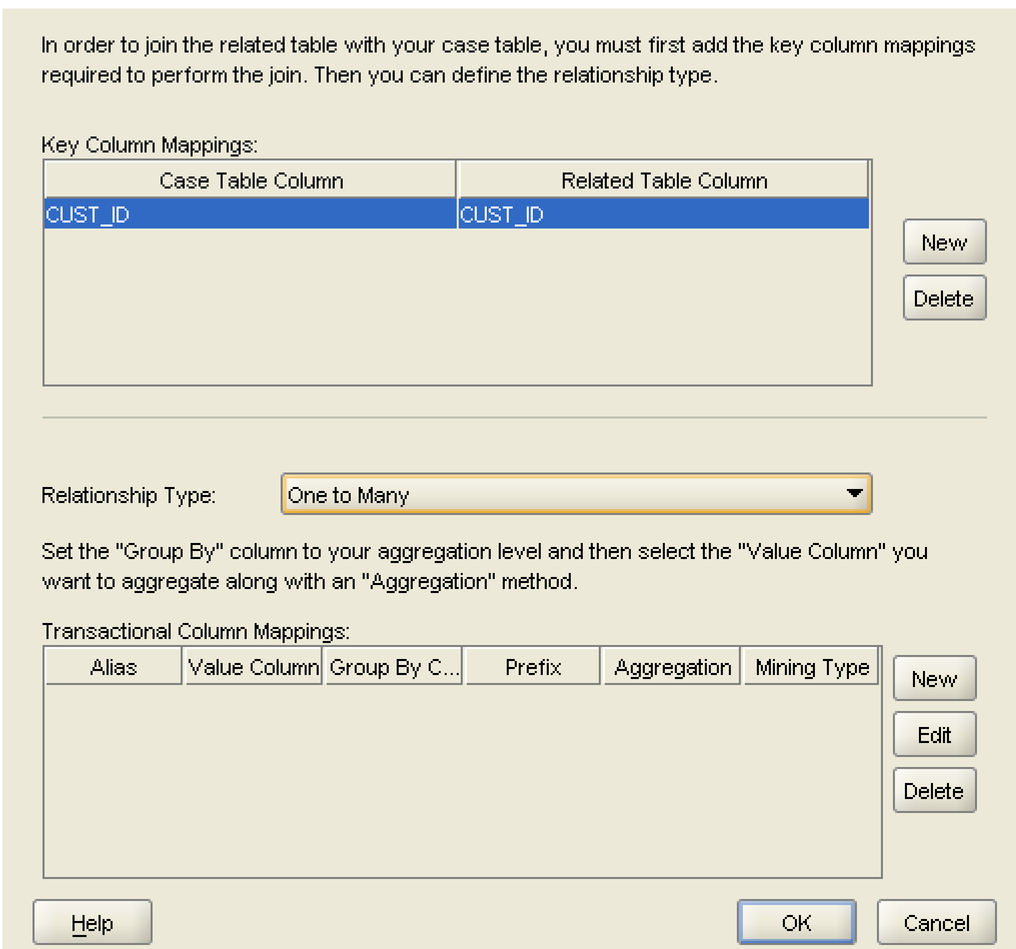
Suppose that you want to include an indication of purchases made by each customer in a certain period – the amount of money spent on each product by each customer.

This information is contained in the SALES table of the SH schema, so proceed as in the Simple Additional Data example to select the SALES table, and click Define to specify the identifier for each table.

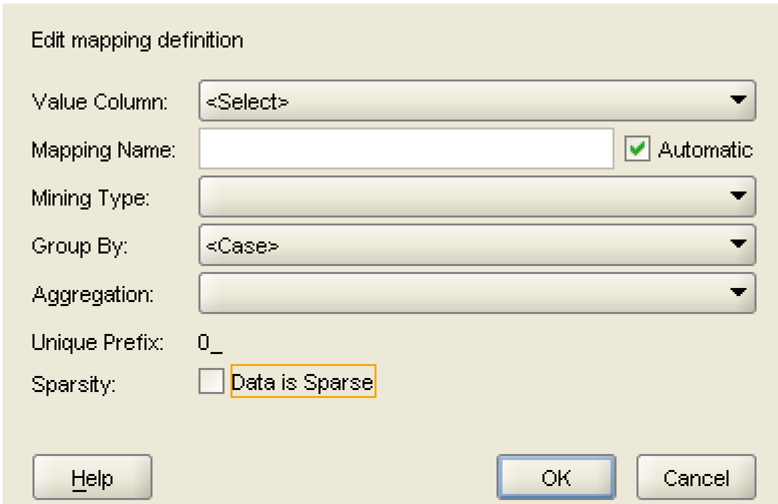
We want to include such information as the fact that customer #1234 spent $35 for Mouse Pads and $127 for Printing Supplies. This type of information can be accommodated in a single table row by creating what is called a Nested Column.

NOTE: Nested columns are not supported for the Decision Tree algorithm

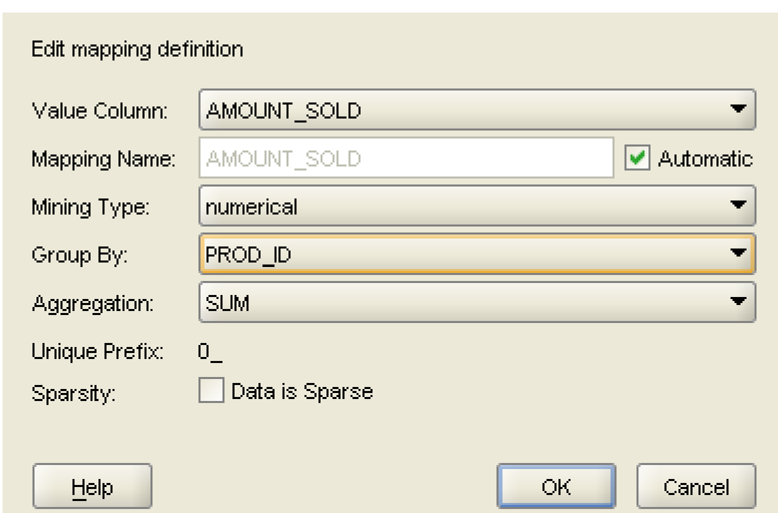
The relationship associates one customer with multiple purchases, so select One to Many in the pull-down menu.



Click New to launch a dialog box used to specify the complex value to be added to each customer’s row.



A new column will be created that can be thought of as a “table within a table”, with two columns labeled NAME and VALUE. NAME is the identifier for an entry in this column, such as the product ID for an item purchased (perhaps several times), and VALUE is the aggregated value, such as the total amount spent for purchases of a particular item. To avoid conflicts in names found in more than one nested column, a unique prefix is added to each NAME value in this nested column. The Mapping Name is an Alias for this new complex column, and by default it is the same as Value Column. For each CUST_ID in the case table, the entries are aggregated and grouped to get the list of products (PROD_ID) and total spent (AMOUNT_SOLD) for each product.



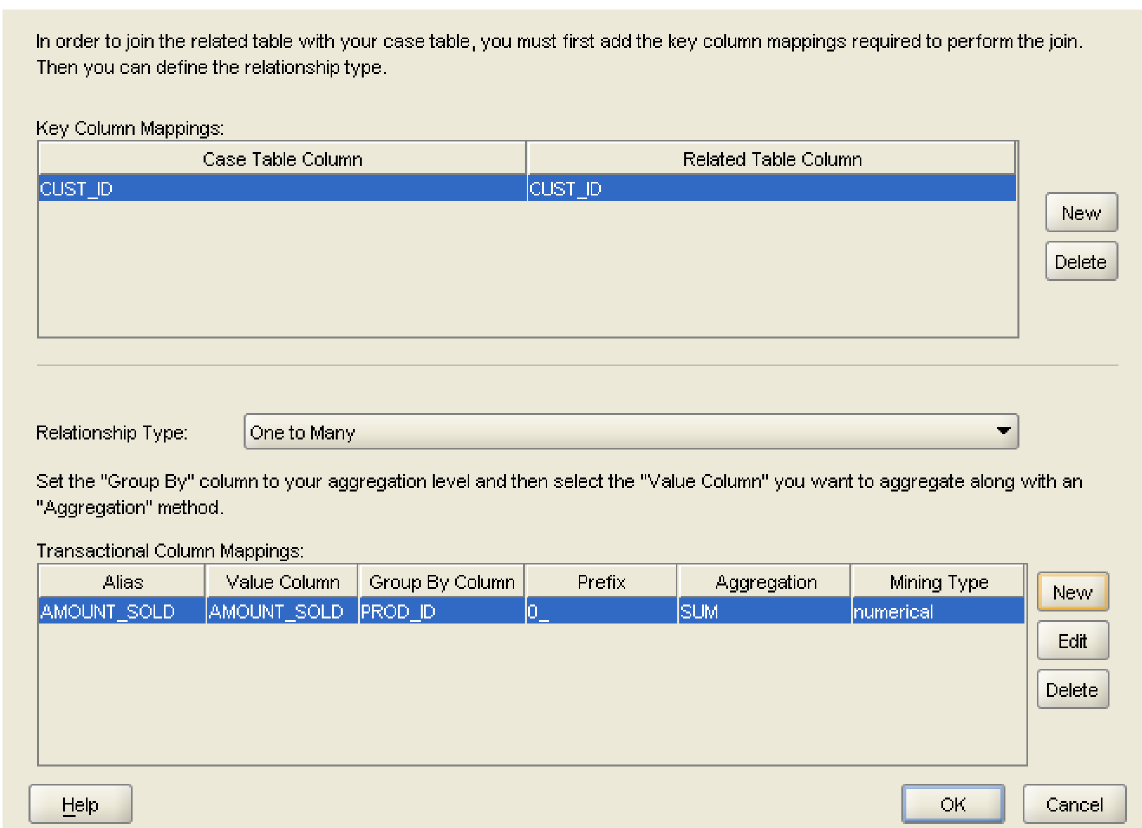
So for example, the source data for the mining operation will have one row for each customer, with a column named AMOUNT_SOLD containing the aggregated sales information for that customer. If customer 123 purchased:

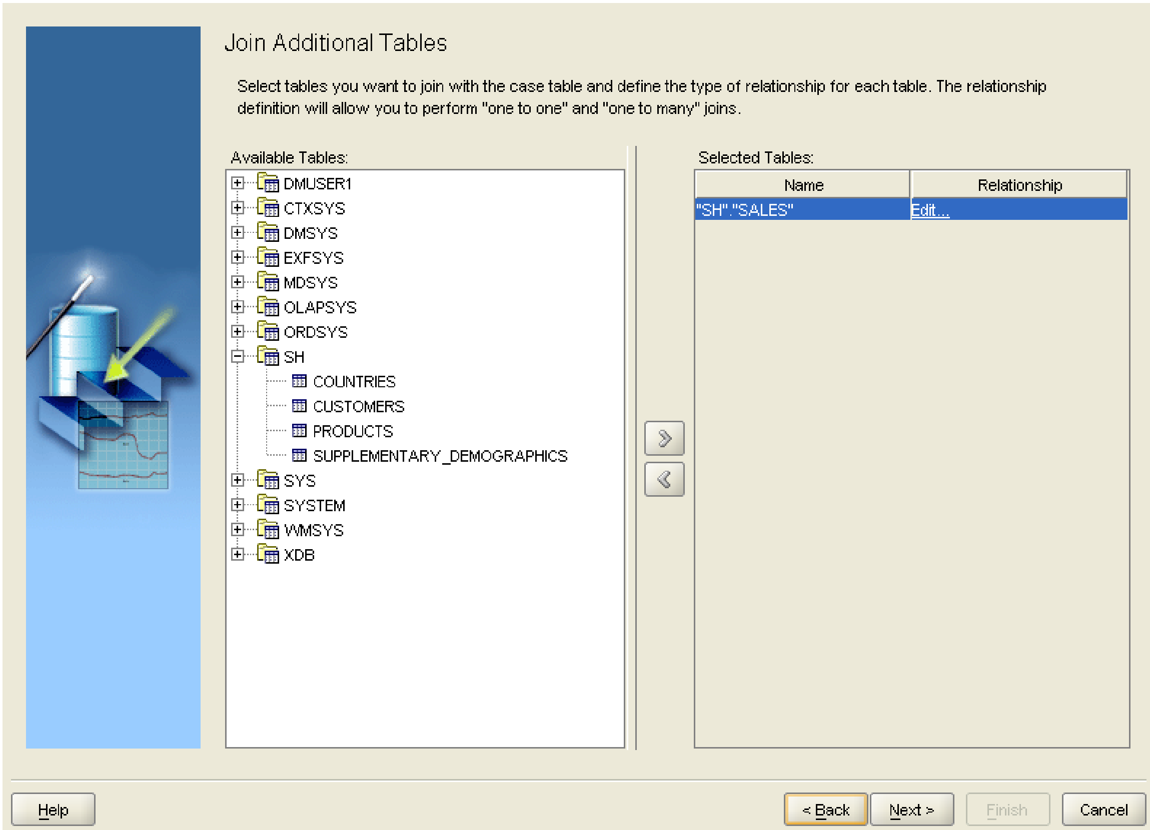
Prod1	$2
Prod23	$4
Prod1	$2
Prod1	$2
Prod23	$4

Then entry in the nested column AMOUNT_SOLD for Cust_id = 123 has the following form:

Prod1	6
Prod23	8

Click OK in the Edit Mapping Definition box to see the summary, and OK to return to the Join Additional Tables step.





Click Next to proceed to input/target selection page.

Transactional Data Only

In special situations such as in Life Sciences problems, where each individual may have a very high number (perhaps thousands) of attributes, all the data is contained in a transactional-format table. This table must contain at least the three columns indicating the unique case ID, the attribute name, and the attribute value.

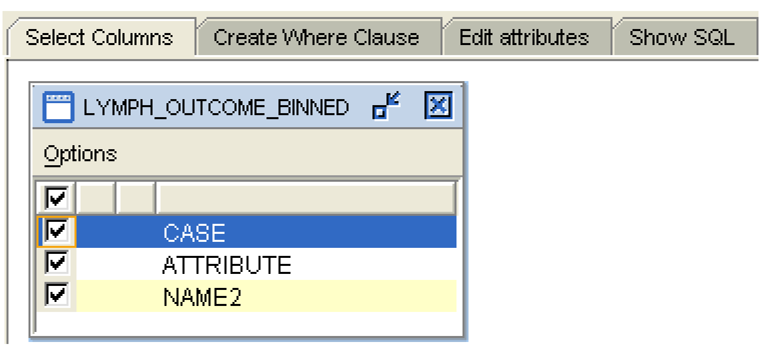
For example, the attributes may be gene expression names and the attribute value is a gene expression value. Typically, the attribute values have been normalized and binned to obtain binary values of 0 and 1 (representing, for example, that the gene expression for a particular case is above (1) or below (0) the average value for that gene.

For each case, there is one attribute name and value pair representing the target value – for example Target=1 means "responds to treatment" and Target=0 means "does not respond to treatment".

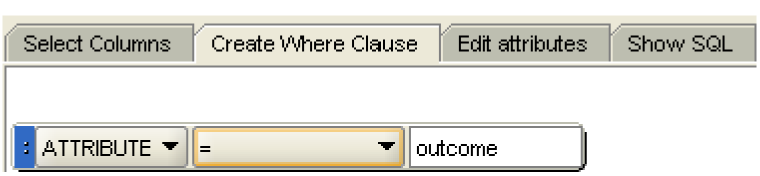
Suppose that we have a transactional table LYMPH_OUTCOME_BINNED with 5591 gene expressions for each of 58 patients and the binary target OUTCOME (0/1) indicating the success in treating Lymphoma patients. The business problem consists of the likely success in treating a particular patient based only on the values of gene expressions for that patient.

The first step is to separate the case table information (ID, OUTCOME) from the gene information to be joined in as a nested column.

In the Create View wizard, select the table and click the checkbox to include all three columns.



Then click the Create Where Clause tab and choose ATTRIBUTE = outcome.



After creating the view (File → Create View), eliminate the ATTRIBUTE column by using the Create View wizard again on the result to produce the case table with two columns containing only the case ID and the target value:

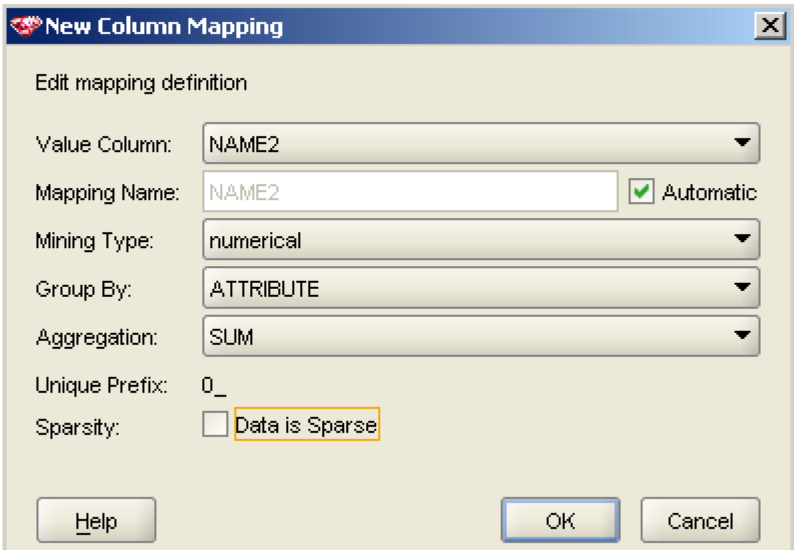
CASE	NAME2
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1
29	1
30	1
31	1
32	1
33	0
34	0
35	0
36	0
37	0
38	0
39	0
40	0
41	0
42	0
43	0
44	0
45	0
46	0
47	0
48	0
49	0

The final data preparation step involves removing the target values from the original table LYMPH_OUTCOME_BINNED using the Filter Single Record transform, explained in Chapter 2. The entry in the Expression Editor is:

Expression

```
"LYMPH_OUTCOME_BINNED"."ATTRIBUTE"   != 'outcome'
```

When using this data in an Activity Guide, the transactional data will be joined to the case table in one-to-many format as follows:



Since there is only one occurrence of a given gene expression value for each patient, the choice of SUM for the aggregation ensures that the values will not be aggregated at all (that is, SUM acts as a NO_OP since there's only one number to "add up" for each case), and the entry in the nested column for a particular patient is exactly the list of gene expression values for that patient.

In this example, each patient has a value for every gene expression, so the data is not sparse; ensure that the appropriate checkbox is cleared.

Click OK and OK again to return to the Join Additional Data page.

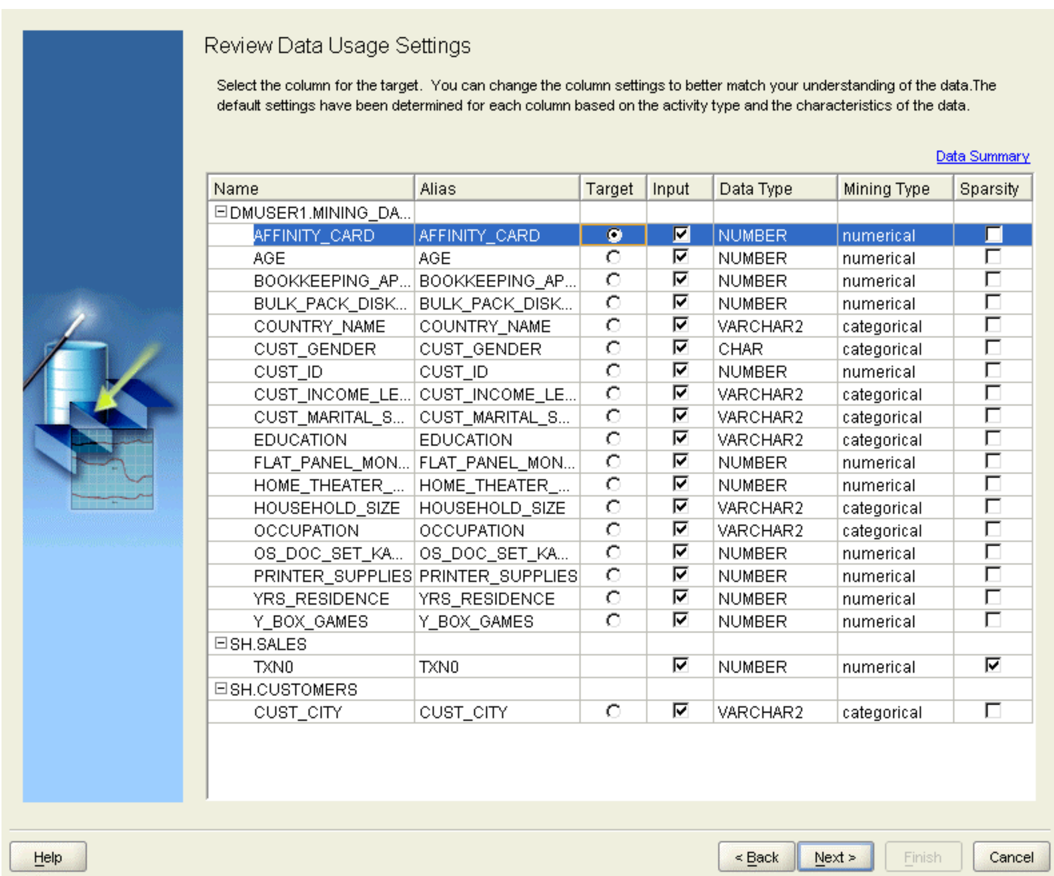
When the join operation dialog is completed, Click Next to proceed to input/target selection page.

Review Data Usage Settings

The case table is displayed along with the data joined in from the other tables; a combination from the second and third join types is shown: the column copied from CUSTOMERS and the new nested column, assigned the alias name TXN1, containing the transactional data derived from several columns in SALES.

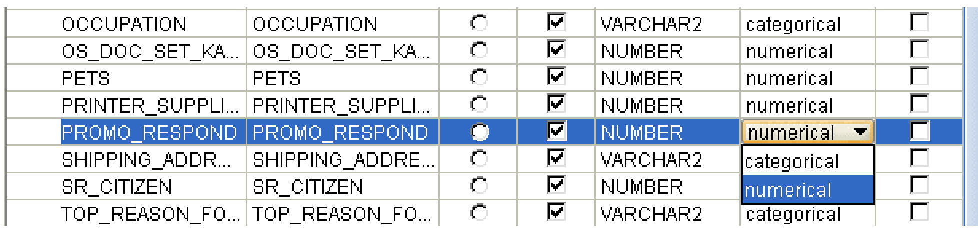
For a Classification, Attribute Importance, or Regression problem, the target, that is the attribute to be predicted, must be specified. In this example, the target is AFFINITY_CARD; click the radio button to indicate the target.

An attribute can be dense or sparse; sparsity is normally a measure of the percentage of cases with NULL value for that attribute. In the case of a nested column containing transactional data, sparsity is an indication of the percentage of possible values included. For example, if an average customer's records show the purchase of 4 of a possible 10,000 products, then that transactional attribute is sparse. Internal heuristics are applied to assign a checkmark or not to the sparsity indicator on this page; you can change the indicator if you have knowledge contradicting the heuristics.

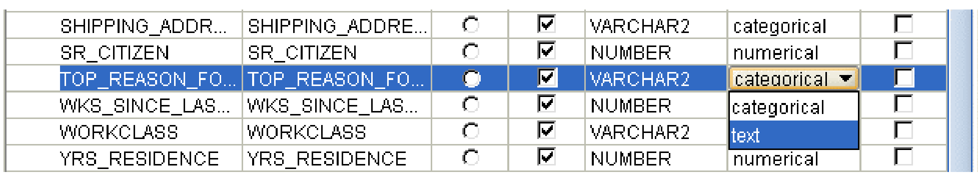


The value in a column has a data type in the definition of the table or view in the database, but the types seen by the data mining engine are different. For example, a NUMBER data type indicating age is numerical from a mining viewpoint, but the numbers 1, 2, and 3 used as labels to indicate Low, Medium, and High are not numerical, and should be described as categorical for mining purposes.

A structured character string such as the values in a column COLOR, with possible values RED, GREEN, and BLUE, is categorical for mining purposes, but unstructured text, such as physician's notes about a patient, should have mining type text. Internal heuristics are used to assign a mining type, but you can change the assignment by clicking the type and selecting from a pull-down menu as shown below:



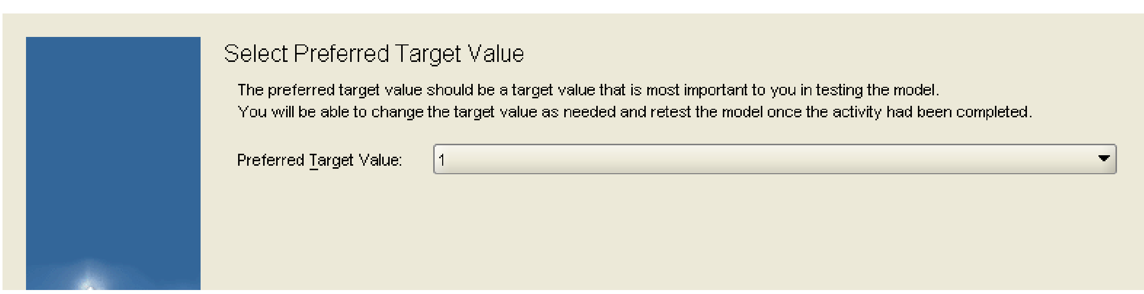
OCCUPATION	OCCUPATION			VARCHAR2	categorical	
OS_DOC_SET_KA...	OS_DOC_SET_KA...			NUMBER	numerical	
PETS	PETS			NUMBER	numerical	
PRINTER_SUPPLI...	PRINTER_SUPPLI...			NUMBER	numerical	
PROMO_RESPOND	PROMO_RESPOND			NUMBER	numerical	
SHIPPING_ADDR...	SHIPPING_ADDRE...			VARCHAR2	categorical	
SR_CITIZEN	SR_CITIZEN			NUMBER	numerical	
TOP_REASON_FO...	TOP_REASON_FO...			VARCHAR2	categorical	



SHIPPING_ADDR...	SHIPPING_ADDRE...			VARCHAR2	categorical	
SR_CITIZEN	SR_CITIZEN			NUMBER	numerical	
TOP_REASON_FO...	TOP_REASON_FO...			VARCHAR2	categorical	
WKS_SINCE_LAS...	WKS_SINCE_LAS...			NUMBER	categorical	
WORKCLASS	WORKCLASS			VARCHAR2	text	
YRS_RESIDENCE	YRS_RESIDENCE			NUMBER	numerical	

Select the Preferred Target Value

The choice on this page indicates which of the target values is the object of the analysis. In the Affinity Card problem, the preferred customer is the high-value customer, designated by the target value 1. Select 1 for the Preferred Target Value and click Next.

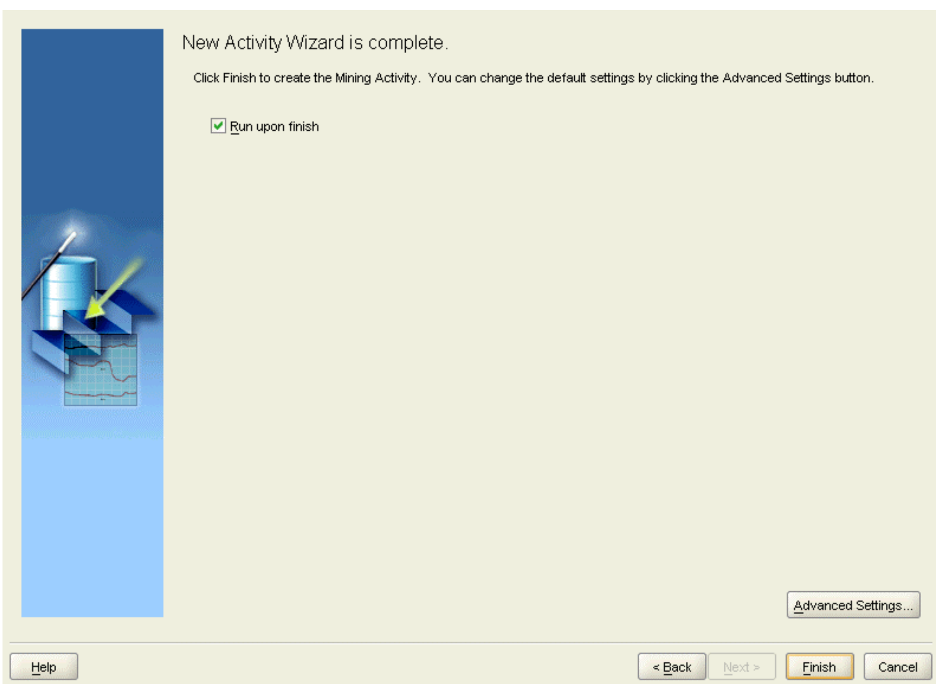


Activity Name

Enter a descriptive name for the activity and click Next

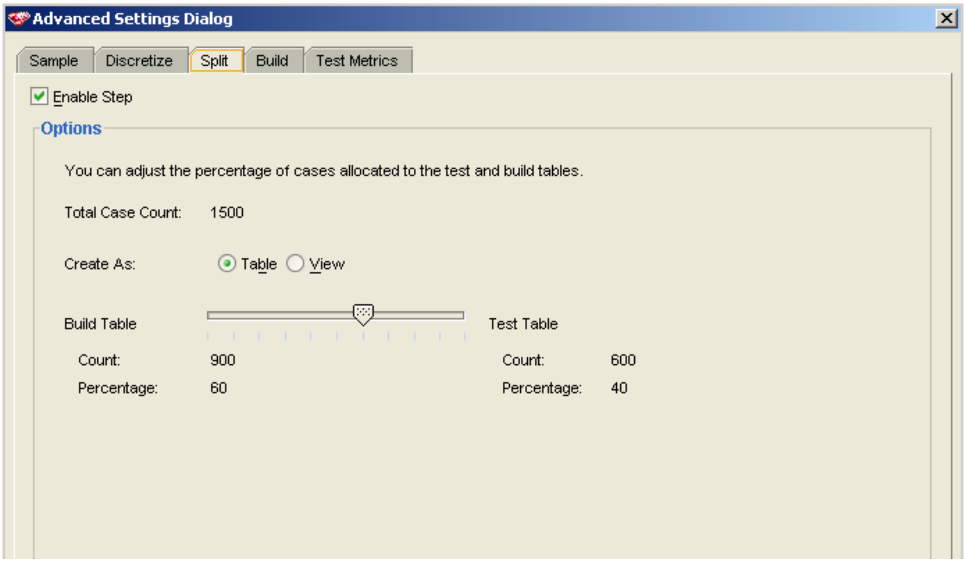
Finishing the Wizard

The last step of the Activity wizard presents the opportunity to automate the process from this point with no further intervention required.



Click “Advanced Settings” to modify any of the settings for any step in the Activity.

Shown below are settings pertaining to splitting the data into Training and Testing subsets. These and other parameters will be explained in the sections discussing activities for individual algorithms. Click OK to return to the Finish page.

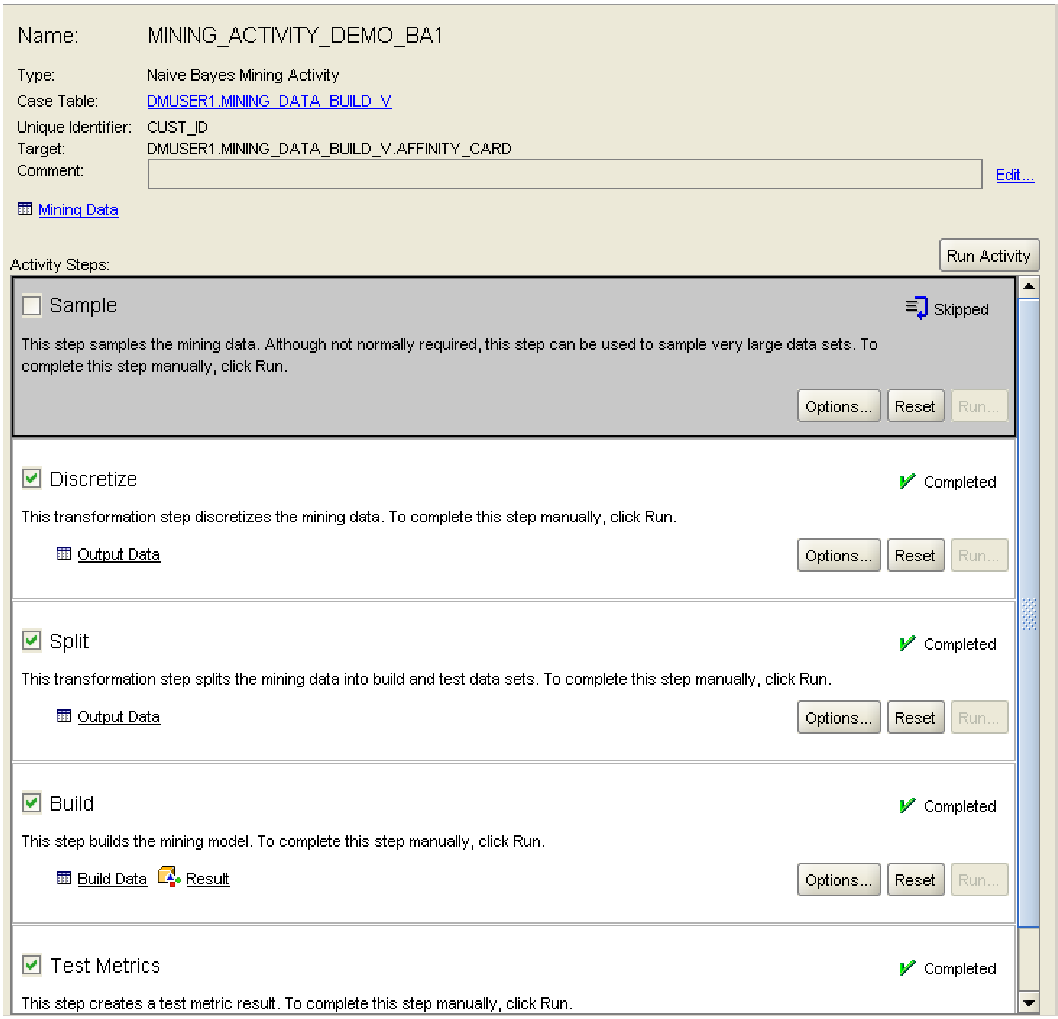


Check “Run upon Finish” to take advantage of default and optimized settings (and your own input if you chose Advanced Settings) throughout the Activity.

When the Activity wizard is completed, the steps appropriate to the chosen activity are displayed. If you chose Run upon Finish, the steps are executed to completion in sequence and a check appears on the right side of each step as it is completed.

If you didn’t check Run Upon Finish, you can click Options in any step to adjust settings. Then click Run Activity at the upper right (below the Edit button) to execute the entire sequence, or click Run within a Step to execute that step alone.

The steps and settings for each activity will be discussed in later sections.



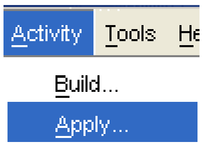
If you joined in additional transactional data, you can see the content of the nested column by clicking Mining Data. However, once the model building begins, the nested column is not visible, so you won't see the column contents if you click Build Data in the Build step.

If the activity builds a supervised model, then the model is applied to the hold-out sample created in the Split step and measurements of the model's effectiveness are reported in the test Metrics step. The display of the test results allows you to select the model that best fits your business problem. You can then apply that model to new data to produce results that will benefit your business. Details are explained in later chapters.

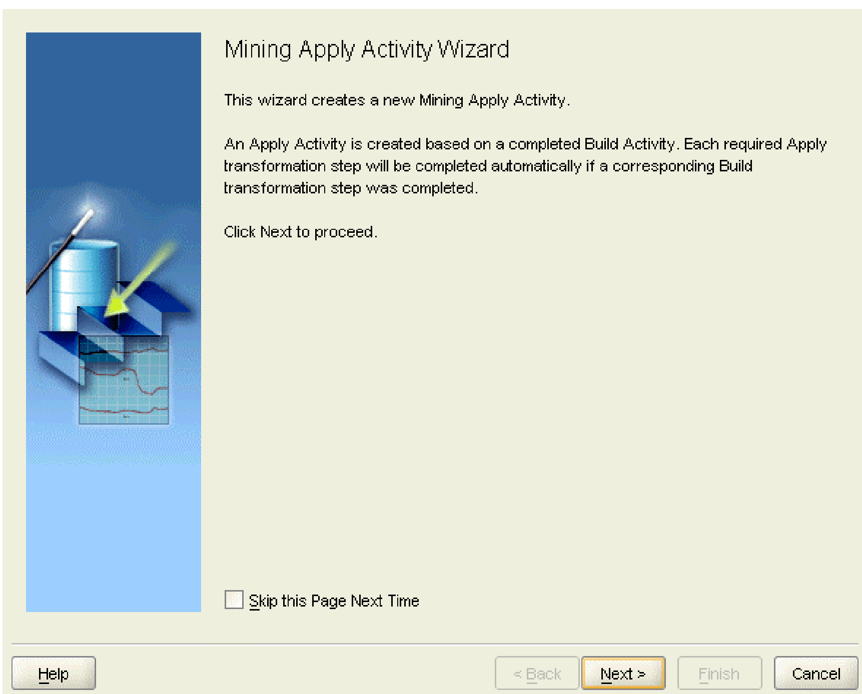
After an activity completes, you can click Reset, then Options in any step, change settings, and click Run Activity to execute the activity from the changed step to conclusion. Note that this action overwrites results obtained before the changes.

The Apply Activity

Launch the Activity Guide Apply wizard from the Activity menu:

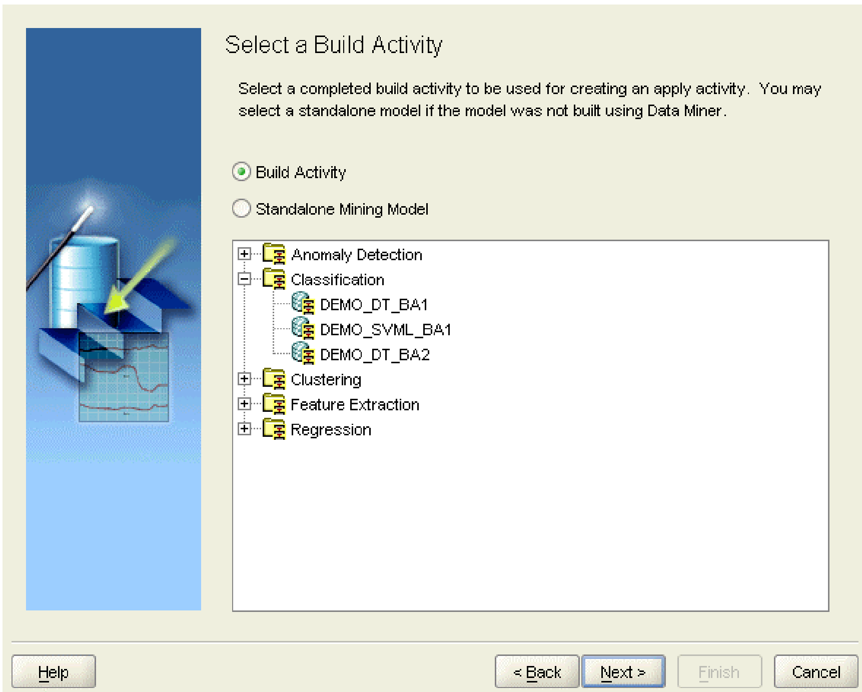


When a model is applied to new data, the input data must be prepared and transformed in exactly the same way that the source data for the Build activity was prepared. As noted on the Welcome screen, the Apply activity is based on a Build activity, and the Build activity will pass to the Apply activity whatever knowledge is required to prepare the input data appropriately. Click Next to proceed.



Select the Build Activity that was used to create the model, and all the information about data preparation and model metadata will be passed to the apply activity. The only decisions required relate to the format of the output.

The Apply Activity will be discussed more in relation to individual algorithms.

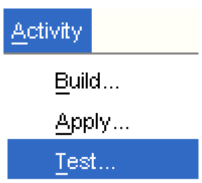


The Test Activity

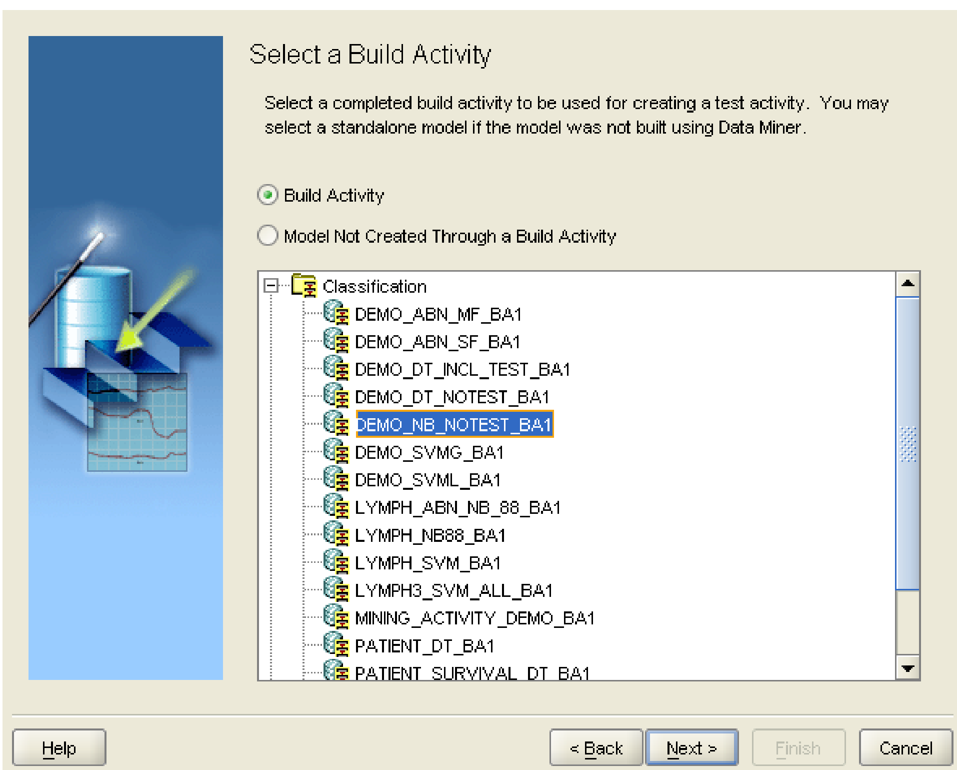
Under most circumstances for Supervised Learning problems, you will rely on the Build Activity to split the data into two mutually exclusive subsets, one used to build the model, the other used in the Test Metrics step of the activity.

However, if the data is already split into Build and Test subsets, you can run a Build activity and specify that the Split step and the Test Metrics step be skipped (by clearing the checkbox in the Split and Test Metrics tabs of Advanced Settings on the Finish page). Then you can launch a separate Test Activity to create the Test Metrics results.

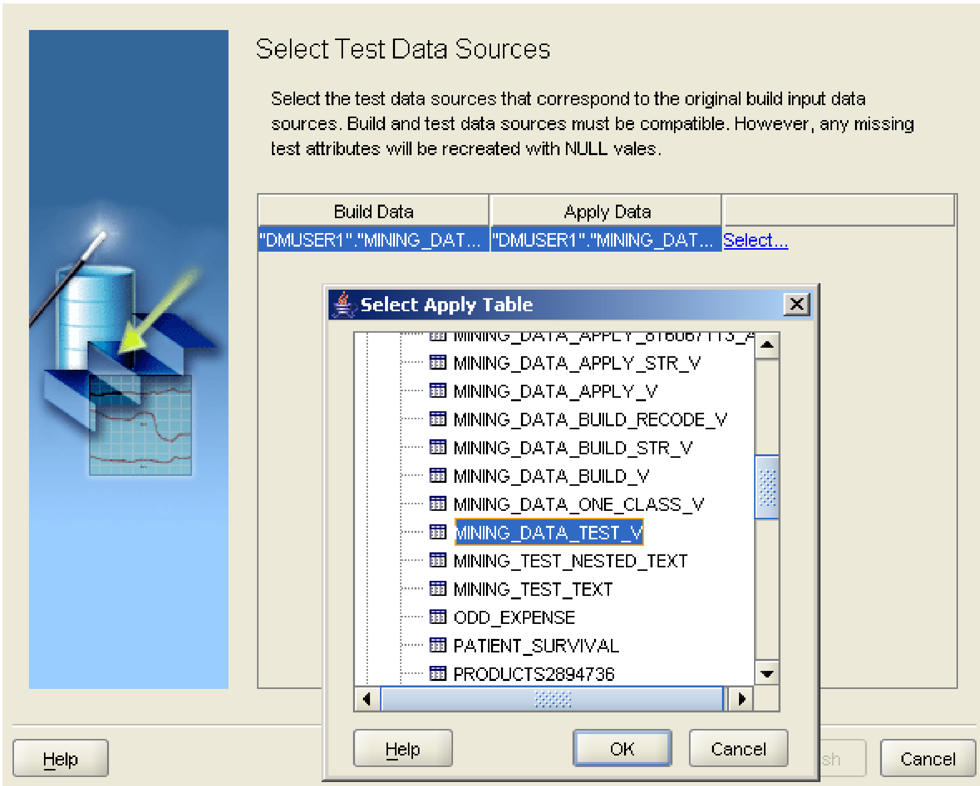
Select Test in the Activity pull-down menu:



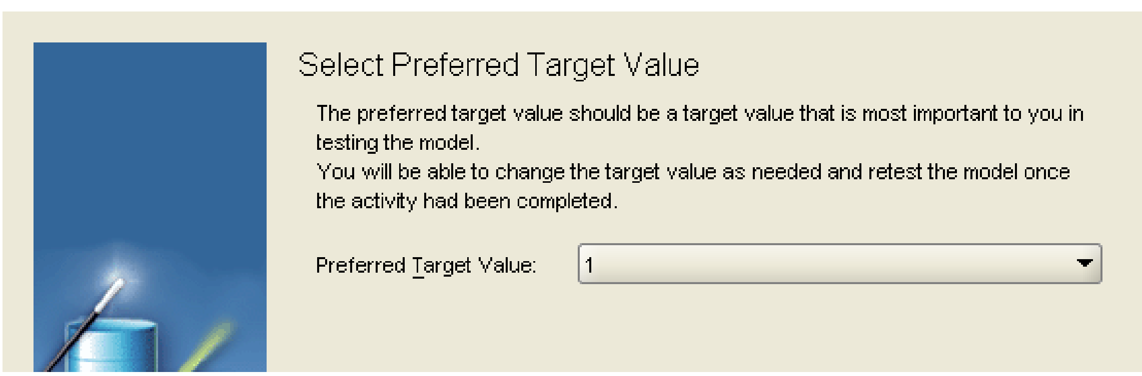
In Step 1 of the Test activity you identify the Build activity that created the model that you wish to test, then click Next.



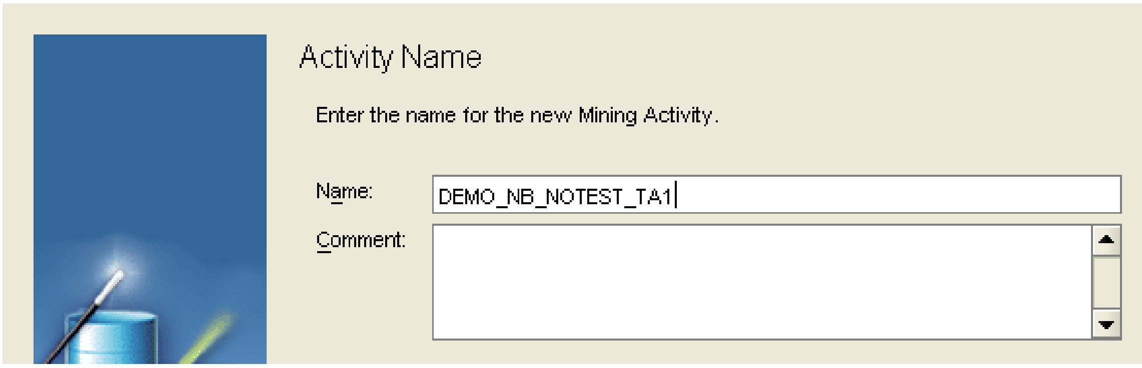
In Step 2, click Select and highlight the table/view to be used for the testing, then click Next.



In Step 3, designate the preferred target value (as in the Build activity), then click Next.



In Step 4, enter a name that relates the Test Activity to the Build Activity, then click Next and click Finish on the final page.



When the activity completes, you will be able to access the Test Metrics.

Chapter 4 – Attribute Importance

If a data set has many attributes, it's likely that not all will contribute to a predictive model; in fact, some attributes may simply add noise - that is, they actually detract from the model's value.

Oracle Data Mining provides a feature called Attribute Importance (AI) that uses the algorithm Minimum Description Length (MDL) to rank the attributes by significance in determining the target value.

Attribute Importance can be used to reduce the size of a Classification problem, giving the user the knowledge needed to eliminate some attributes, thus increasing speed and accuracy.

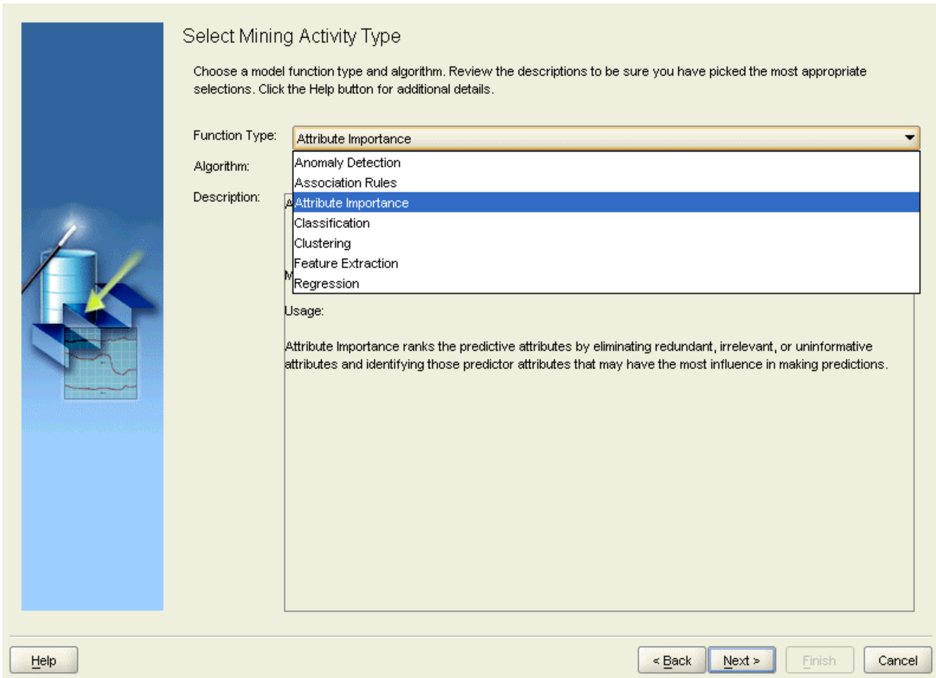
Note that the Adaptive Bayes Network and the Decision Tree algorithms include components that rank attributes as part of the model build operation, so Attribute Importance is most useful as a preprocessor for Naïve Bayes or Support Vector Machines.

Recall that the view MINING_DATA_BUILD_V discussed in Chapter 2 represents the result of a test marketing campaign. A small random sample of customers received affinity cards (sometimes called "loyalty cards", swiped at the point of sale to identify the customer and to activate selected discounts) and their purchases were tracked for several months. A business decision defined a threshold of spending; anyone spending more than the threshold amount is called a "high-revenue customer", and the value 1 is entered in the AFFINITY_CARD column for that customer.

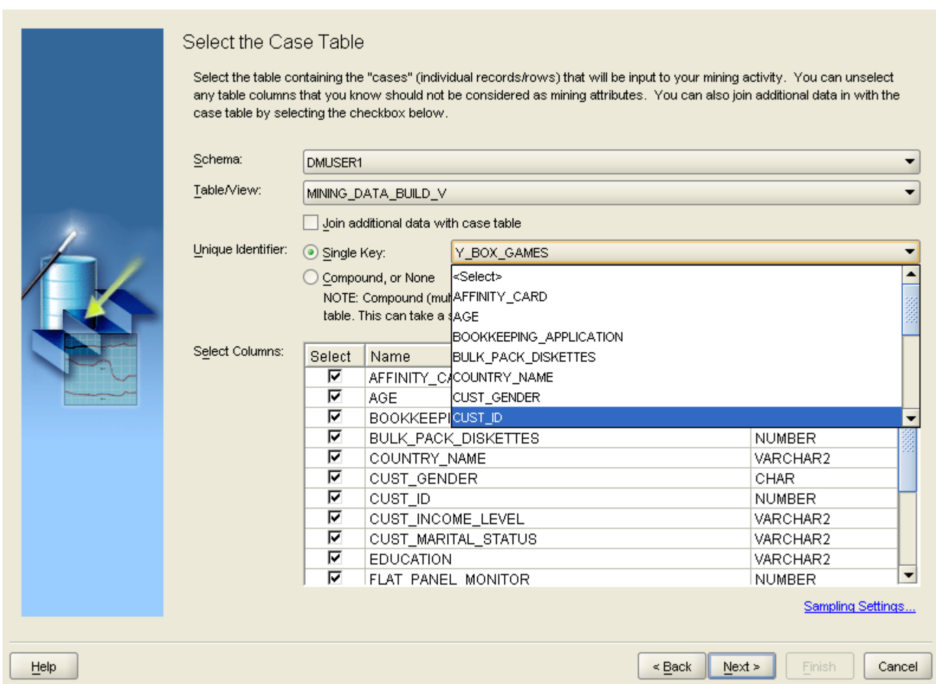
The business problem consists of identifying the likely high-revenue customers from among all customers for the purpose of offering incentives to increase loyalty among high-revenue customers. The data mining solution consists of building a predictive model from the results of the test campaign that can be applied to the entire customer base in order to distinguish the most valuable customers from the others.

The first question is: "what characteristics are the best indicators of a high-revenue customer?" To determine the answer, an Attribute Importance activity is defined and executed.

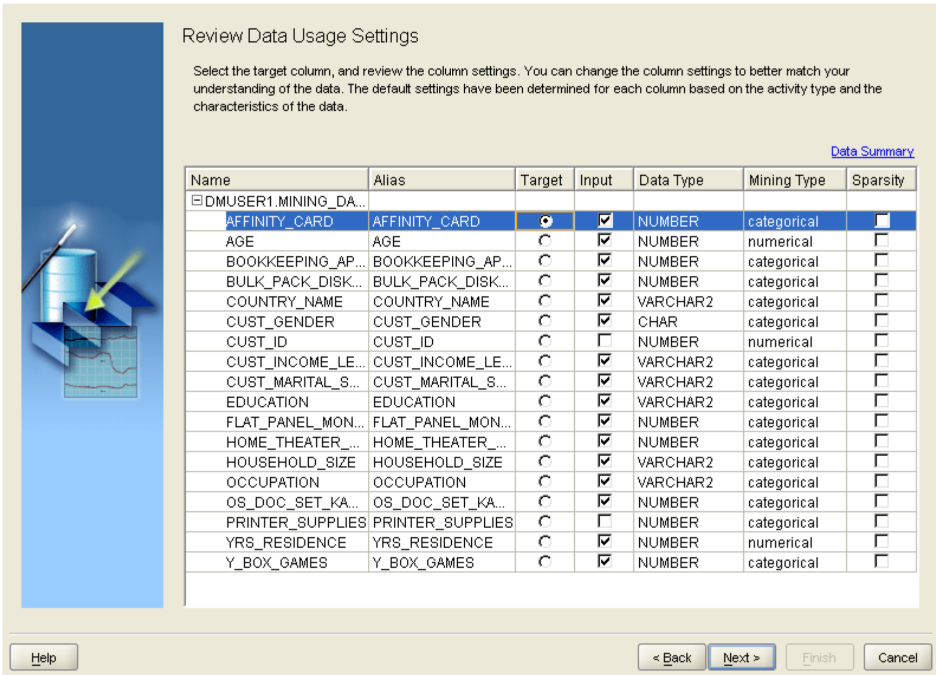
Choose Build from the Activity pull-down menu to launch the activity wizard, and select Attribute Importance as the Function (there is only one choice of algorithm). Click Next.



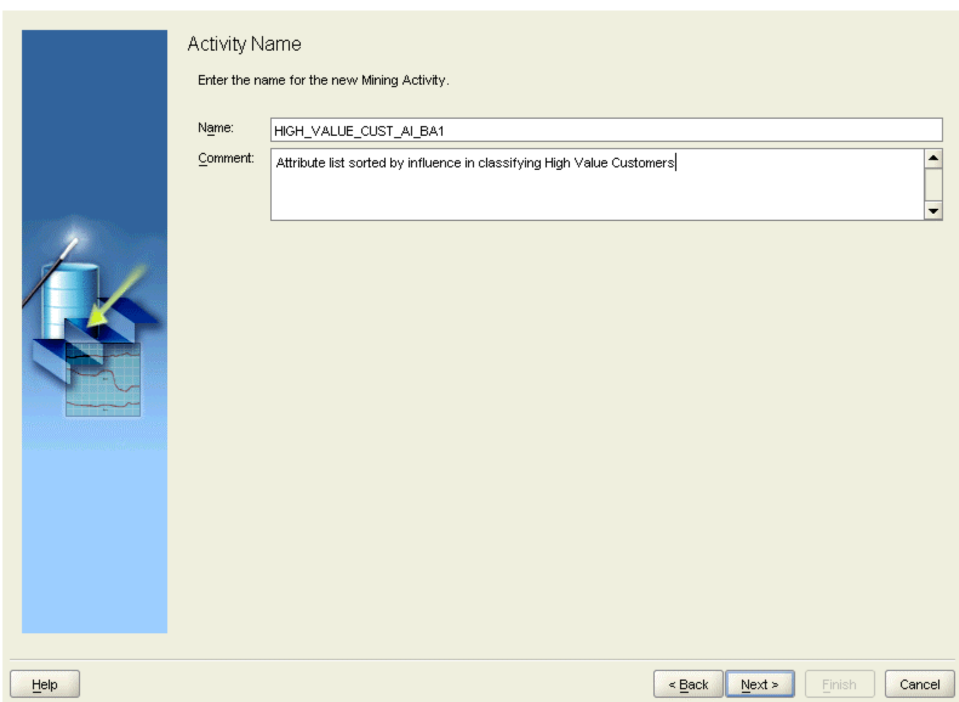
Select MINING_DATA_BUILD_V as the Case table. Select CUST_ID as the Identifier and ensure that the checkbox for additional data is cleared. Click Next.



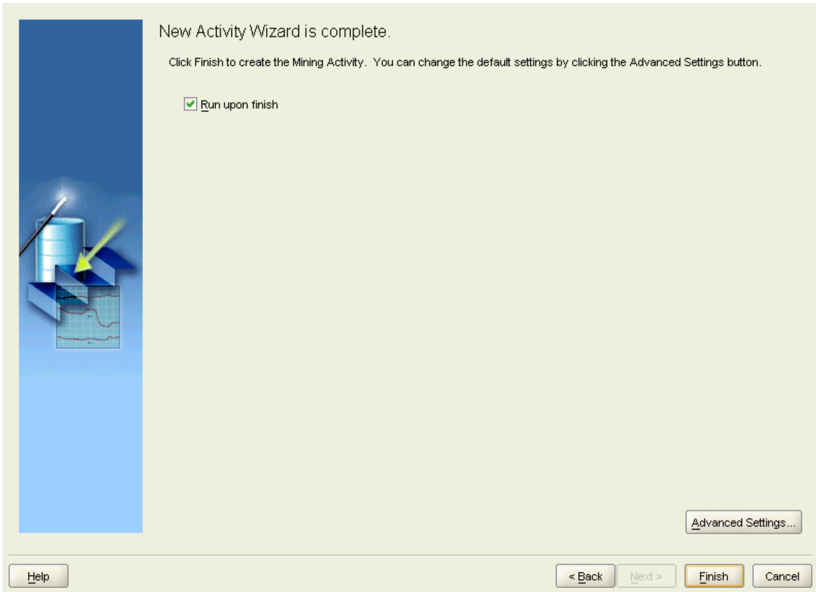
The goal is to distinguish high-value customers from the others. This information is stored in the attribute AFFINITY_CARD (1 = High-value, 0 = Low-value), so click the radio button to specify AFFINITY_CARD as the Target. Click Next.



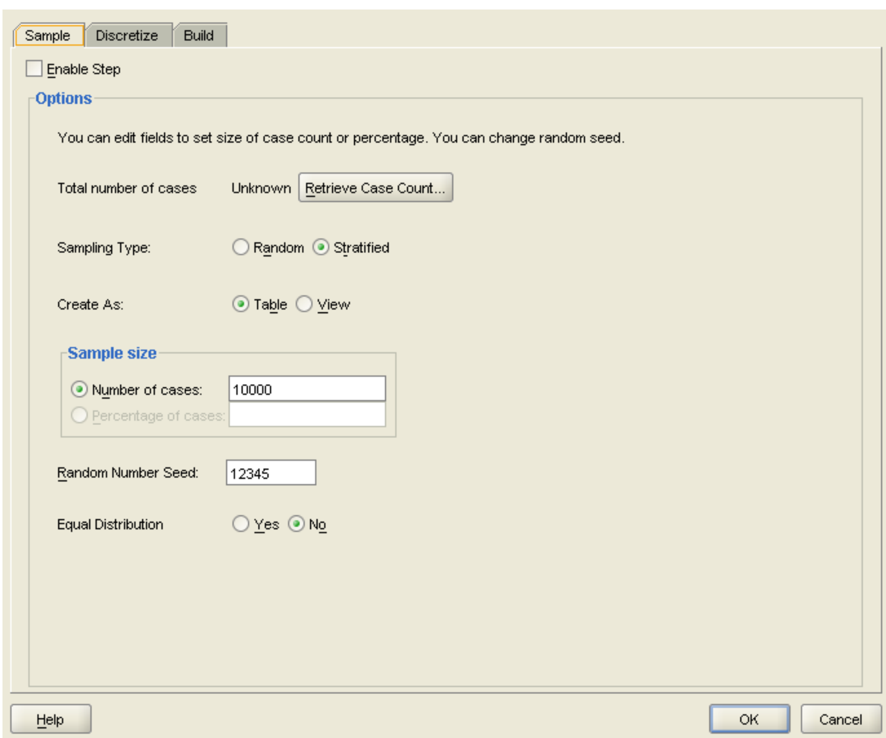
Enter a name for the activity that will clearly identify its purpose. Optionally enter a description in the Comment box and click Next.



You may accept default settings or modify the parameters. Click Advanced Settings to see the user-definable parameters.



There is a tabbed page of settings for each step in the activity, including a checkbox to indicate whether the step should be included in the execution of the activity or skipped. In this example, the activity wizard had determined that the dataset is so small that sampling is not desirable. If you check Enable Step, then you can choose the sample size and the sampling method.

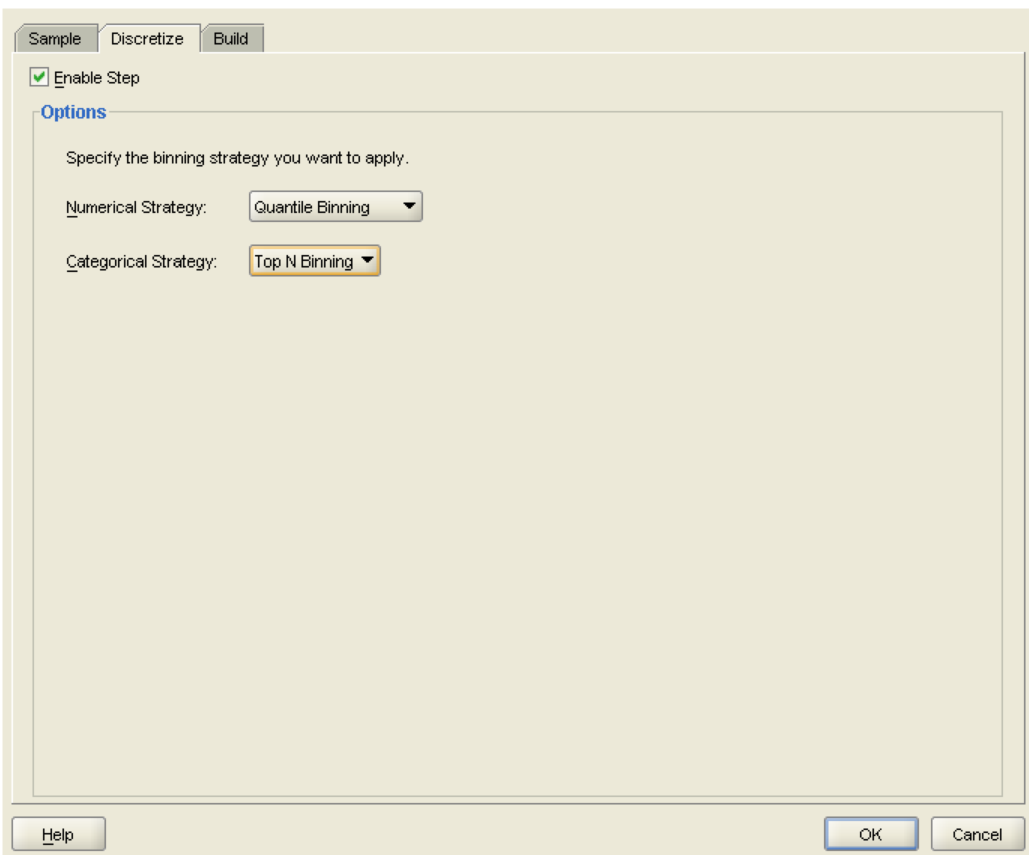


The data will be discretized (that is, binned); numerical data will be binned into ranges of values, and categorical data will be divided into one bin for each of the values with highest distribution (TopN method) and the rest recoded into a bin named “Other”.

The default number of bins is set internally, and depends upon the algorithm and the data characteristics. You can override the defaults on an attribute-by-attribute basis by using the binning wizard (Data → Transform → Discretize) prior to defining an Activity (and turning off Discretization in the Advanced Settings of the activity).

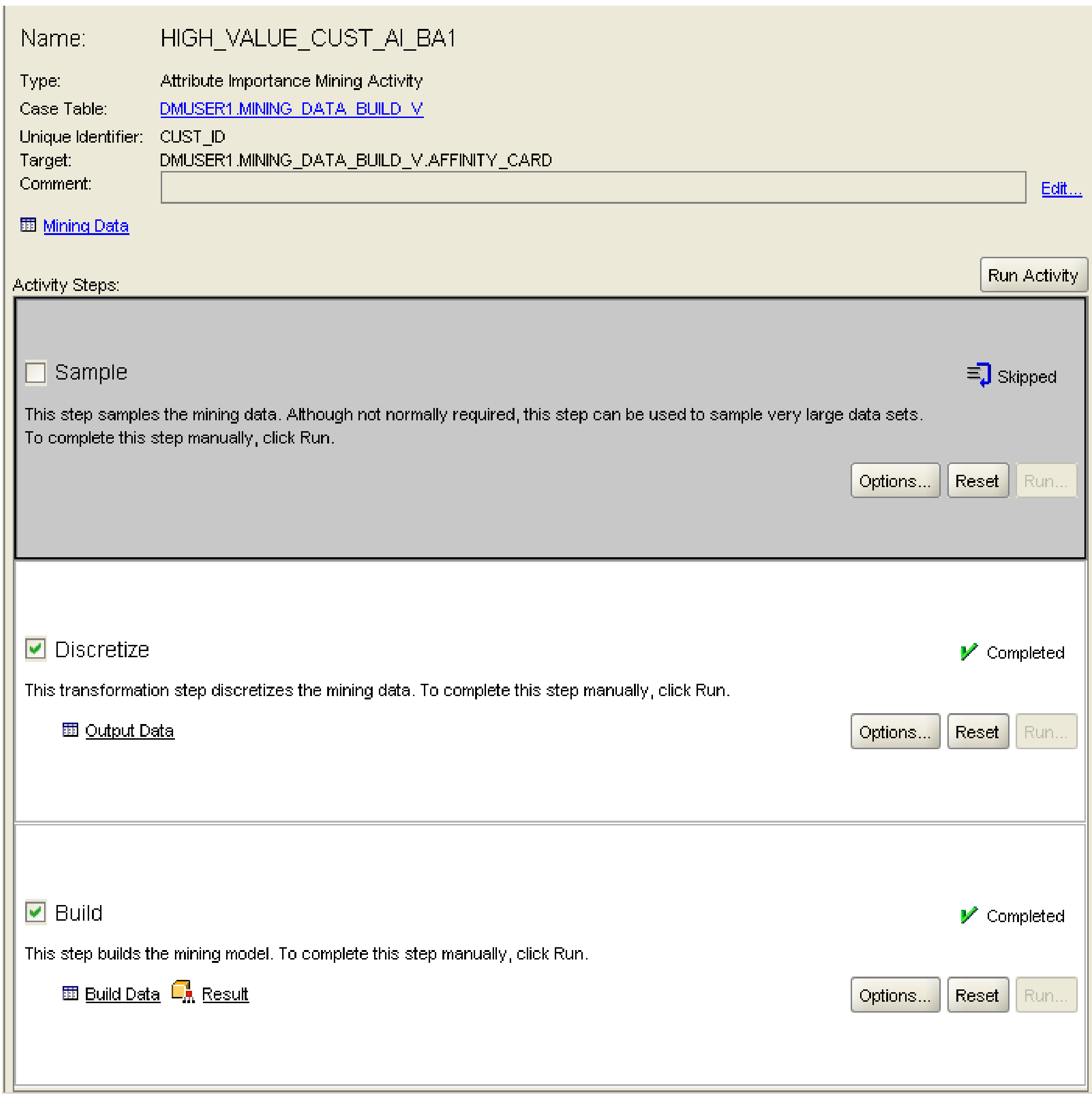
You can change the numerical binning from the default Quantile to the Equi-width method. Categorical binning has only one strategy.

Quantile binning creates bins with approximately equal numbers of cases in each bin, irrespective of the width of the numerical range. Equi-width binning creates bins of identical width, irrespective of the number of cases in each bin – in fact this strategy could generate empty bins.



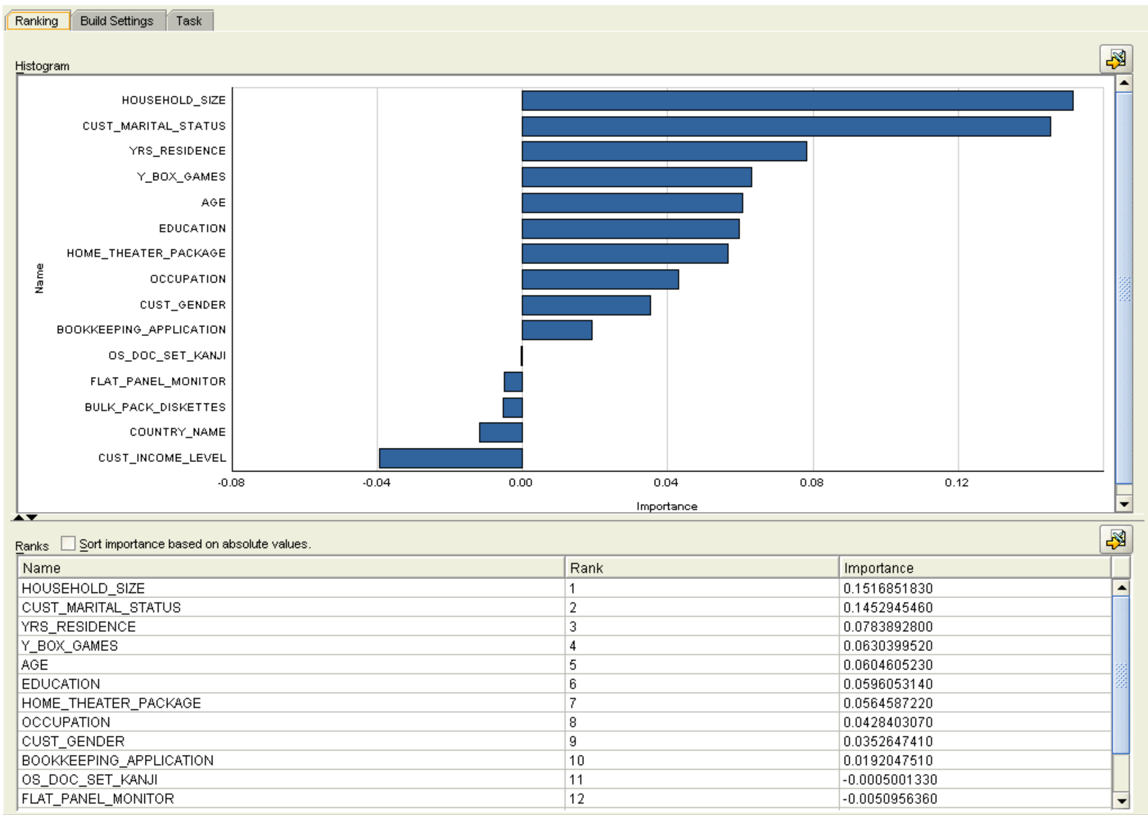
There are no user-defined settings for the Build step of the Attribute Importance algorithm, so click OK to return to the final wizard page, ensure that Run upon Finish is checked, and click Finish.

The steps of the activity are displayed, and an indication appears in each step as the step is either skipped or completed.



When all steps are complete, click Result in the Build step to display the chart and table containing the ranked list of attributes.

This information is useful on its own and also as preparation for building Naïve Bayes and Support Vector Machine models.



Name	Rank	Importance
HOUSEHOLD_SIZE	1	0.1516851830
CUST_MARITAL_STATUS	2	0.1452945460
YRS_RESIDENCE	3	0.0783892800
Y_BOX_GAMES	4	0.0630399520
AGE	5	0.0604605230
EDUCATION	6	0.0596053140
HOME_THEATER_PACKAGE	7	0.0564587220
OCCUPATION	8	0.0428403070
CUST_GENDER	9	0.0352647410
BOOKKEEPING_APPLICATION	10	0.0192047510
OS_DOC_SET_KANJI	11	-0.0005001330
FLAT_PANEL_MONITOR	12	-0.0050956360

Chapter 5 – Classification – Naïve Bayes

A solution to a Classification problem predicts a discrete value for each case: 0 or 1; Yes or No; Low, Medium, or High.

Oracle Data Mining provides four algorithms for solving Classification problems; the nature of the data determines which method will provide the best business solution, so normally you find the best model for each algorithm and pick the best of those for deployment. This chapter discusses the Naïve Bayes algorithm.

Naïve Bayes looks at the historical data and calculates conditional probabilities for the target values by observing the frequency of attribute values and of combinations of attribute values.

For example, suppose A represents "the customer is married" and B represents "the customer increases spending", and you want to determine the likelihood that a married customer will increase spending.

The Bayes theorem states that

Prob(B given A) = Prob(A and B)/Prob(A)

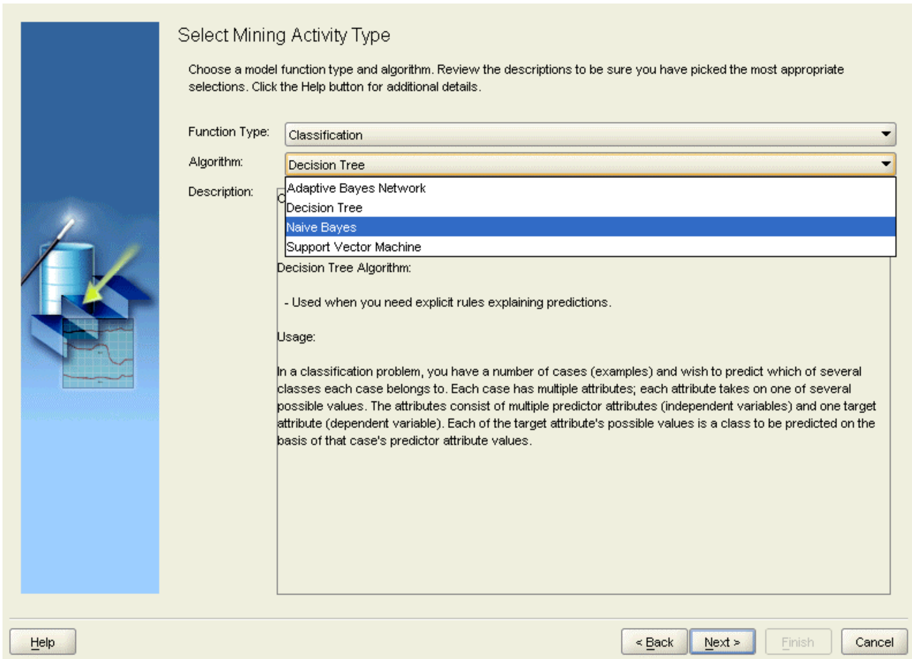
In fact, the formula is made up of many factors similar to this equation because A is usually a complex statement such as "the customer is married AND is between the ages of 25 and 35 AND has 3 or 4 children AND purchased Product Y last year AND … "

So, (keeping to the simple version) to calculate the probability that a customer who is married will increase spending, the algorithm must count the number of cases where A and B occur together as a percentage of all cases ("pairwise" occurrences), and divide that by the number of cases where A occurs as a percentage of all cases ("singleton" occurrences).

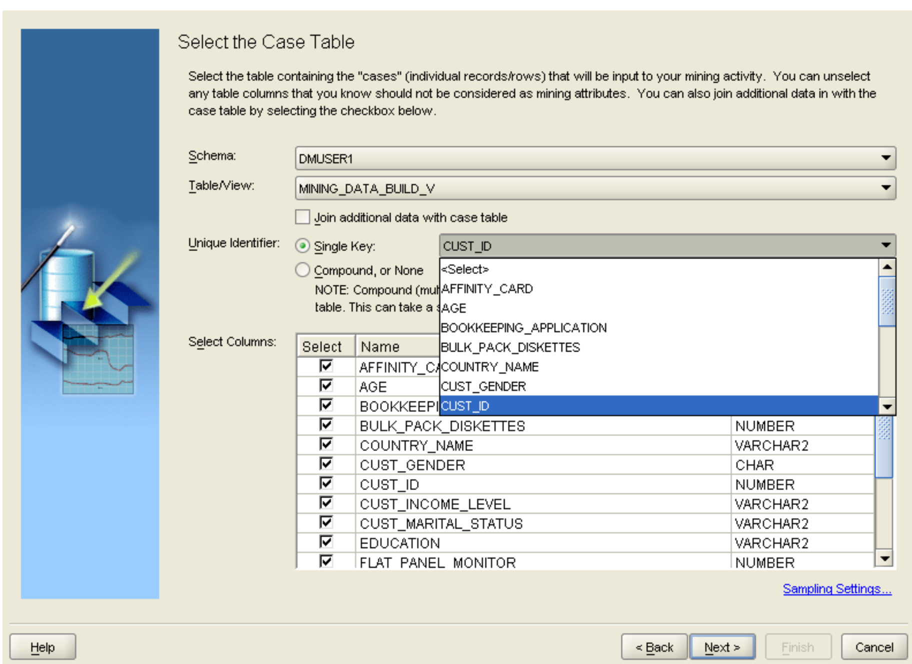
If these percentages are very small, they probably won't contribute to the effectiveness of the model, so for the sake of speed and accuracy, any occurrences below a certain Threshold are ignored.

The Build Activity

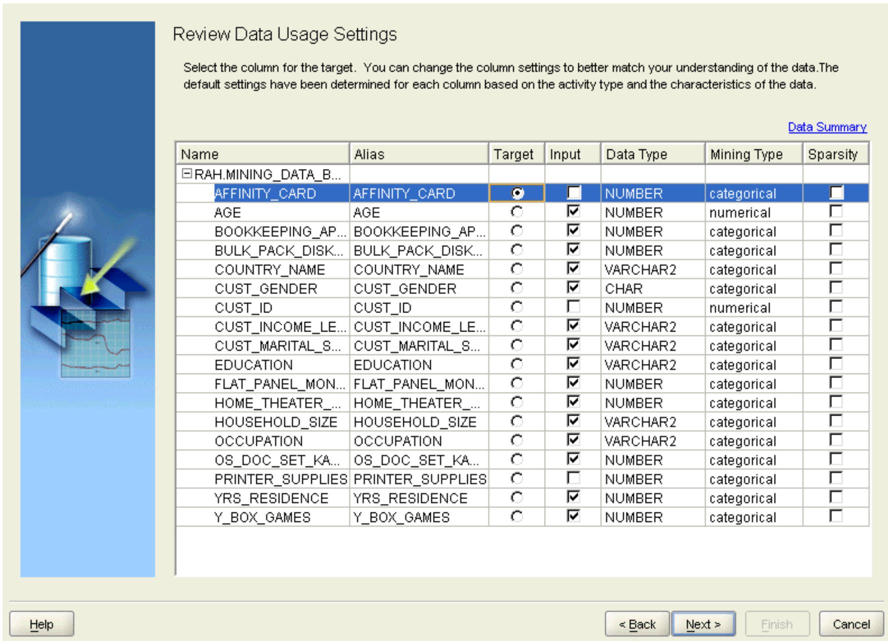
Select Build from the Activity pull-down menu to launch the activity. Select Classification as the Functionality and Naïve Bayes as the algorithm. Click Next.



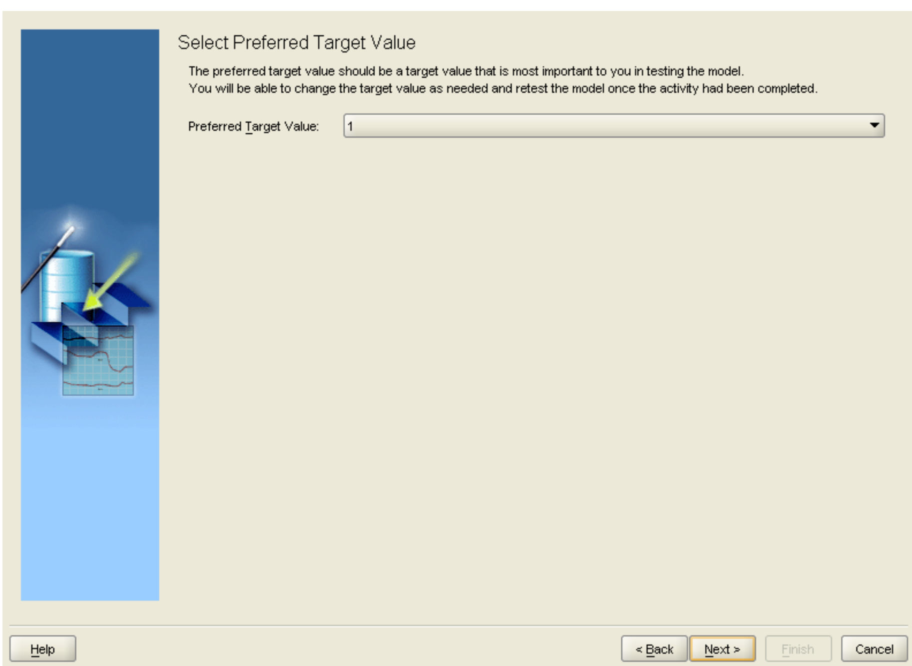
Choose MINING_DATA_BUILD_V as the case table and select CUST_ID as the Identifier. In this example, no additional data will be joined, so ensure that the checkbox is cleared, and click Next.



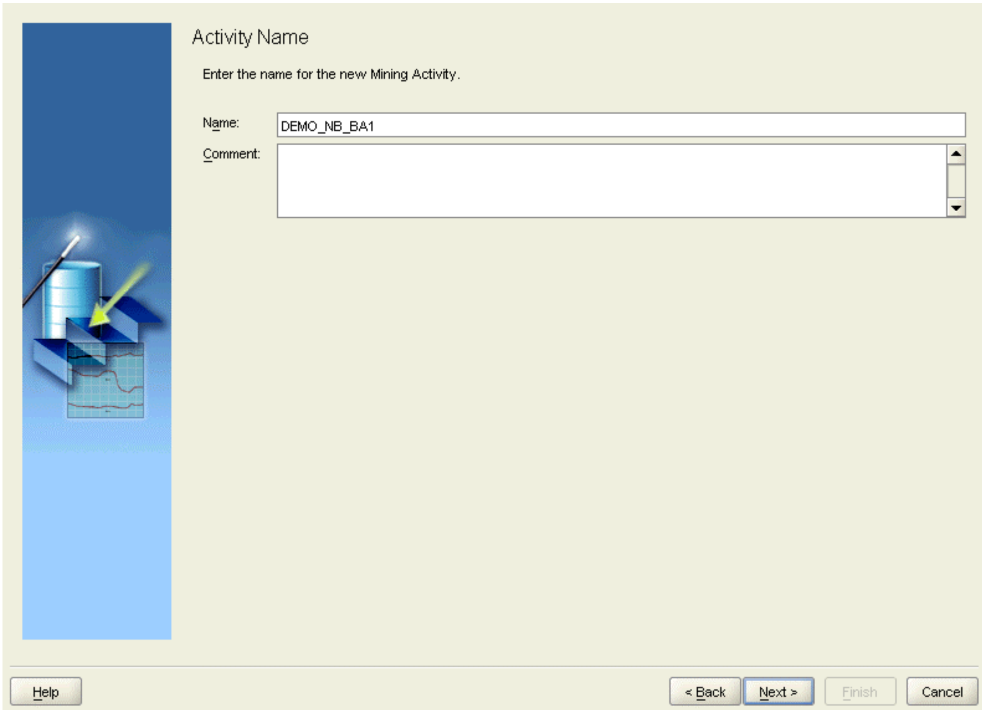
The goal is to distinguish high-value customers from the others. This information is stored in the attribute AFFINITY_CARD (1 = High-value, 0 = Low-value), so click the radio button to specify AFFINITY_CARD as the Target. Click Next.



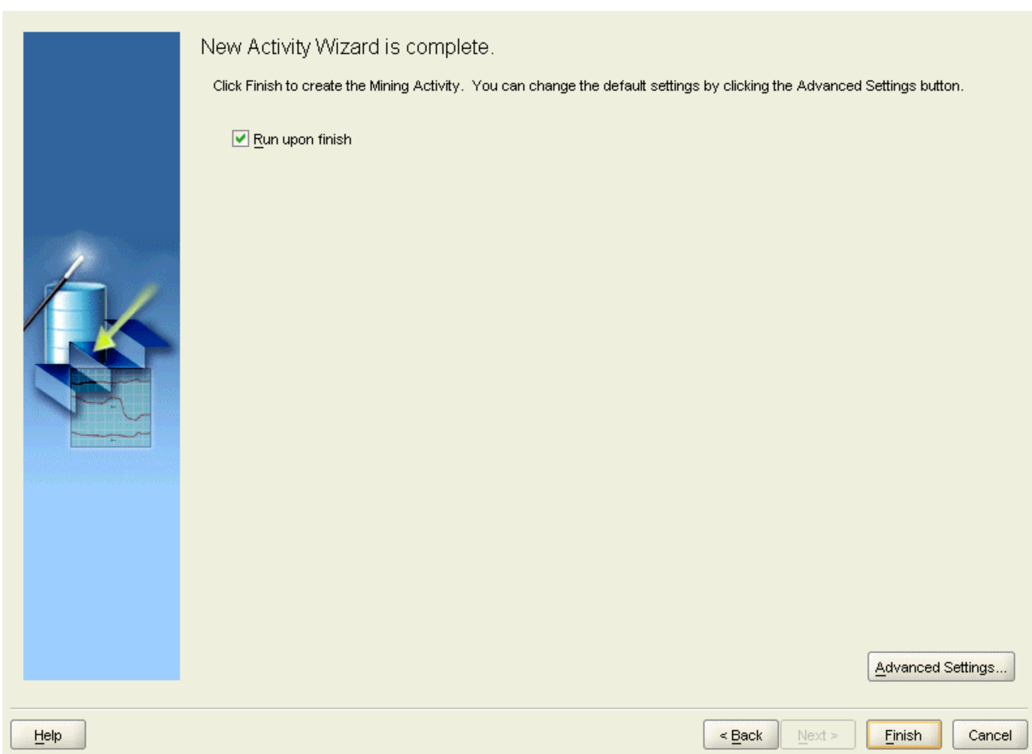
The preferred target value indicates which cases you are trying to identify. In this case, the goal is to find the high-value customers – that is, the cases with AFFINITY_CARD = 1, so select 1 from the pull-down menu and click Next.



Enter a name that explains the activity and click Next.

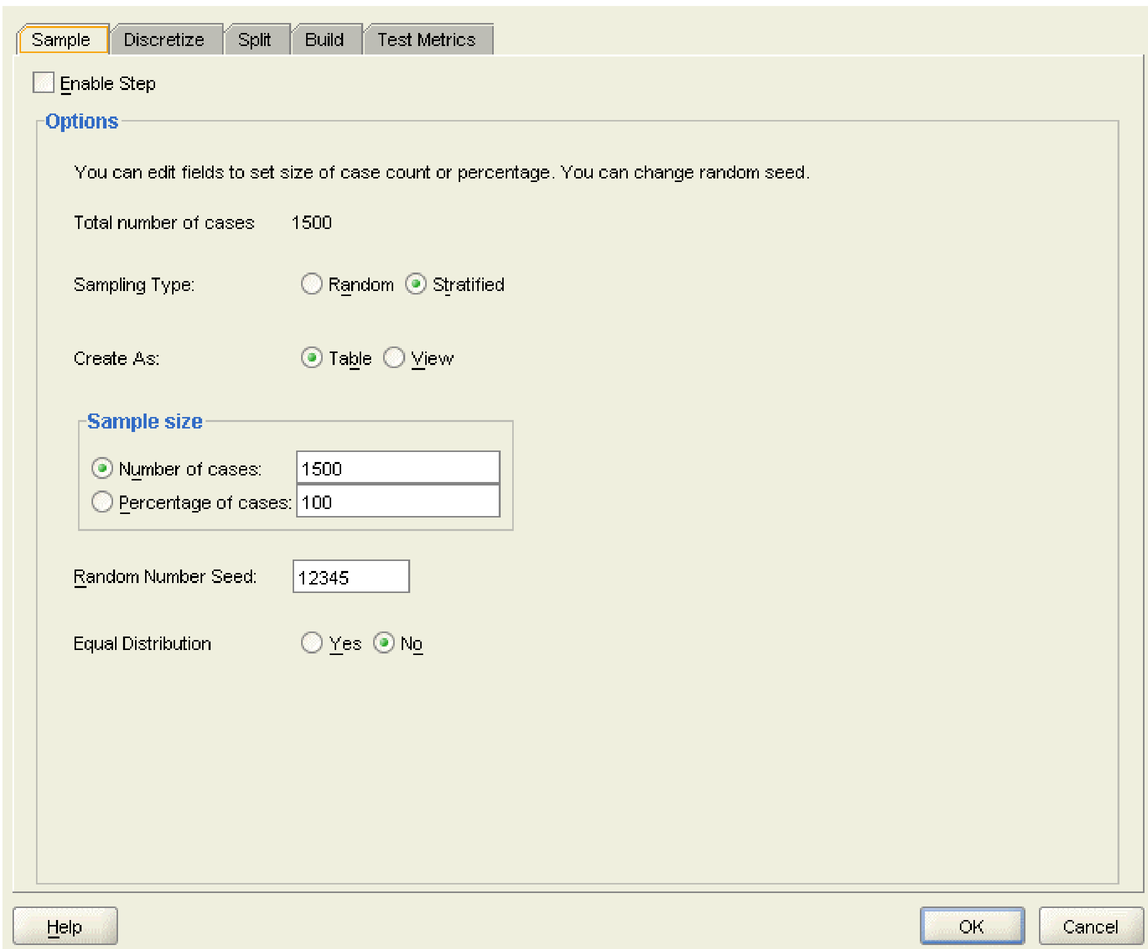


On the final wizard page, click Advanced Settings to display (and possibly modify) the default settings.



In general, the Sample step is not Enabled; Oracle Data Mining scales to any size dataset, but if there are hardware limitations, sampling is desirable.

If Sampling is enabled, you can choose the sample size and the method of sampling. Random sampling chooses the number of specified cases with approximately the same distribution of target values as in the original data. Stratified sampling chooses cases so as to result in data with approximately the same number of cases with each target value. Stratified sampling is valuable in situations where the preferred target value is rare (such as the problem of detecting illegal activity or a rare disease).

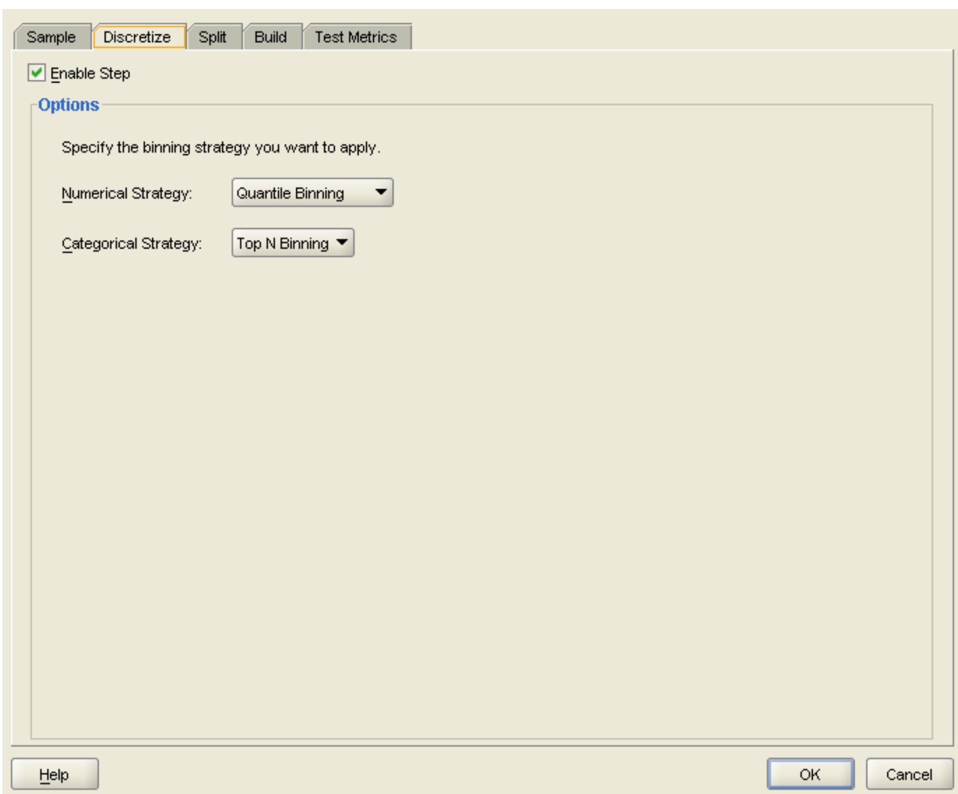


The data will be discretized (that is, binned); numerical data will be binned into ranges of values, and categorical data will be divided into one bin for each of the values with highest distribution (TopN method) and the rest recoded into a bin named “Other”. Each bin is labeled with an integer; Naïve Bayes relies on counting techniques to calculate probabilities, and integers are much easier to count than decimal numbers or character strings.

The default number of bins is set internally, and depends upon the algorithm and the data characteristics. You can override the defaults on an attribute-by-attribute basis by using the binning wizard (Data → Transform → Discretize) prior to defining an Activity (and turning off Discretization in the Advanced Settings of the activity).

You can change the numerical binning from the default Quantile to the Equi-width method. Categorical binning has only one strategy.

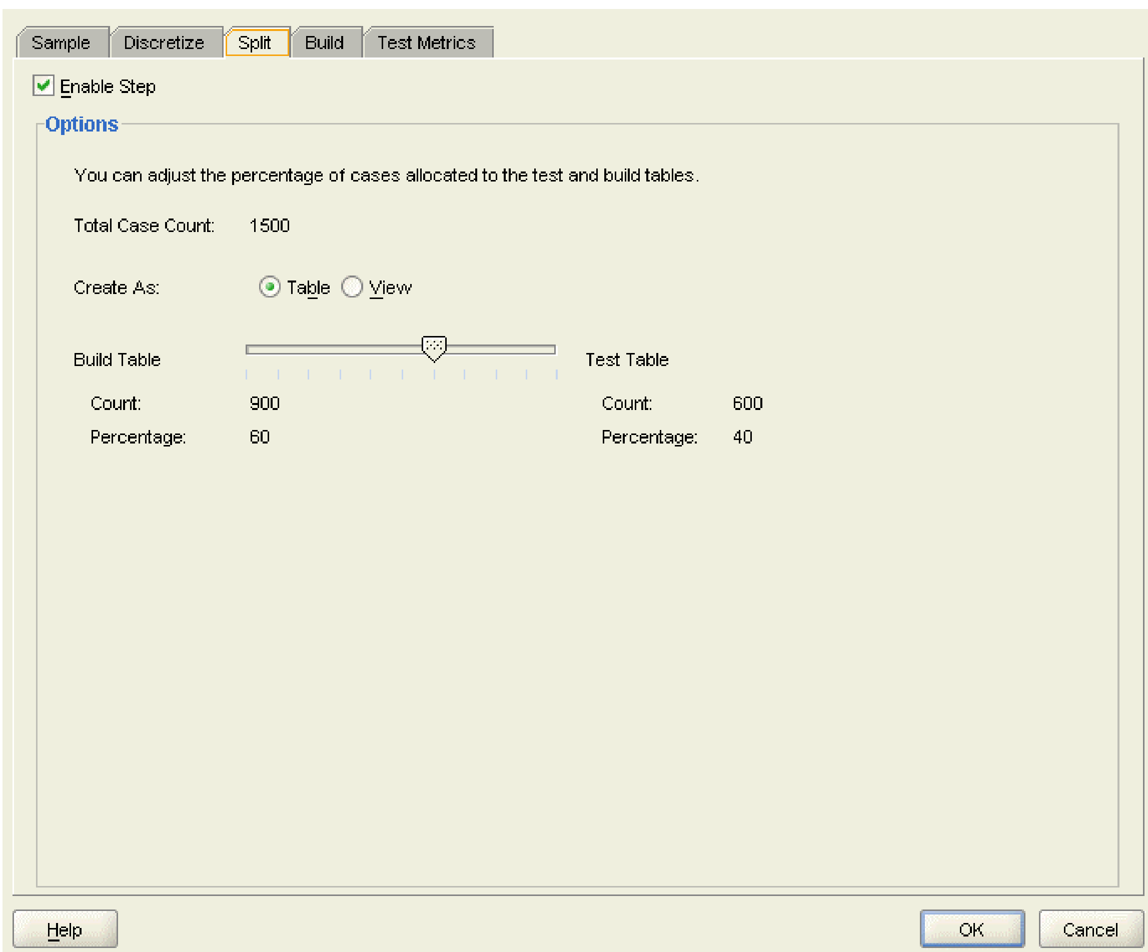
Quantile binning creates bins with approximately equal numbers of cases in each bin, irrespective of the width of the numerical range. Equi-width binning creates bins of identical width, irrespective of the number of cases in each bin – in fact this strategy could generate empty bins.



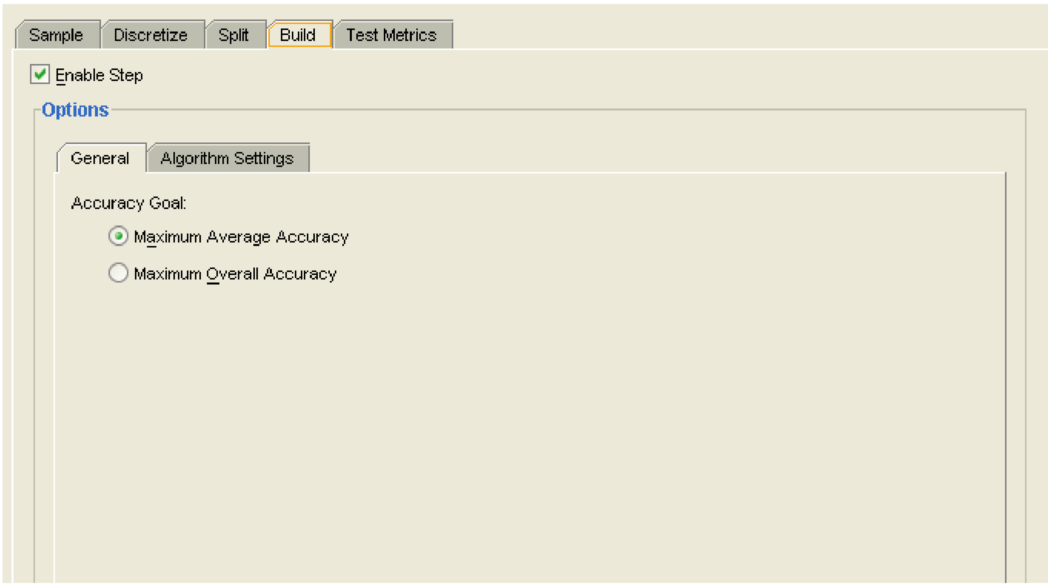
Recall that the source data was derived from a test marketing campaign, and each customer has been assigned the value 0 or 1 in the column AFFINITY_CARD to indicate high-value (1) or low-value (0). It is known what happened in the past; now a model will be built that will learn from the source data how to distinguish between high and low value customers, and will predict what will happen in the future. The model will be applied to all customers to predict who fits the profile of a customer who will produce high revenue. The most interesting cases are the customers who are currently not high-revenue customers, but who are predicted to be likely high-value customers in the future.

In order to test a model, some of the source data will be set aside (called a hold-out sample or a test dataset) to which the model will be applied so the predicted value can be compared to the actual value (the value in the column AFFINITY_CARD) in each case.

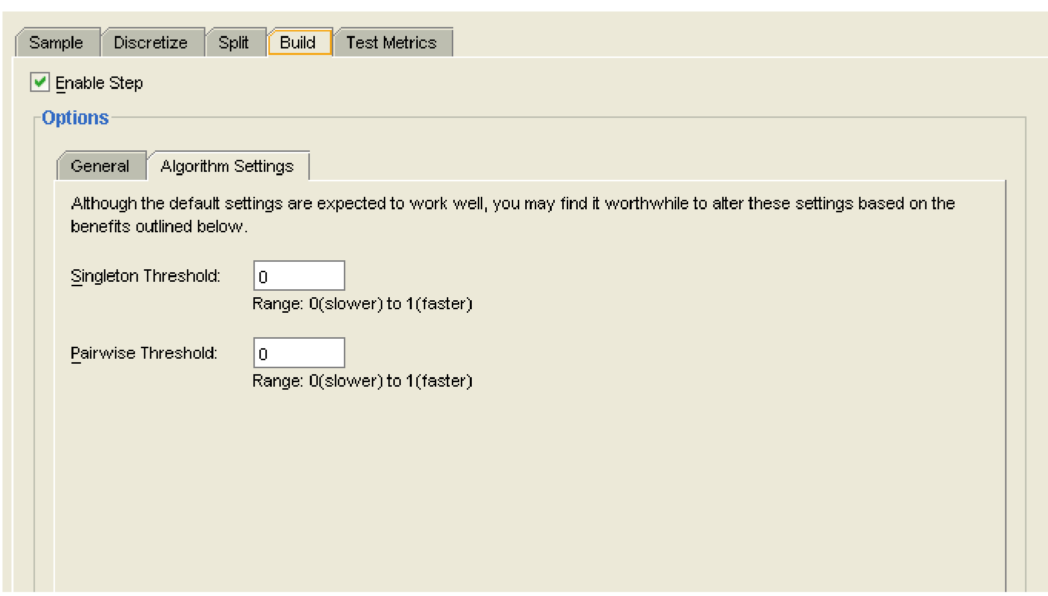
The default split into 60% build data and 40% test data can be modified on this page:



The Build Settings are displayed in tabs. The General Tab allows you to tune the model build towards a model with maximum overall accuracy or a model which is optimally accurate for each Target value. For example, a model may be very good at predicting low-value customers but bad at predicting high-value customers. Typically you want a model that is good at predicting all classes, so Maximum Average Accuracy is the default.



As explained before, Singleton and Pairwise thresholds have to do with eliminating rare and possibly noisy cases. The default threshold is set to 0 in both cases; it may be worthwhile to try raising the thresholds slightly (for example .1, .01) to note the effects.



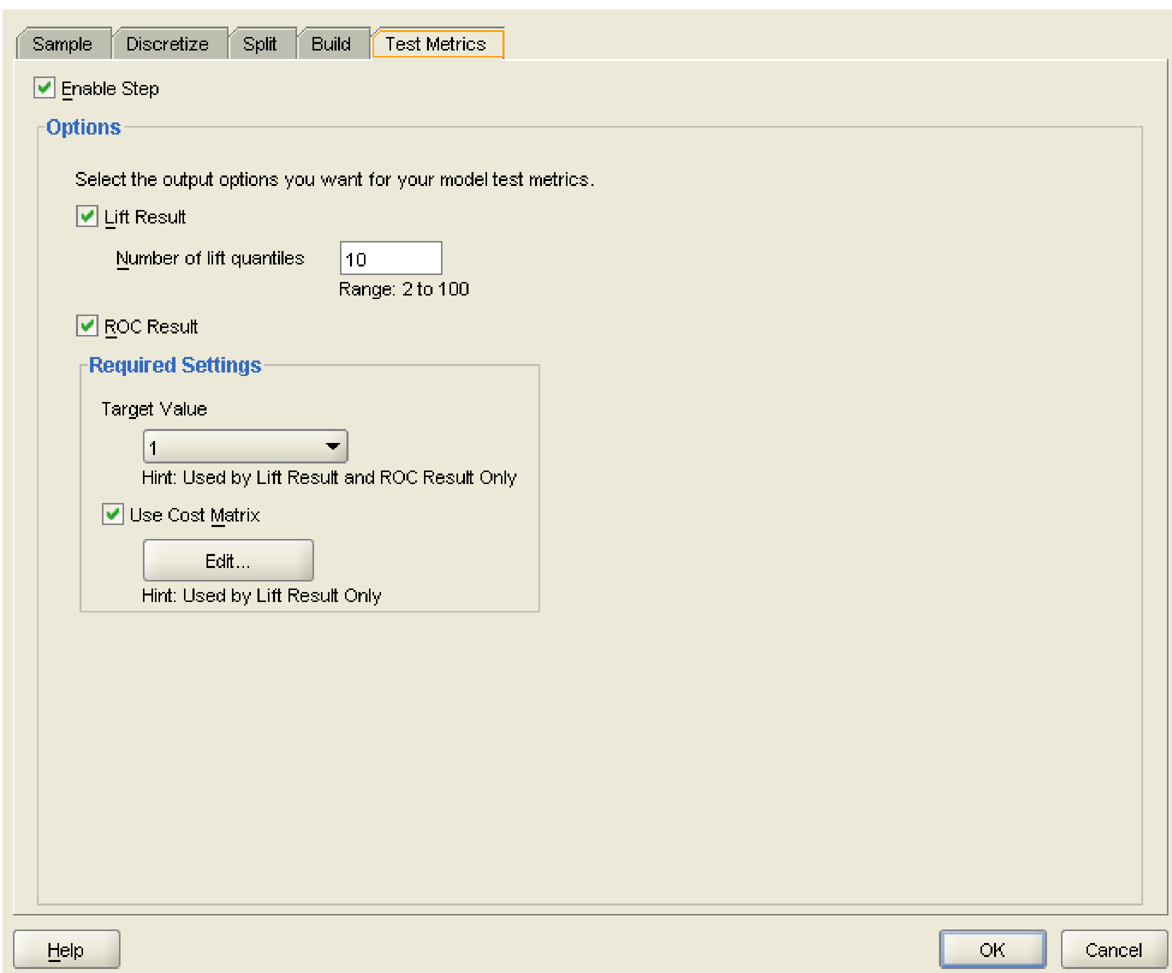
For a Classification problem, all possible test metrics are available.

ROC is a method of experimenting with “what if” analysis – if the probability threshold is changed, how will it affect the model?

The Confusion Matrix indicates the types of errors that the model is likely to make, and is closely tied to the ROC results.

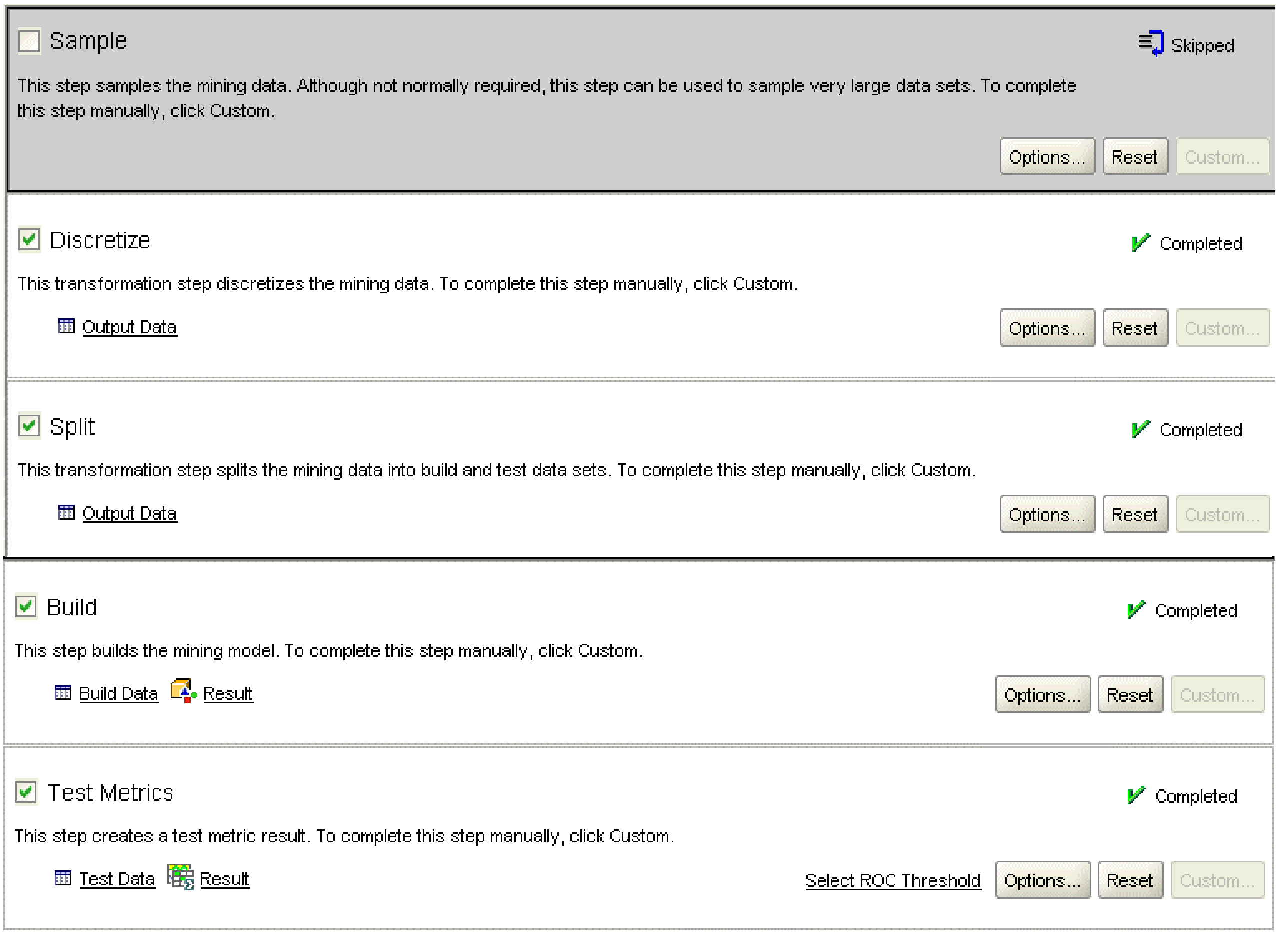
Lift is a different type of model test. It is a measure of how “fast” the model finds the actual positive target values. (The origin is in Marketing: “How much of my Customer database must I solicit to find 50% of the customers likely to buy Product X?”)

These methods will be discussed further when viewing the Test Results.



Click OK to return to the activity wizard. Ensure that Run When Finished is checked and click Finish.

The activity steps are displayed and executed.

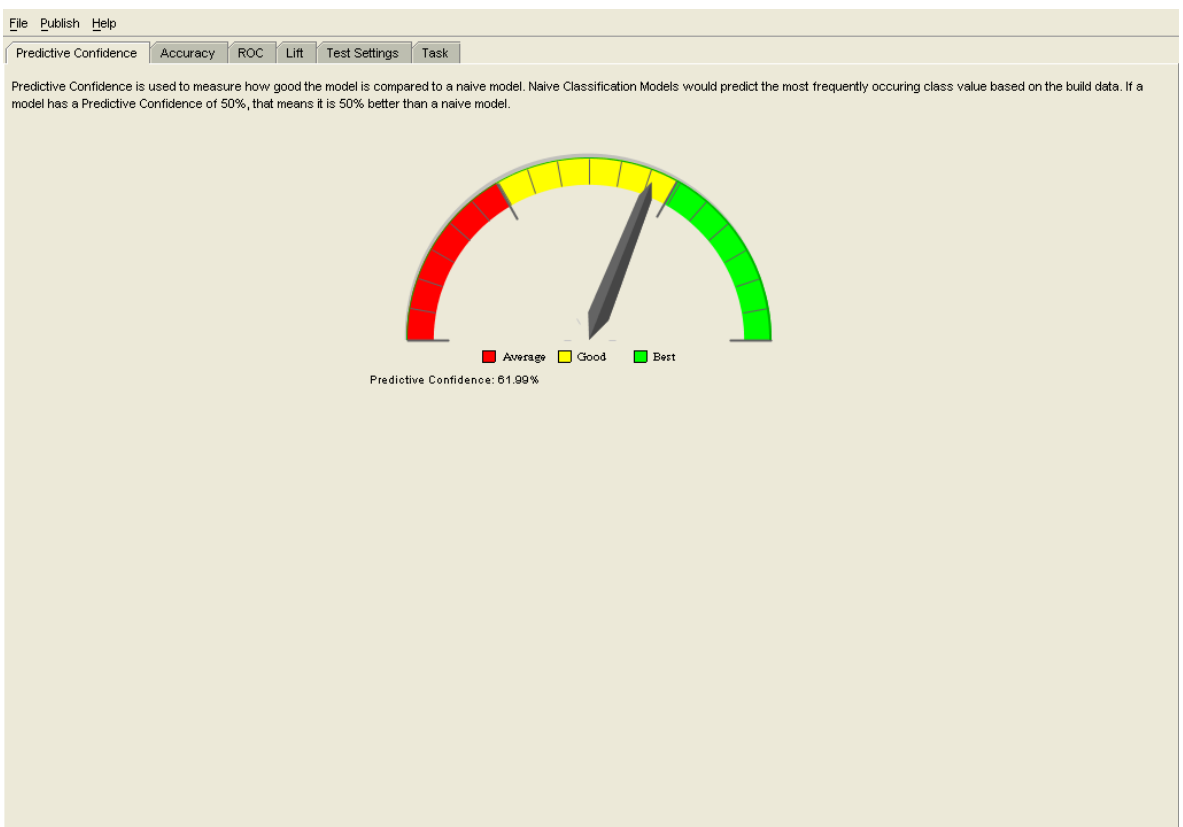


When all steps of the activity are completed, click Result in the Test Metrics step.

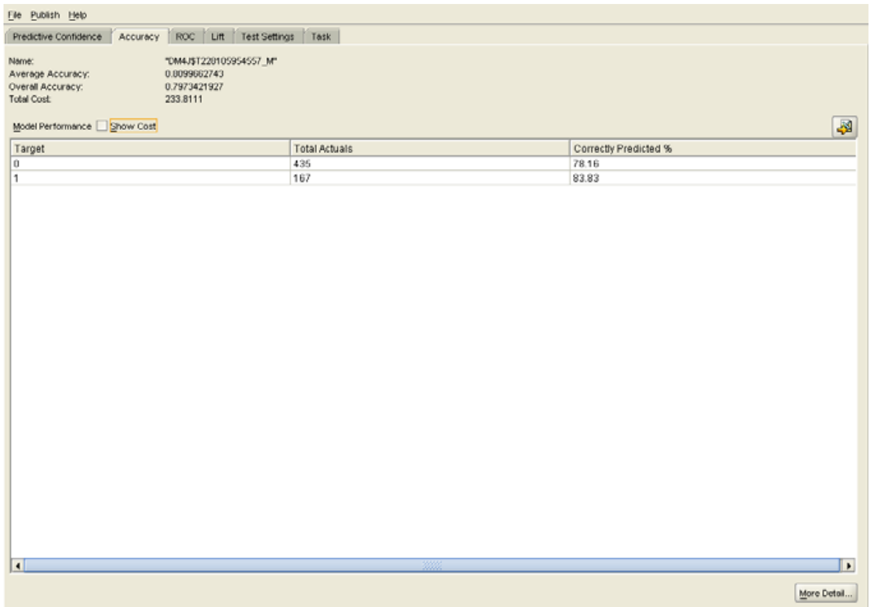
The figures you see may differ slightly from what is shown below, due to the random method of selecting the training data together with the fact that the source dataset is quite small.

The initial page shown is Predictive Confidence, and is a visual indication of the effectiveness of the model compared to a guess based on the distribution of target values in the Build dataset. For example, if the cases in the Build dataset have 40% target value 1 and 60% target value 0, and you are searching for cases with target value 1, you would expect to be successful about 40% of the time when selecting cases at random. However, using the predictive model in this test, you should expect to improve that success rate by about 62%.

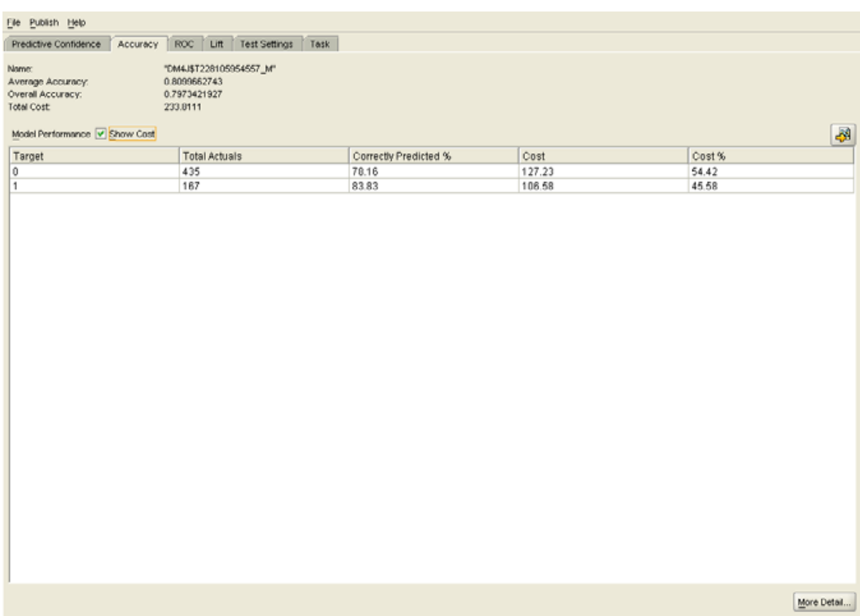
If the needle points to the lowest point on the left of the dial, then the model is no better than a random guess; any other setting indicates some predictive value in the model.



The Accuracy page shows several different interpretations of the model's accuracy when applied to the hold-out sample (the Test dataset). The actual target values are known, so the predictions can be compared to the actual values. The simplest (default) display indicates a class-by-class accuracy – in this example, there are 435 cases with target values of 0, and the model correctly predicted 78.16 % of them. Likewise, the model correctly predicted 83.83 % of the 167 cases of 1.

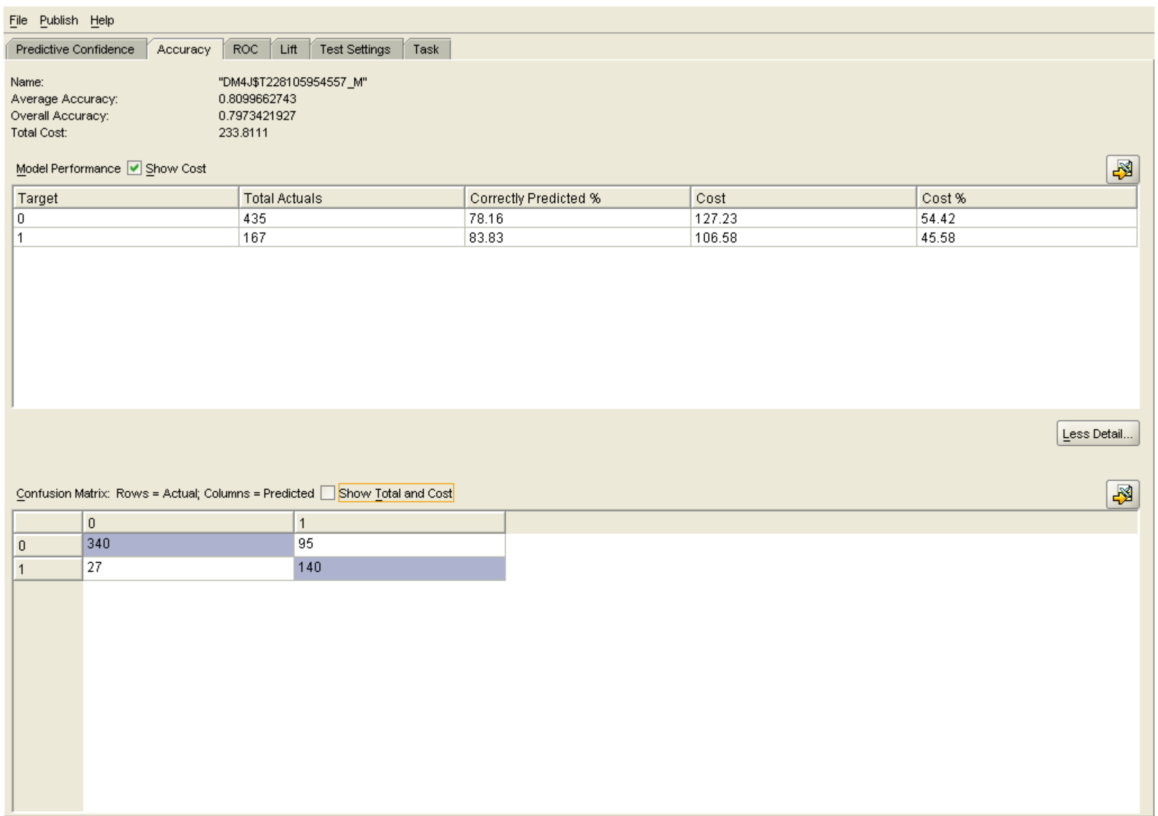


Click the checkbox for Show Cost to see another measure. Cost is an indication of the damage done by an incorrect prediction, and is useful in comparing one model to another. Lower Cost means a better model.

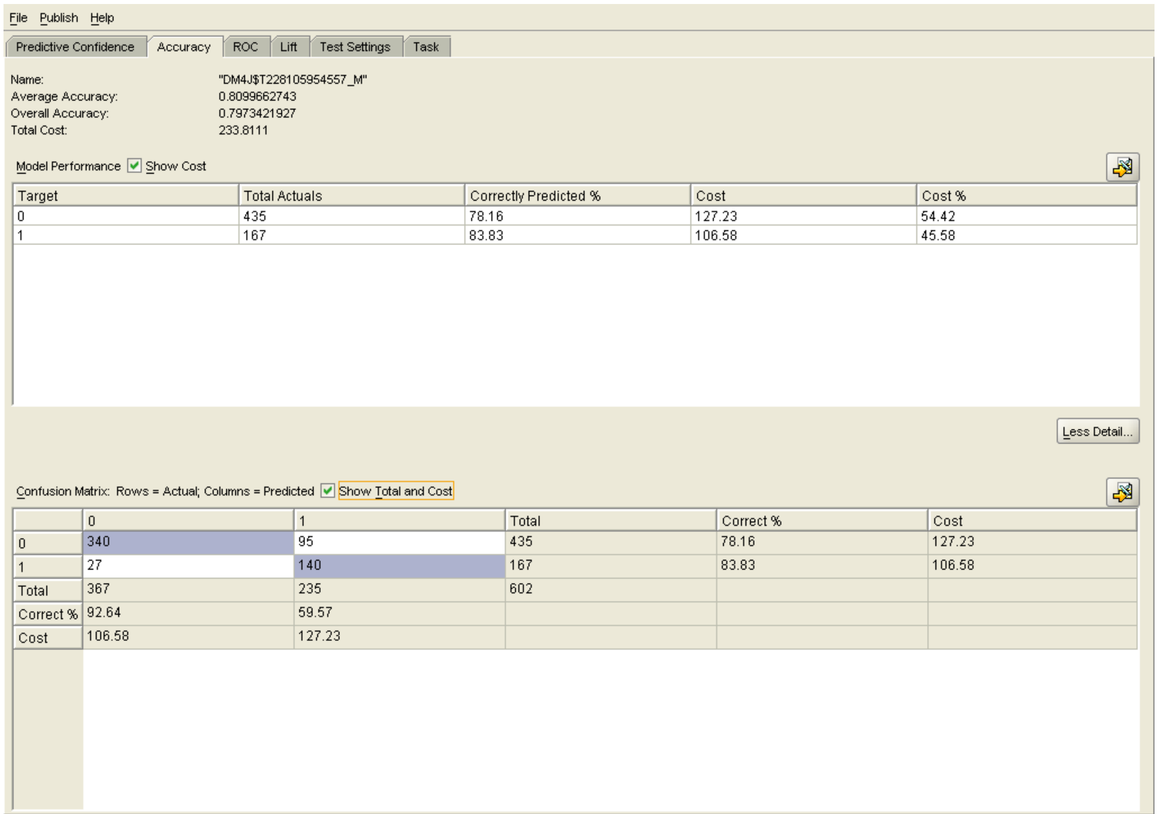


Click the More Detail button to expose the Confusion Matrix, which shows the types of errors that should be expected from this model.

The Confusion Matrix is calculated by applying the model to the hold-out sample from the test campaign. The values of AFFINITY_CARD are known and are represented by the rows; the columns are the predictions made by the classification model. For example, the number 27 in the lower left cell indicates the false-negative predictions – predictions of 0 when the actual value is 1, while the number 95 in the upper right cell indicates false-positive predictions – predictions of 1 when the actual value is 0.



Finally, click the checkbox for Show Total and Cost to display all statistics derived from the Confusion Matrix.

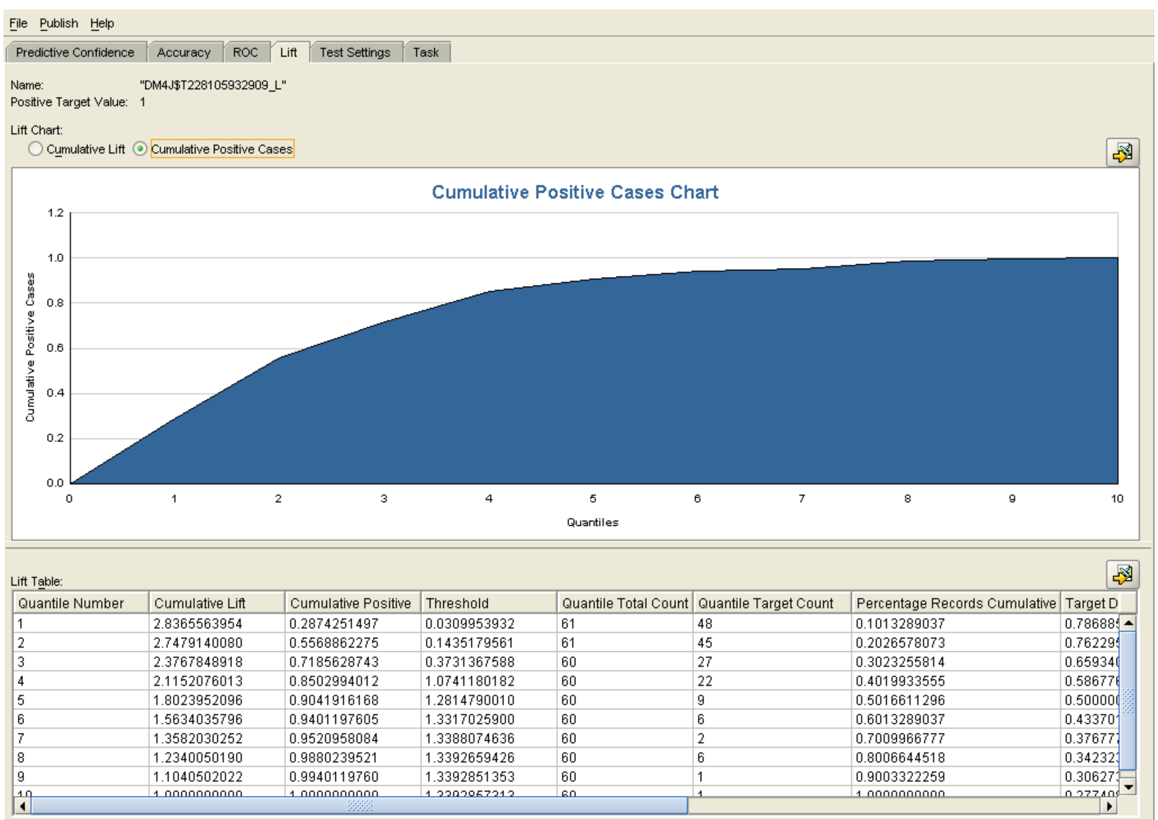


Click the Lift tab to see two graphs showing different interpretations of the lift calculations. The Cumulative Positive Cases Chart is commonly called the Lift Chart or the Gains Chart.

ODM applies the model to test data to gather predicted and actual target values (the same data that was used to calculate the Confusion Matrix), sorts the predicted results by Probability (that is, Confidence in a positive prediction), divides the ranked list into equal parts (quantiles – the default number is 10), and then counts the Actual positive values in each quantile.

This test result indicates the increase in positive responses that will be achieved by marketing to the top percentage of individuals ranked by probability to respond positively, rather than a similar random percentage of the customer base. In this example, the Lift for the top 30% is 2.37, indicating at least twice the response expected compared to marketing to a random 30%. In fact, the next column indicates that over 71% of likely responders are found in the top 3 quantiles.

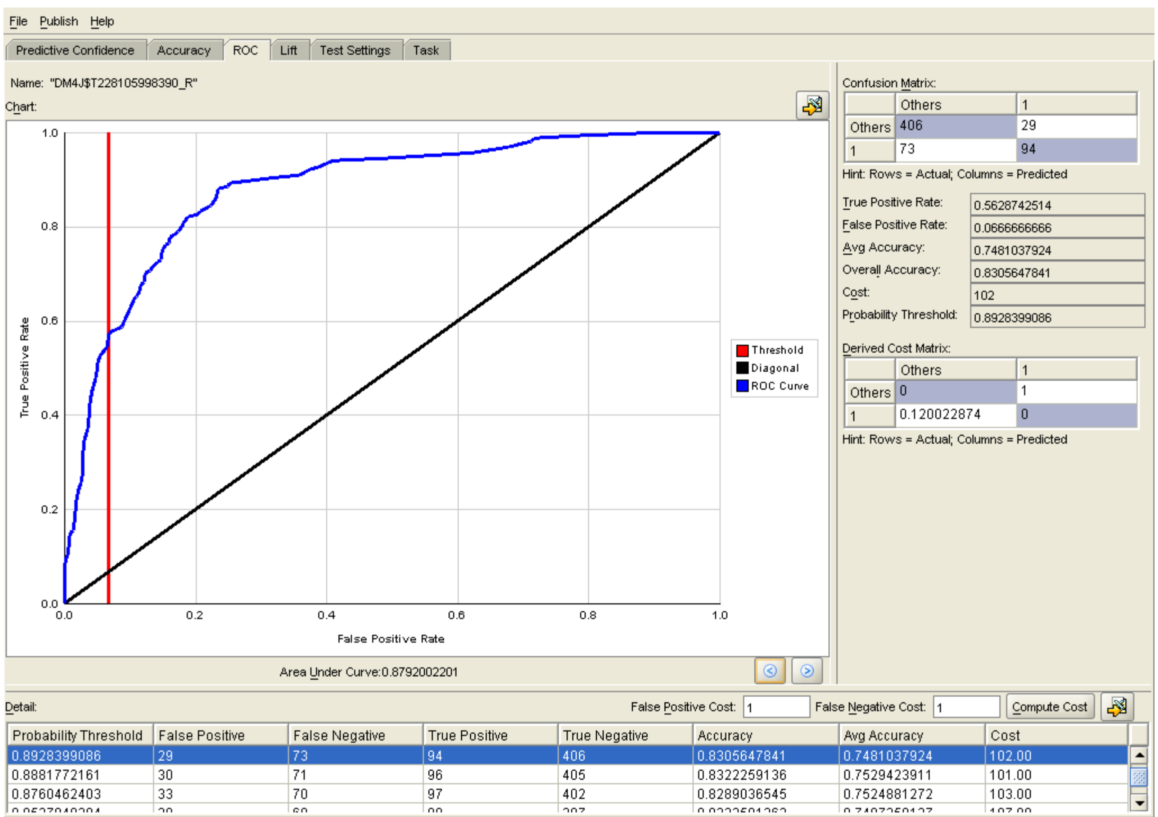
Even though the origin of this test metric is in the area of Marketing, it is a valuable measure of the efficiency of any model.



Quantile Number	Cumulative Lift	Cumulative Positive	Threshold	Quantile Total Count	Quantile Target Count	Percentage Records Cumulative	Target D
1	2.8365563954	0.2874251497	0.0309953932	61	48	0.1013289037	0.786885
2	2.7479140080	0.5568862275	0.1435179561	61	45	0.2026578073	0.762295
3	2.3767848918	0.7185628743	0.3731367588	60	27	0.3023255814	0.659340
4	2.1152076013	0.8502994012	1.0741180182	60	22	0.4019933555	0.586776
5	1.8023952096	0.9041916168	1.2814790010	60	9	0.5016611296	0.500000
6	1.5634035796	0.9401197605	1.3317025900	60	6	0.6013289037	0.433701
7	1.3582030252	0.9520958084	1.3388074636	60	2	0.7009966777	0.376777
8	1.2340050190	0.9880239521	1.3392659426	60	6	0.8006644518	0.342323
9	1.1040502022	0.9940119760	1.3392851353	60	1	0.9003322259	0.306273
10	1.0000000000	1.0000000000	1.3392857313	60	1	1.0000000000	0.277409

Click the ROC tab to explore possible changes in the model's parameters.

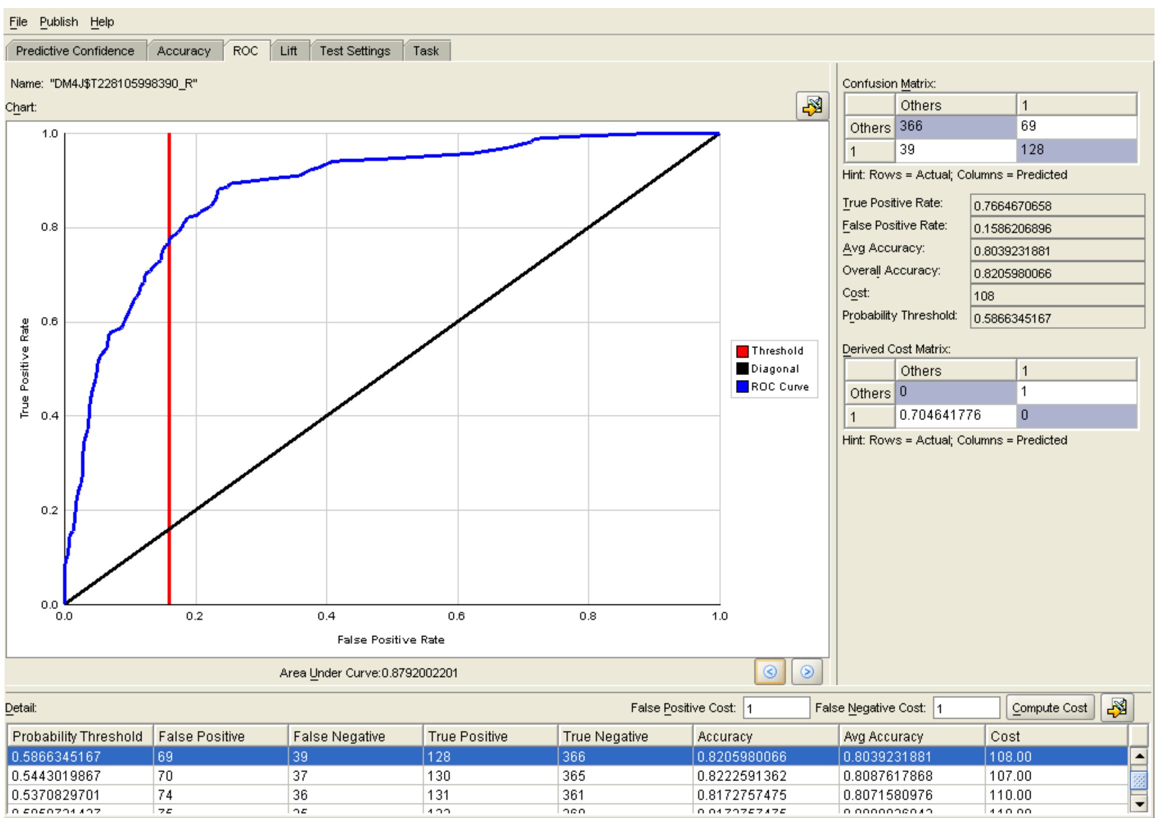
The ROC metric gives the opportunity to explore "what-if" analysis. You can experiment with modified model settings to observe the effect on the Confusion Matrix. For example, suppose the business problem requires that the false-negative value (in this example 73) be reduced as much as possible within the confines of a business requirement of a maximum of 200 positive predictions. It may be that you will offer an incentive to each customer predicted to be high-value, but you are constrained by budget to a maximum of 200 incentives. On the other hand, the 73 false negatives represents "missed opportunities", so you want to avoid such mistakes.



Probability Threshold	False Positive	False Negative	True Positive	True Negative	Accuracy	Avg Accuracy	Cost
0.8928399086	29	73	94	406	0.8305647841	0.7481037924	102.00
0.8881772161	30	71	96	405	0.8322259136	0.7529423911	101.00
0.8760462403	33	70	97	402	0.8289036545	0.7524881272	103.00

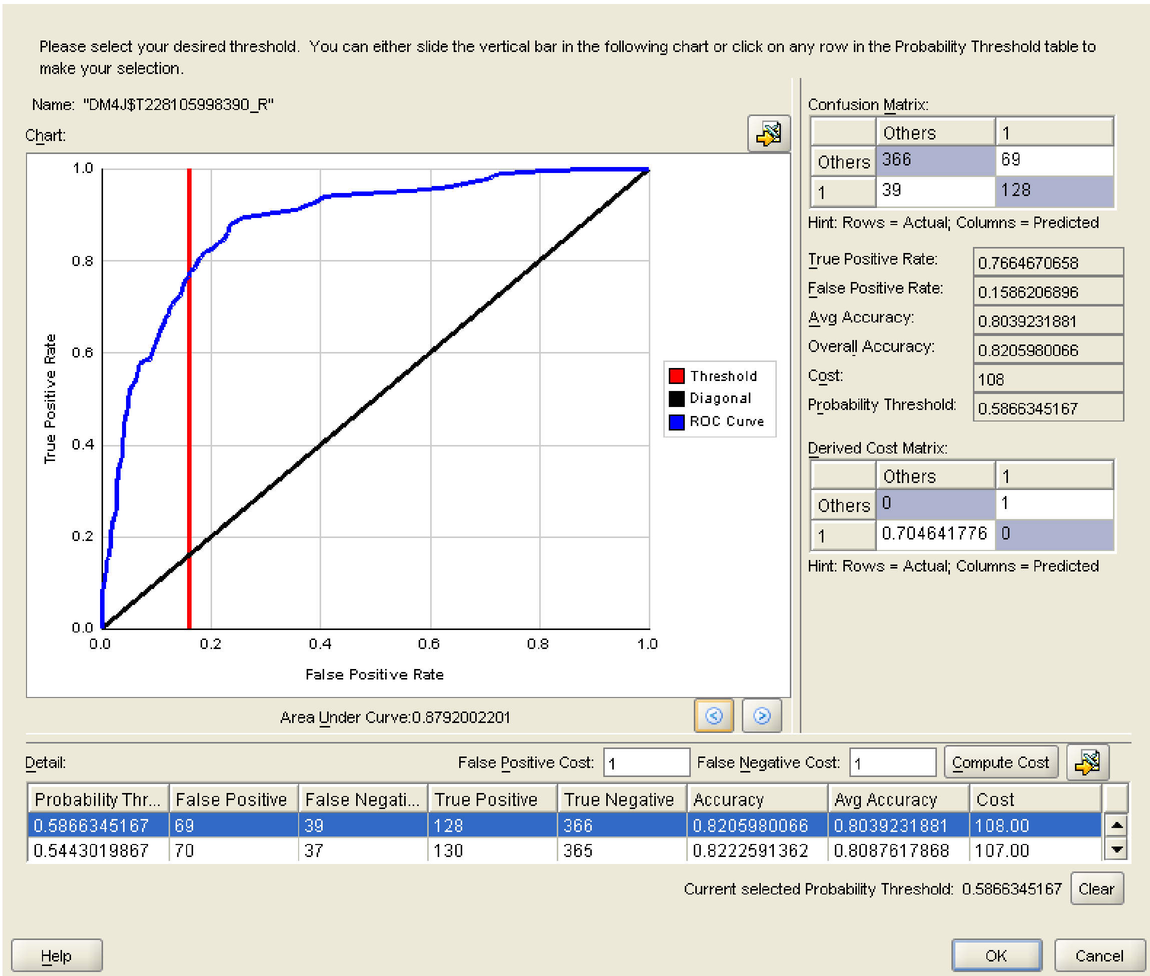
Move the red vertical line (either by clicking the arrows at the lower right below the graph, or by highlighting a row of the Detail chart in the bottom half of the page) and observe the changes in the Confusion Matrix. The example shows that the false negatives can be reduced to 39 while keeping the total positive predictions under 200 (69 + 128 = 197).

The underlying mathematics is that the Cost Matrix, used in making the prediction, is being modified, resulting in a probability threshold different from .5. Normally, the probability assigned to each case is examined and if the probability is .5 or above, a positive prediction is made. Changing the Cost matrix changes the “positive prediction” threshold to some value other that .5, and it is highlighted in the first column of the table beneath the graph.

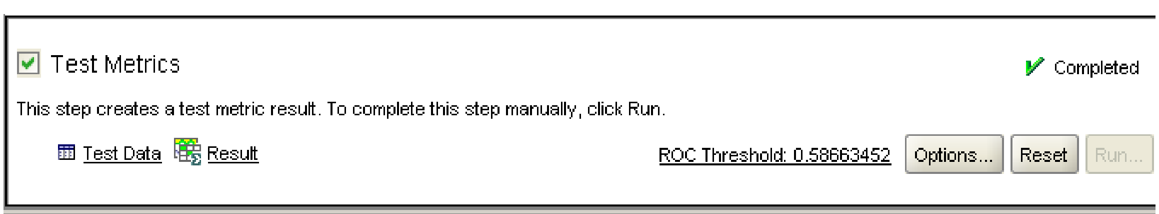


This page is experimental in nature; to make a permanent change to the actual model, return to the Activity display and click Select ROC Threshold to open a Threshold Selection Dialog window.

Now highlight the row containing the threshold determined by experimentation, then click OK. The model is now modified.



The modified threshold is now displayed in the Test Metrics step of the activity.

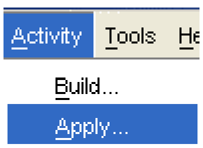


The Apply Activity

Suppose that after further experimentation, the Naïve Bayes model built in the previous section is determined to be the best solution to the business problem. Then the model is ready to be applied to the general population, or to new data. This is sometimes referred to as “scoring the data”.

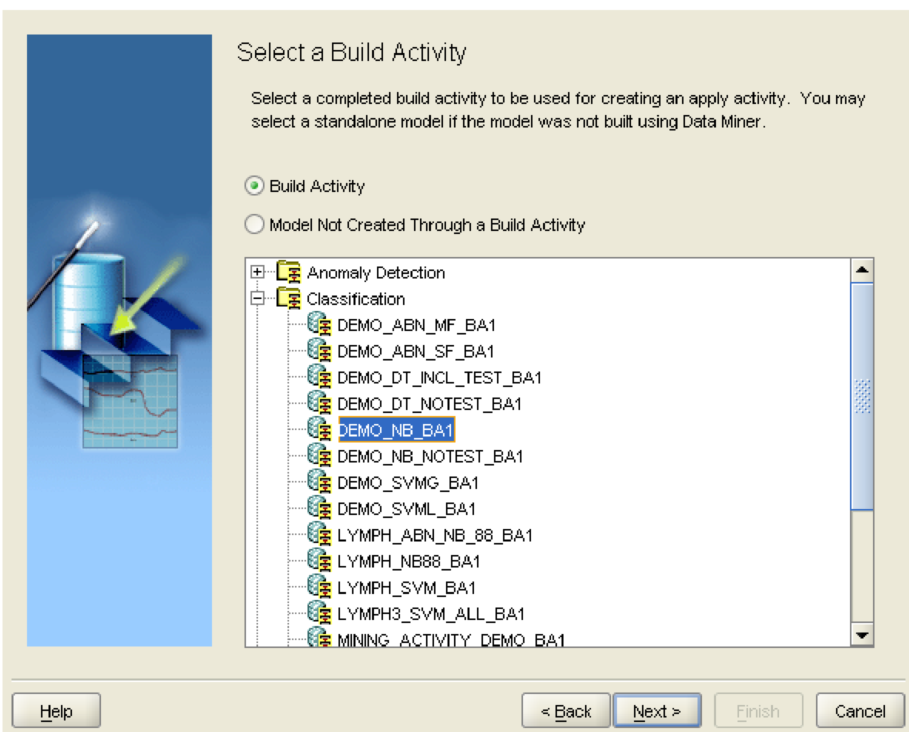
In this example, a dataset named MINING_DATA_APPLY_V will represent the new data to be scored by the model.

Launch the Activity Guide Apply wizard from the Activity menu

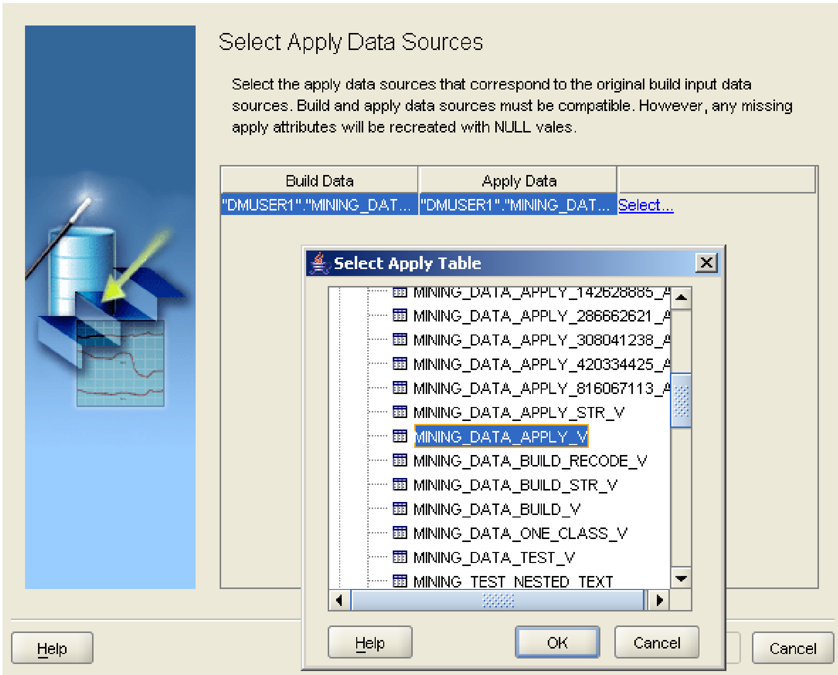


When a model is applied to new data, the input data must be prepared and transformed in exactly the same way that the source data for the Build activity was prepared. As noted on the Welcome screen, the Apply activity is based on a Build activity, and the Build activity will pass to the Apply activity whatever knowledge is required to prepare the input data appropriately. Click Next.

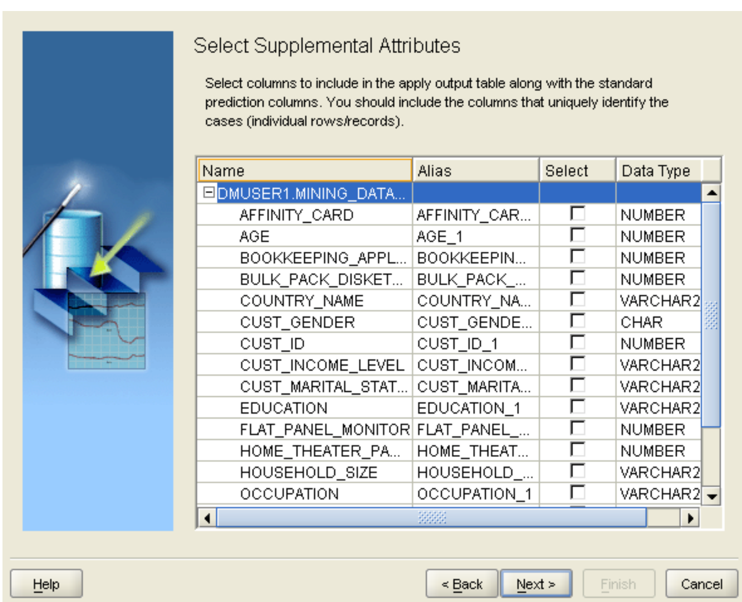
Select the Classification Build Activity that was used to create the model, and all the information about data preparation and model metadata will be passed to the apply activity. Click Next.



Click Select, expand the schema containing your data, and highlight the input data for the Apply Activity. Click OK then Next.



In the Select Supplemental Attributes page, you may click the Select box to join in additional columns to be included in the table holding the result of the Apply operation. By default, the Apply Result contains only the case identifier and the prediction information; if information such as name, address, or other contact information is contained in the source data, it can be included now. However, it is often more convenient to produce the "bare bones" Apply output table, then join in additional data in a separate operation. Click Next to proceed.



You have a choice of formats for the output table.

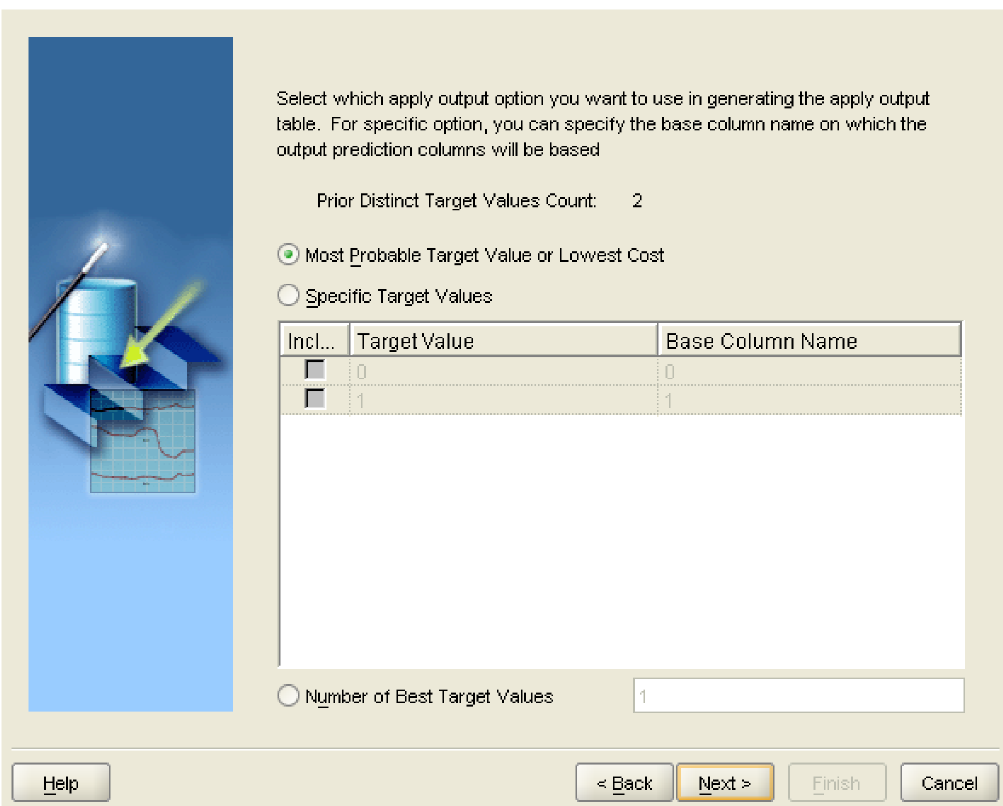
When the model is applied to a particular case, a score (normally a probability) is generated for each possible target value, producing a sorted list of values starting with the most likely value and going down to the least likely value. This list has only two entries if the target is binary, but is longer for multi-class problems (for example, which of seven cars is a person most likely to buy).

In the example of ranking seven cars, you may want to know only the top three choices for each person; in that case, click the radio button next to Number of Best Target Values and enter 3 in the window. The output table will have three rows for each individual containing the prediction information for the top 3 cars.

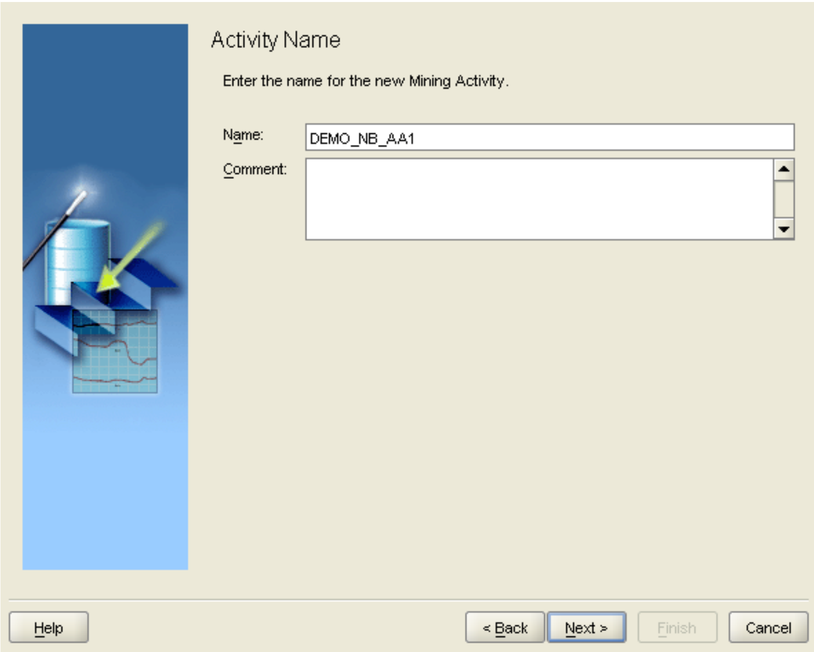
You may want to know each person's score for a particular car; in that case, click the radio button next to Specific Target Values and check the box next to the desired target value. The output table will have one row for each individual containing the prediction information for that one target class, even if it is very unlikely.

You may want to know the most likely target value for each individual. Click the radio button next to Most Probable Target Value or Lowest Cost, and the output table will have one row for each individual.

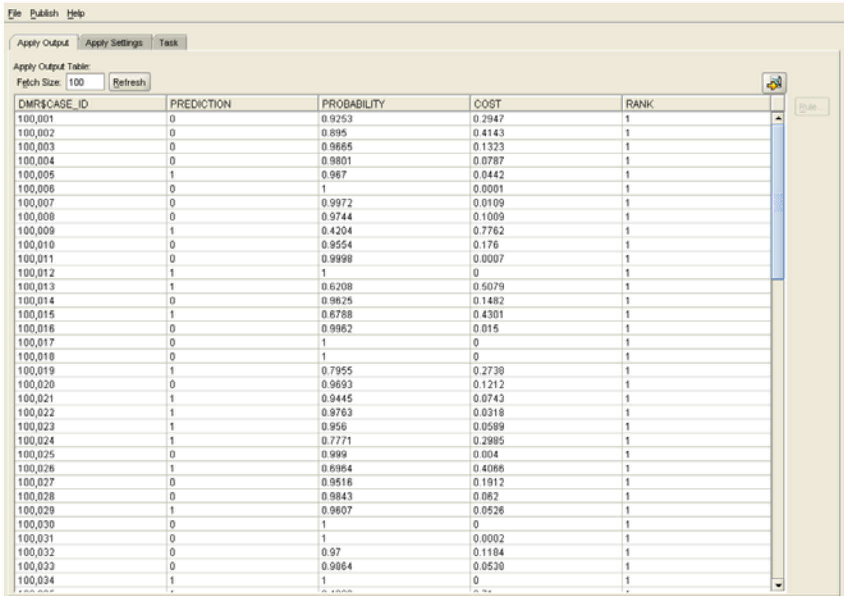
After making a choice of format, click Next.



Enter a name similar to the Build Activity and click Next, then Finish on the final page.



When the activity completes, click Result in the Apply step to see a sample of the output table. The format shown is the Most Probable, so the table contains, on each row, the identifier, the most likely target value, and the probability (confidence in that prediction). Cost is another measure – low Cost means high Probability – and it represents the cost of an incorrect prediction. In this format, the rank for each prediction is 1; if you asked for the top three predictions, each row for a given case would have rank 1, 2, or 3.



DMR$CASE_ID	PREDICTION	PROBABILITY	COST	RANK
100,001	0	0.9253	0.2947	1
100,002	0	0.895	0.4143	1
100,003	0	0.9665	0.1323	1
100,004	0	0.9801	0.0787	1
100,005	1	0.967	0.0442	1
100,006	0	1	0.0001	1
100,007	0	0.9972	0.0109	1
100,008	0	0.9744	0.1009	1
100,009	1	0.4204	0.7762	1
100,010	0	0.9554	0.176	1
100,011	0	0.9998	0.0007	1
100,012	1	1	0	1
100,013	1	0.6208	0.5079	1
100,014	0	0.9625	0.1482	1
100,015	1	0.6788	0.4301	1
100,016	0	0.9962	0.015	1
100,017	0	1	0	1
100,018	0	1	0	1
100,019	1	0.7955	0.2738	1
100,020	0	0.9693	0.1212	1
100,021	1	0.9445	0.0743	1
100,022	1	0.9763	0.0318	1
100,023	1	0.956	0.0589	1
100,024	1	0.7771	0.2985	1
100,025	0	0.999	0.004	1
100,026	1	0.6964	0.4066	1
100,027	0	0.9516	0.1912	1
100,028	0	0.9843	0.062	1
100,029	1	0.9607	0.0526	1
100,030	0	1	0	1
100,031	0	1	0.0002	1
100,032	0	0.97	0.1184	1
100,033	0	0.9864	0.0530	1
100,034	1	1	0	1

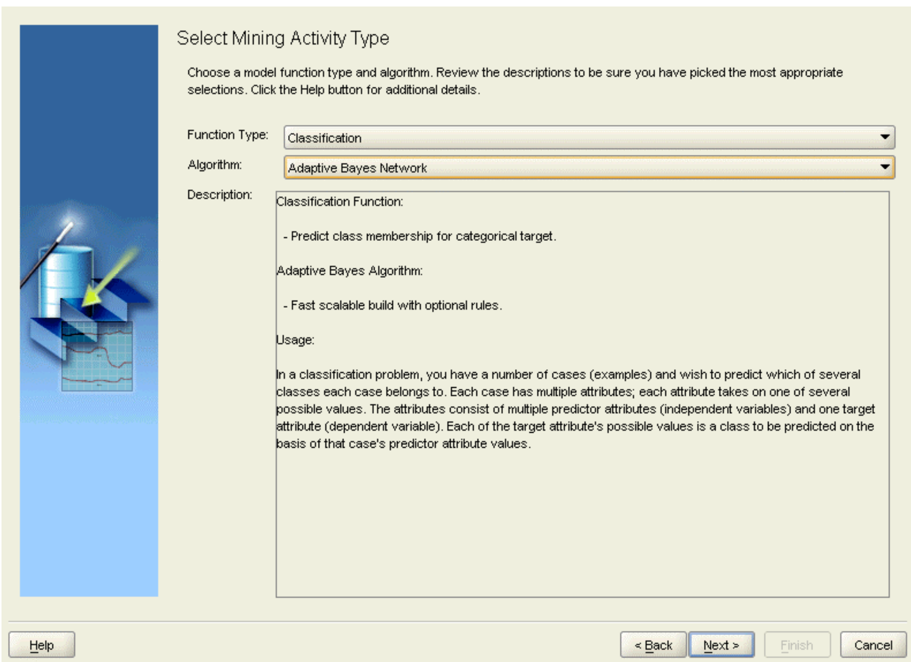
Chapter 6 – Classification: Adaptive Bayes Network

<u>NOTE:</u> Oracle Data Miner 11.1 does *not* support Adaptive Bayes Network for classification; use Decision Tree , described in Chapter 7, if you need rules.

A solution to a Classification problem predicts a discrete value for each case: 0 or 1; Yes or No; Low, Medium, or High.

Oracle Data Mining provides four algorithms for solving Classification problems; the nature of the data determines which method will provide the best business solution, so normally you find the best model for each algorithm and pick the best of those for deployment. This chapter discusses the Adaptive Bayes Network algorithm.

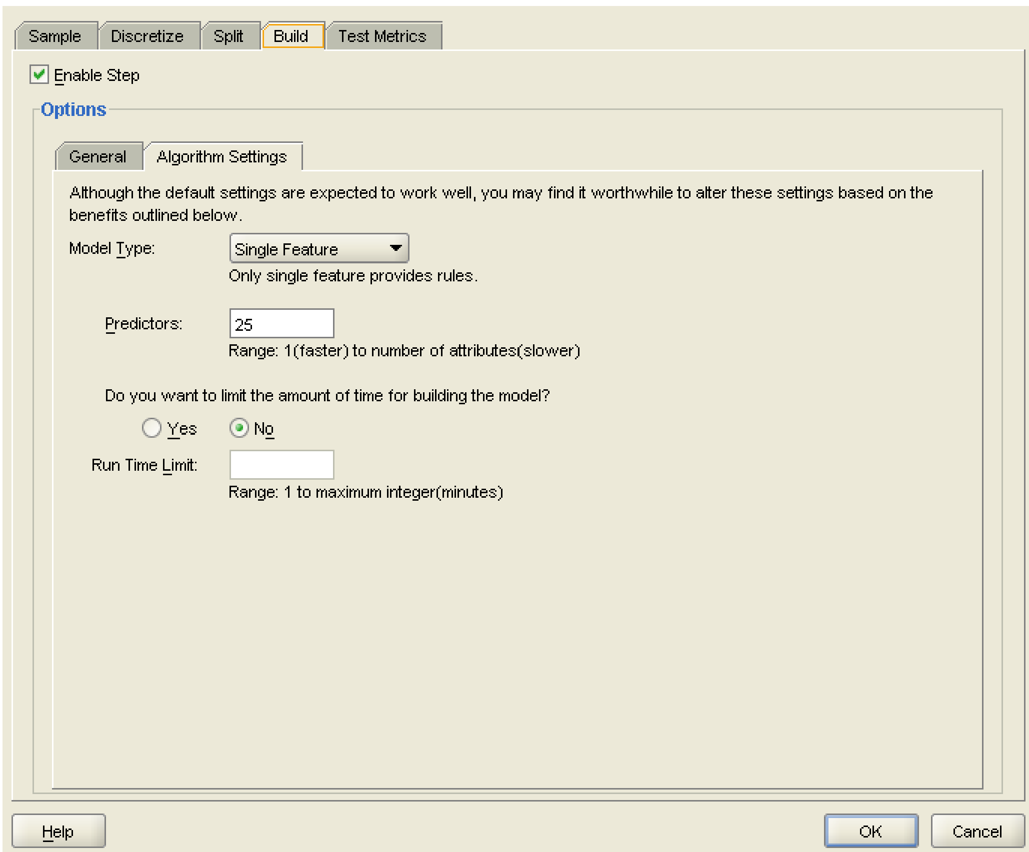
Select Build from the Activity pull-down menu to launch the activity. Select Classification as the Functionality and Adaptive Bayes Network as the Algorithm. Click Next.



Using as source data MINING_DATA_BUILD_V and target AFFINITY_CARD, all steps in the Build Activity until the Final Step are identical to those for Naïve Bayes; refer to Chapter 5 for explanations.

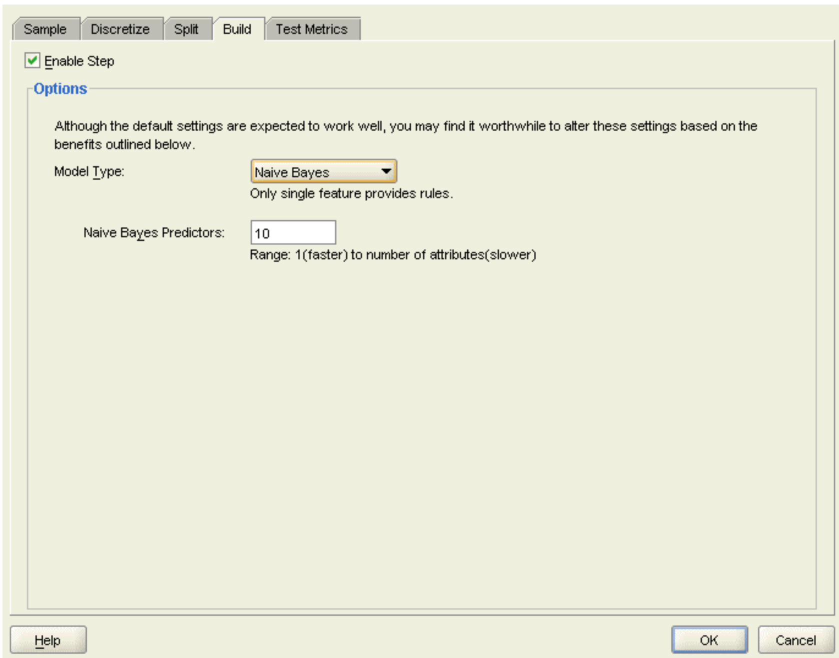
On the Final Step page, click Advanced Settings to see (and possibly modify) the default settings. All settings pages except Build-Algorithm Settings are identical to those for Naïve Bayes; refer to Chapter 5 for explanations.

There are three choices for Model Type; the default is Single Feature.

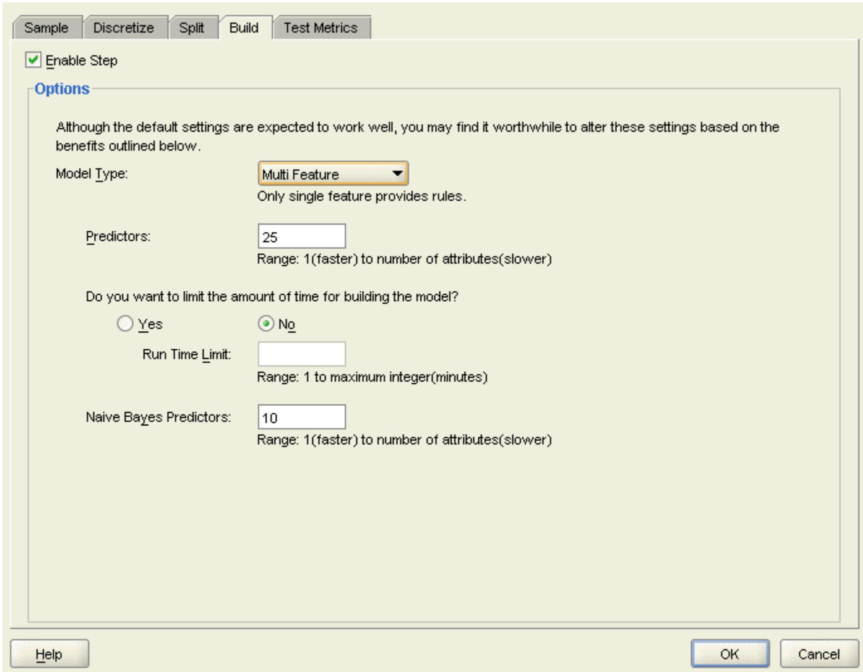


ABN begins by ranking the attributes using a form of Attribute Importance, and then builds a Naive Bayes model as a baseline using fixed parameter settings (both thresholds set to 0) and the number of attributes (Naïve Bayes Predictors – see the Multi-feature display below) specified by the user taken in order from the ranked list, so it's not exactly what you'd get by using ODM's NB algorithm directly.

You may choose Naïve Bayes as the Model Type to stop the build process at this point. If you have run Attribute Importance to determine the number of attributes having positive influence on the predictive power of the model, then you can enter that number in Naïve Bayes Predictors for the most efficient Naïve Bayes model.



If Multi-Feature is chosen as the Model Type, then ABN begins to build a sequence of little "trees" called features; each feature has a number of levels determined by the fact that adding a new level doesn't add to the model's accuracy. When the depth set by this test is reached, a new feature is built with root node split on the attribute next on the ranked list.



At each step of the building process, the model is tested against the model prior to the last step, including the baseline NB model. In particular, when an individual feature completes building, it is tested versus the model without that feature, and if there's no improvement, the new feature is discarded (pruned). When the number of consecutive discarded features reaches a number set internally by the algorithm, ABN stops building and what remains is the completed model.

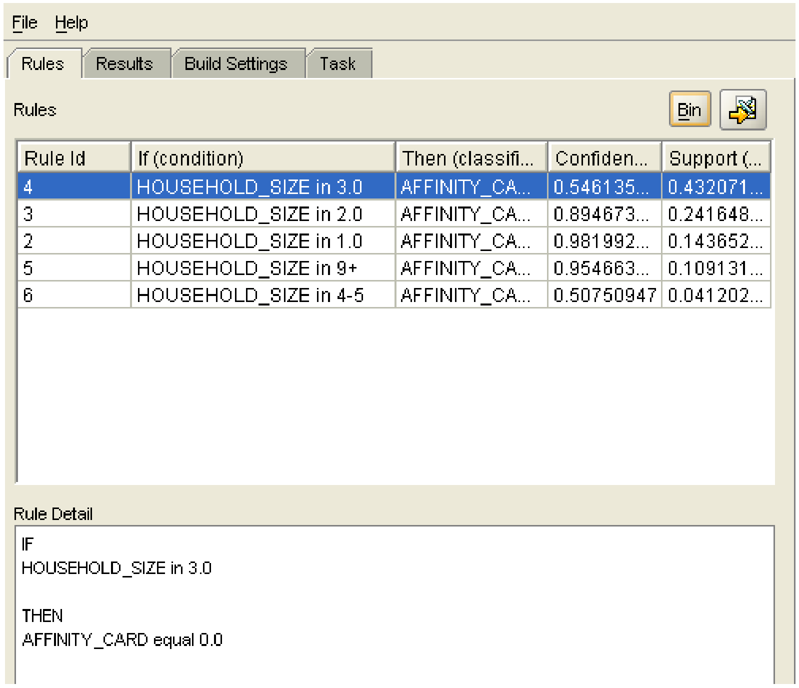
In a development environment, it may be desirable to limit the build time in early experiments; this can be done by clicking the radio button next to Yes and entering a number of minutes in the Run Time Limit window. When the specified elapsed time is reached, ABN stops building at the next convenient stopping point.

If you require human readable rules, then you must choose the option of Single Feature Build (the default).

Click OK to return to the final step and click Finish to run the activity.

When the activity completes, click Result in the Test Metrics step to evaluate the model. The interpretation is identical as for the Test Metrics for Naïve Bayes; refer to Chapter 5 for explanations.

If you chose the default Model Type, Single Feature, you can click Result in the Build step to see the rules generated. Since the source data is a very small sample data set, the rules are very simple. You should not expect meaningful rules unless the source data is much larger, for example over 20,000 rows.


File Help
Rules Results Build Settings Task
Rules
Bin
Rule Id | If (condition) | Then (classifi... | Confiden... | Support (...
4 | HOUSEHOLD_SIZE in 3.0 | AFFINITY_CA... | 0.546135... | 0.432071...
3 | HOUSEHOLD_SIZE in 2.0 | AFFINITY_CA... | 0.894673... | 0.241648...
2 | HOUSEHOLD_SIZE in 1.0 | AFFINITY_CA... | 0.981992... | 0.143652...
5 | HOUSEHOLD_SIZE in 9+ | AFFINITY_CA... | 0.954663... | 0.109131...
6 | HOUSEHOLD_SIZE in 4-5 | AFFINITY_CA... | 0.50750947 | 0.041202...
Rule Detail
IF
HOUSEHOLD_SIZE in 3.0
THEN
AFFINITY_CARD equal 0.0

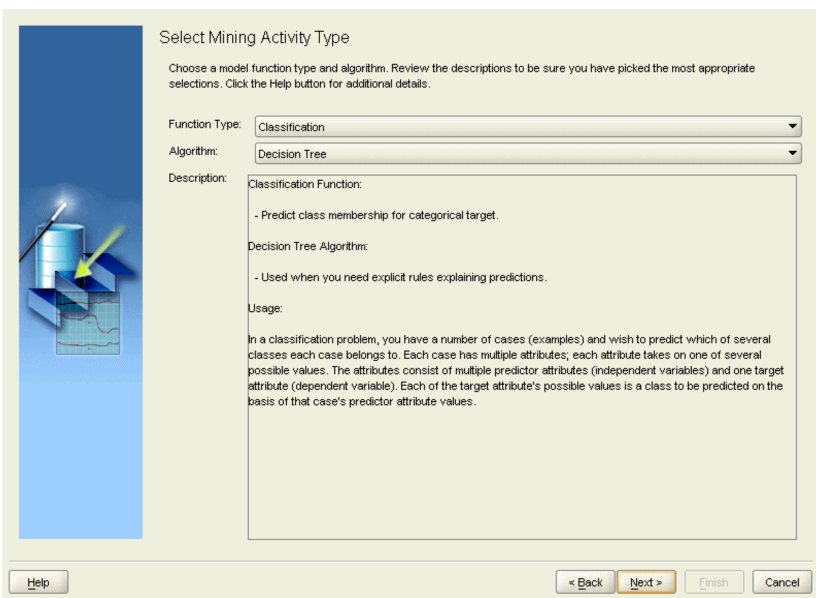
Chapter 7 – Classification: Decision Trees

A solution to a Classification problem predicts a discrete value for each case: 0 or 1; Yes or No; Low, Medium, or High.

Oracle Data Mining provides four algorithms for solving Classification problems; the nature of the data determines which method will provide the best business solution, so normally you find the best model for each algorithm and pick the best of those for deployment. This chapter discusses the Decision Tree algorithm.

Oracle Data Mining implements the Classification component of the well-known C&RT algorithm, with the added enhancement of supplying Surrogate splitting attributes, if possible, at each node (see the explanation of the Build results, below).

Select Build from the Activity pull-down menu to launch the activity. Select Classification as the Functionality and Decision Tree as the algorithm. Click Next.



All steps in the Build Activity until the Final Step are identical to those for Naïve Bayes; refer to Chapter 5 for explanations.

On the Final Step page, click Advanced Settings to view or modify the default settings. All settings pages except Build are identical to those for Naïve Bayes; refer to Chapter 5 for explanations.

The Decision Tree algorithm performs internal optimization to decide which attributes to use at each branching split. At each split, a Homogeneity Metric is used to determine the attribute values on each side of the binary branching that ensures that the cases satisfying each splitting criterion are predominantly of one target value. For example, it might be determined that most customers over the age of 35 are high-value customers, while those below 35 are low-value customers. There are two Homogeneity Metrics – Gini and Entropy – with Gini being the default. Gini tries to make one side of the branch as "pure" as possible (that is, the highest possible percentage of one class), while Entropy attempts to balance the branches as well as separating the classes as much as possible.

The building of the tree by creating branches continues until one of several user-defined stopping rules is met. A node is said to contain N records if N cases of the source data satisfy the branching rules to that point. Using the default values shown below, the branching stops if:

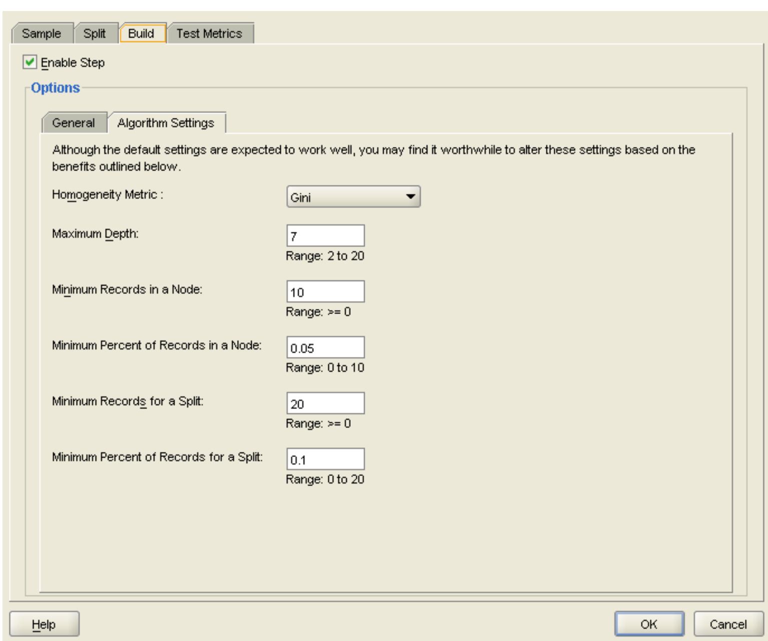
- the branching has created 7 levels of branches in the tree

A node is not split further if:

- a node contains fewer than 20 records
- a node contains less than 10% of source records

A split is rolled back if it produces a node:

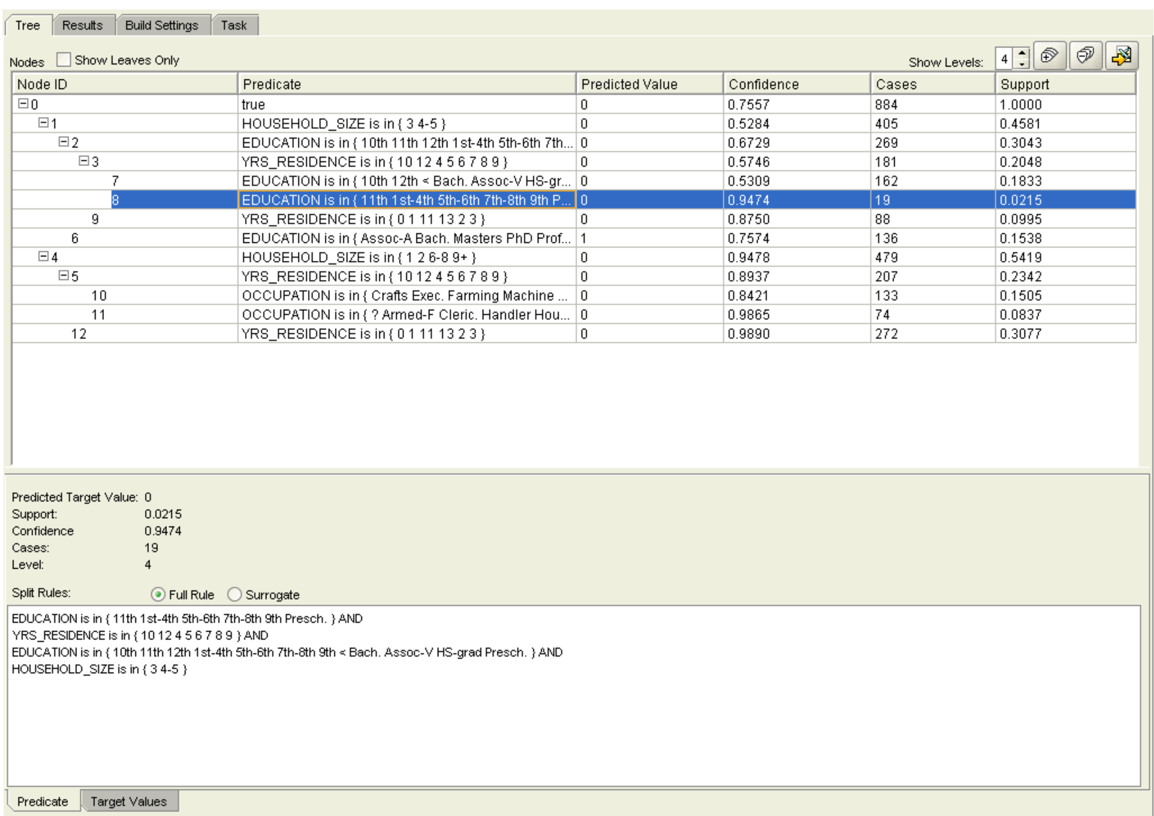
- with fewer than 10 records
- with less than 5% of the source records



When the activity completes, click Result in the Test Metrics step to evaluate the model. The interpretation is identical as for the Test Metrics for Naïve Bayes; refer to Chapter 5 for explanations.

To see the structure of the tree, click Result in the Build step. The default view shows all nodes and the attribute values used to determine splits. For example, Node 1 is split into nodes 2 and 6 based on the value of the attribute EDUCATION. You can highlight a node to show the rule for a record to be included in that node.

- Predicted Value is the target value of the majority of records in that node.
- Confidence is the percentage of records in the node having the predicted target value.
- Cases is the actual number of cases in the source data satisfying the rule for that node.
- Support is the percentage of cases in the source data satisfying the rule for that node.


Tree | Results | Build Settings | Task

Nodes | Show Leaves Only | Show Levels: 4

Node ID	Predicate	Predicted Value	Confidence	Cases	Support
0	true	0	0.7557	884	1.0000
1	HOUSEHOLD_SIZE is in { 3 4-5 }	0	0.5284	405	0.4581
2	EDUCATION is in { 10th 11th 12th 1st-4th 5th-6th 7th...	0	0.6729	269	0.3043
3	YRS_RESIDENCE is in { 10 12 4 5 6 7 8 9 }	0	0.5746	181	0.2048
7	EDUCATION is in { 10th 12th < Bach. Assoc-V HS-gr...	0	0.5309	162	0.1833
8	EDUCATION is in { 11th 1st-4th 5th-6th 7th-8th 9th P...	0	0.9474	19	0.0215
9	YRS_RESIDENCE is in { 0 1 11 13 2 3 }	0	0.8750	88	0.0995
6	EDUCATION is in { Assoc-A Bach. Masters PhD Prof...	1	0.7574	136	0.1538
4	HOUSEHOLD_SIZE is in { 1 2 6-8 9+ }	0	0.9478	479	0.5419
5	YRS_RESIDENCE is in { 10 12 4 5 6 7 8 9 }	0	0.8937	207	0.2342
10	OCCUPATION is in { Crafts Exec. Farming Machine ...	0	0.8421	133	0.1505
11	OCCUPATION is in { ? Armed-F Cleric. Handler Hou...	0	0.9865	74	0.0837
12	YRS_RESIDENCE is in { 0 1 11 13 2 3 }	0	0.9890	272	0.3077

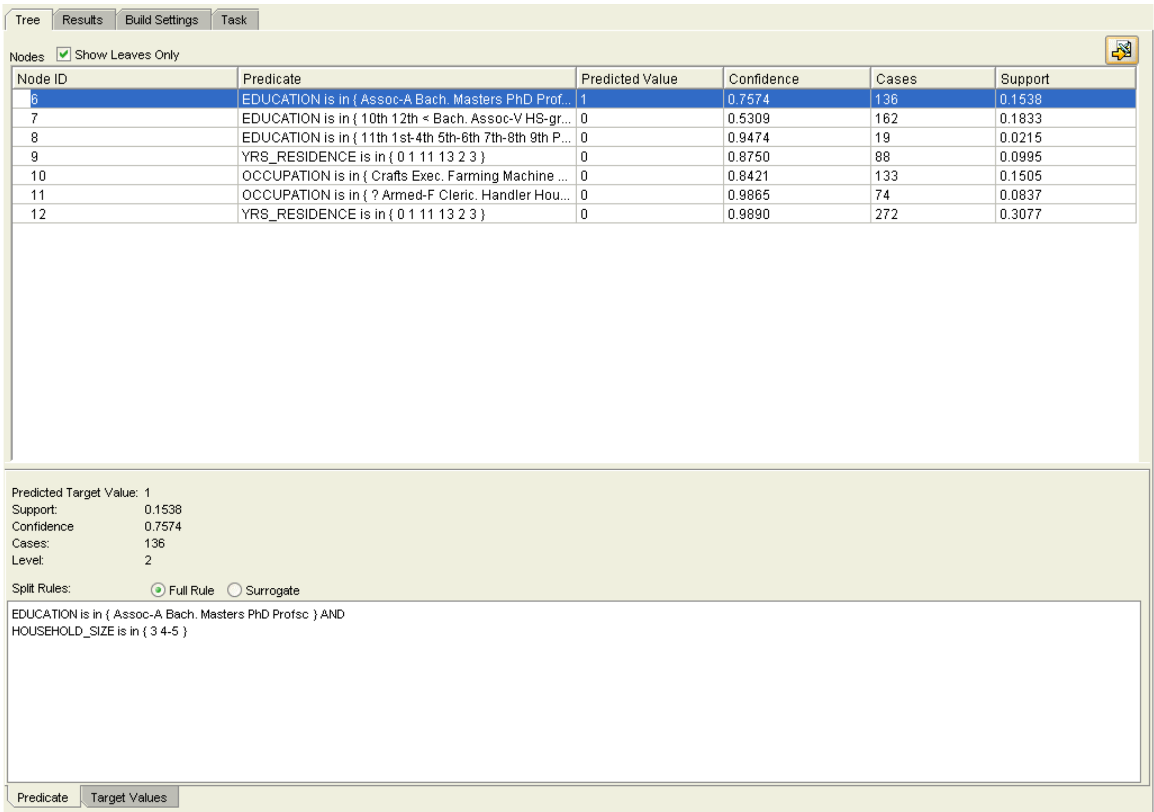
Predicted Target Value: 0
Support: 0.0215
Confidence 0.9474
Cases: 19
Level: 4

Split Rules: Full Rule | Surrogate

EDUCATION is in { 11th 1st-4th 5th-6th 7th-8th 9th Presch. } AND
YRS_RESIDENCE is in { 10 12 4 5 6 7 8 9 } AND
EDUCATION is in { 10th 11th 12th 1st-4th 5th-6th 7th-8th 9th < Bach. Assoc-V HS-grad Presch. } AND
HOUSEHOLD_SIZE is in { 3 4-5 }

Predicate | Target Values

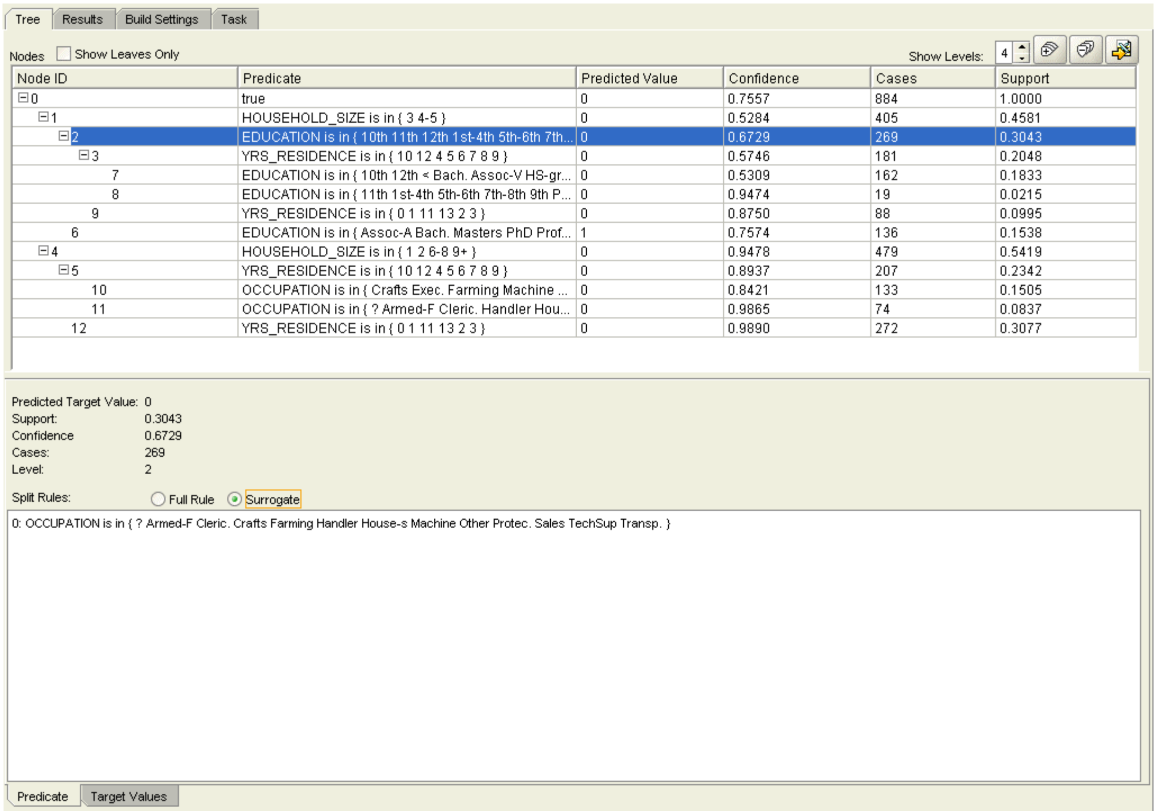
Click the checkbox Show Leaves Only to eliminate the intermediate nodes and to display only the terminal nodes (also called Leaves); these are the nodes used to make the predictions when the model is applied to new data.



A decision tree is sensitive to missing values when applied to new data. For example, if a split in the tree (and therefore an element in the rule determining the prediction) uses the attribute Household_size, and Household_size is missing in a record to be scored, then the scoring might fail. However, if the splitting attribute is missing, the ODM Decision Tree algorithm provides an alternative attribute (known as a surrogate) to be used in its place, if another attribute can be found that is somewhat correlated to the missing attribute. If both the splitting attribute and its surrogate are missing, the predicted value is determined at the parent node of the split.

To display the surrogate, highlight a node and click the radio button Surrogate.

If the model shown is applied to a record with no value in the EDUCATION column, then the value in OCCUPATION will be used to determine a prediction.



Chapter 8 – Classification: Support Vector Machines

A solution to a Classification problem predicts a discrete value for each case: 0 or 1; Yes or No; Low, Medium, or High.

Oracle Data Mining provides four algorithms for solving Classification problems; the nature of the data determines which method will provide the best business solution, so normally you find the best model for each algorithm and pick the best of those for deployment. This chapter discusses the Support Vector Machines algorithm.

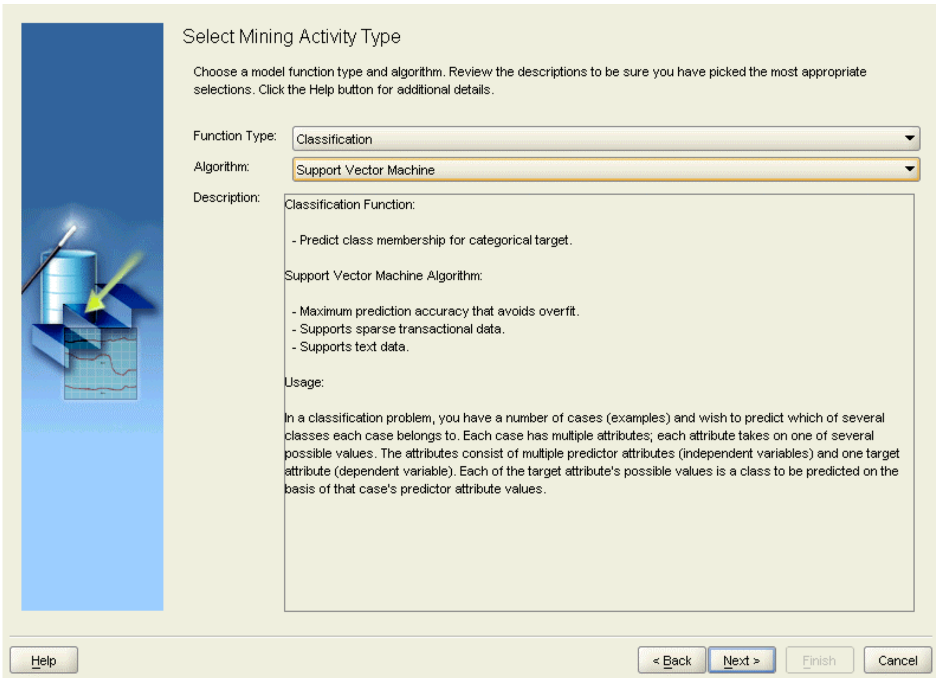
Oracle Data Mining's Support Vector Machines (SVM) algorithm is actually a suite of algorithms, adaptable for use with a variety of problems and data. By swapping one kernel for another, SVM can fit diverse problem spaces. Oracle Data Mining supports two kernels, Linear and Gaussian.

Data records with N attributes can be thought of as points in N-dimensional space, and SVM attempts to separate the points into subsets with homogeneous target values; points are separated by hyperplanes in the linear case, and in the non-linear case (Gaussian) by non-linear separators. SVM finds the vectors that define the separators giving the widest separation of classes (the "support vectors"). This is easy to picture in the case of N = 2; then the solution defines a straight line (linear) or a curve (non-linear) separating the differing classes of points in the plane.

SVM solves regression problems by defining an N-dimensional "tube" around the data points, determining the vectors giving the widest separation. See Chapter 9 for a discussion of the Regression case.

SVM can emulate some traditional methods, such as linear regression and neural nets, but goes far beyond those methods in flexibility, scalability, and speed. For example, SVM can act like a neural net in calculating predictions, but can work on data with thousands of attributes, a situation that would stymie a neural net. Moreover, while a neural net might mistake a local change in direction as a point of minimum error, SVM will work to find the global point of minimum error.

Select Build from the Activity pull-down menu to launch the activity. Select Classification as the Functionality and Support Vector Machines as the algorithm. Click Next.

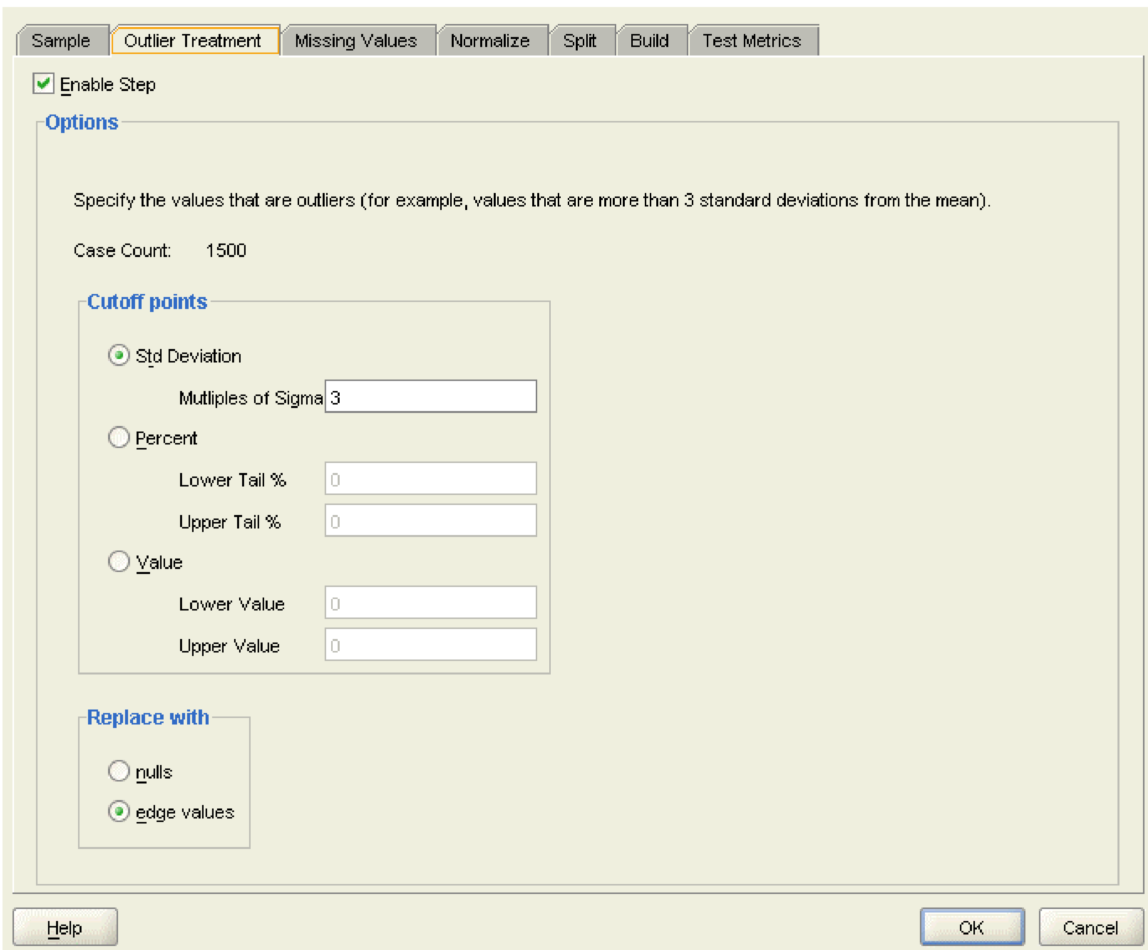


All steps in the Build Activity until the Final Step are identical to those for Naïve Bayes; refer to Chapter 5 for explanations.

On the Final Step page, click Advanced Settings to view or modify the default settings. The Sample, Split, and Test Metrics settings pages are identical to those for Naïve Bayes; refer to Chapter 5 for explanations.

Click the Outlier Treatment tab to modify settings related to extreme values. The default treatment, as shown, recodes any value farther than three standard deviations from the mean to a value exactly three standard deviations from the mean.

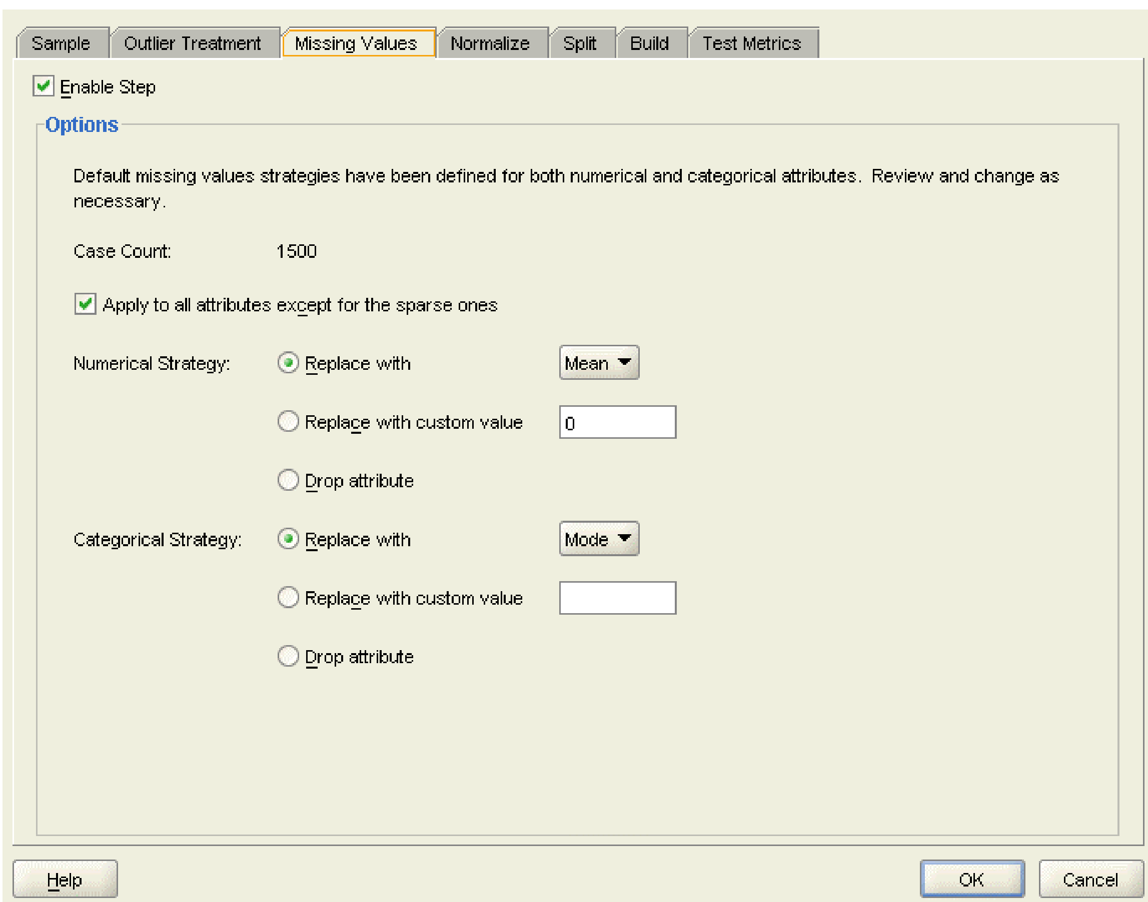
You can change the definition of “outlier” by changing the number of standard deviations, or by entering an explicit cutoff point, either as a percentage of records or as an actual value. You can also choose to discard extreme values rather than to recode them to “edge” values.



SVM is sensitive to missing values. The default treatment replaces a numerical missing value with the Mean (Average) for that attribute, and replaces a categorical missing value with the Mode (the most frequently occurring value).

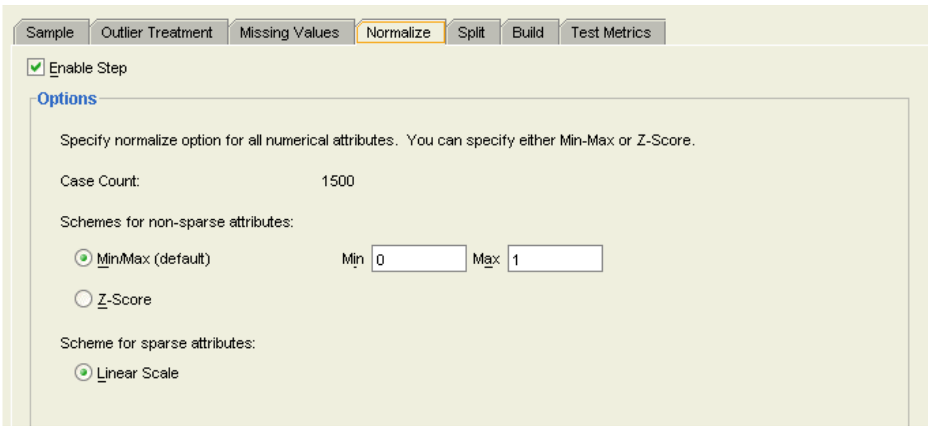
You can choose to replace a missing value with a fixed user-defined value. (See the discussion of the Predict function in Appendix C for an indication of how to supply missing values using a data mining algorithm).

There is a difference between a missing value that is unknown and a missing value that has meaning. For example, if an attribute contains a list of products and quantities purchased by a customer in a retail store, each entry may have a small number of products chosen from thousands of possible products. Most of the products are “missing”, with the meaning that the missing products were not purchased. Such an attribute is referred to as “sparse”, and normally you don’t want to impose a missing value treatment on that attribute. The default is to skip such attributes in defining a Missing Value Treatment, but you can change that by clicking the checkbox below the Case Count.

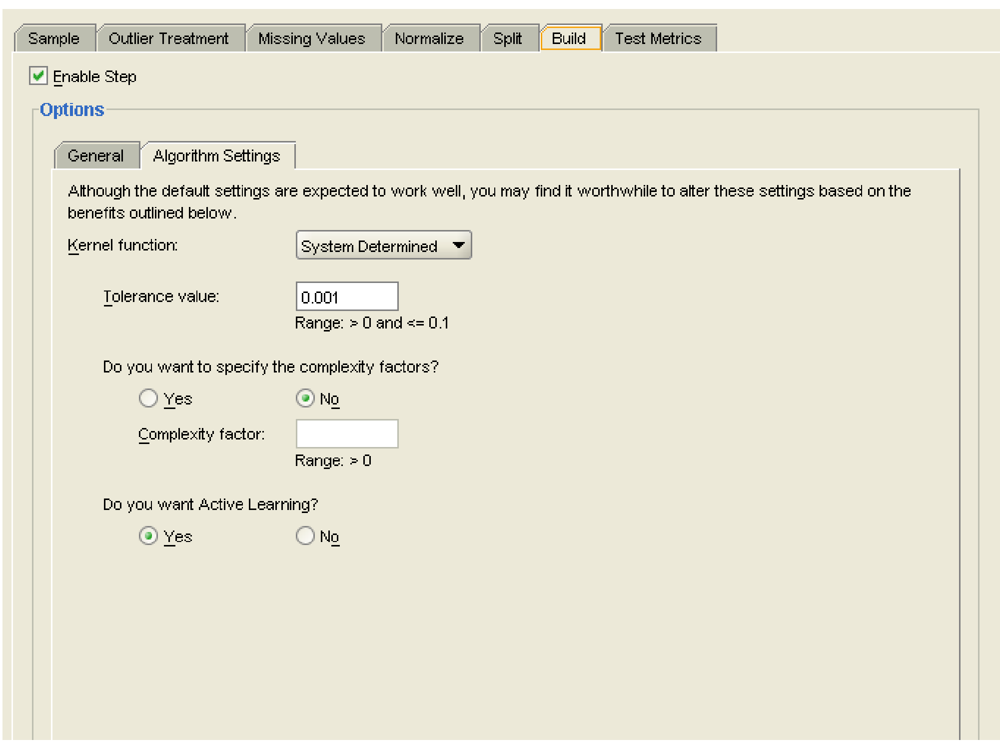


SVM requires that numerical data be normalized; the default normalization method is min-max, which recodes all values to be in the range from 0 to1, maintaining the relative position of each value. You can change the resulting range.

If the data has extreme outlier values that must be maintained, the z-score method normalizes most values to the range from –1 to 1, but allows values outside that range representing the outliers.

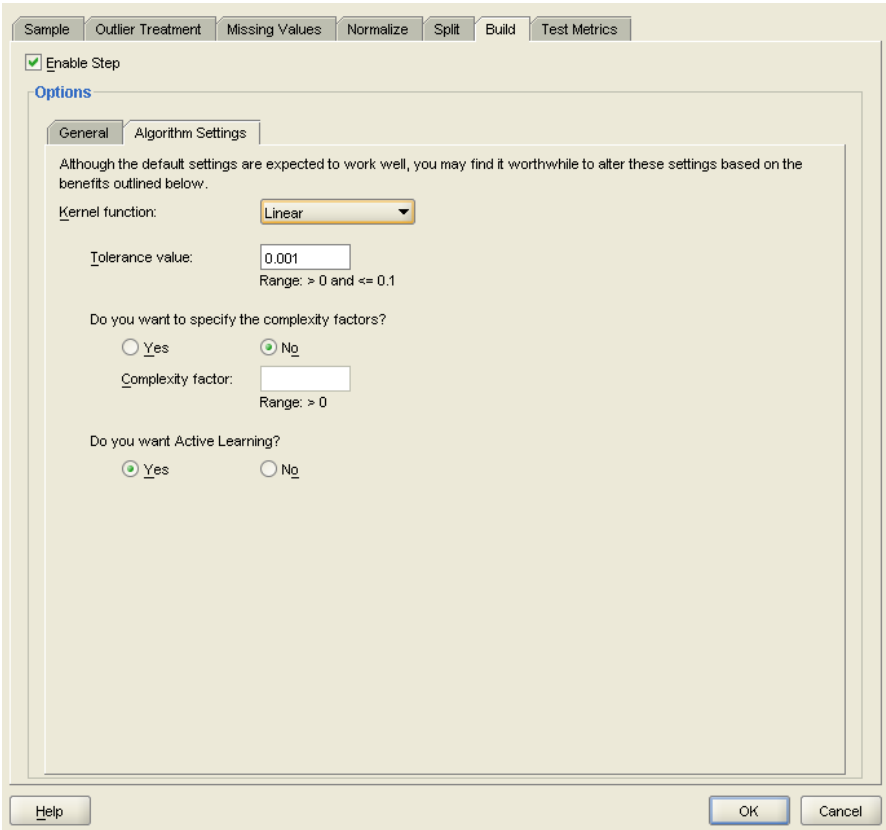


Click the Build tab, then Algorithm Settings to see the default setting for Kernel Function, which allows the algorithm to select automatically the appropriate version of SVM to use.



Click the pull-down menu for Kernel Type to see the alternatives and to expose the parameters for each. You have a choice of two kernels: Linear and Gaussian (non-linear).

For the Linear case:

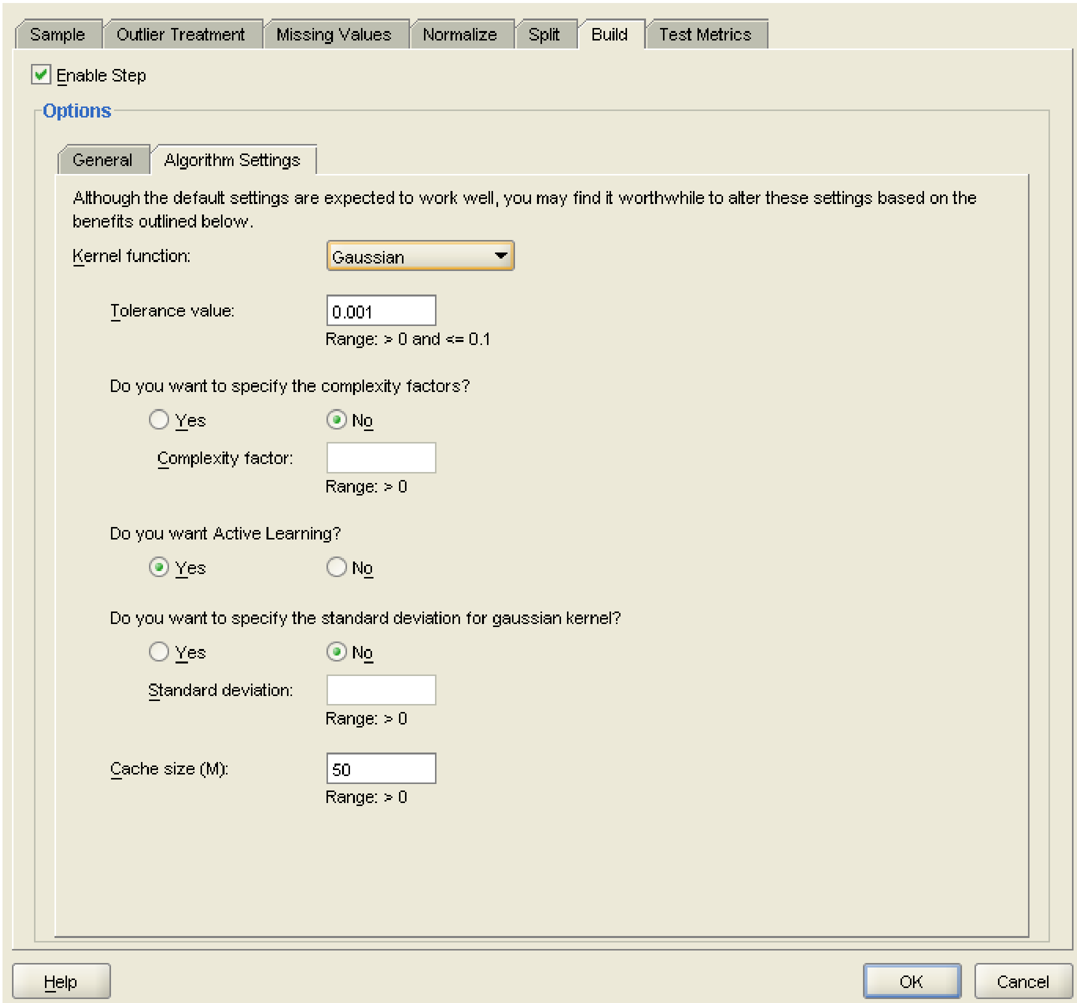


Tolerance is a stopping mechanism – a measure of when the algorithm should be satisfied with the result and consider the building process complete. The default is .001; a higher value will give a faster build but perhaps a less accurate model.

A model is called overfit (or overtrained) if it works well on the build data, but is not general enough to deal with new data. The Complexity Factor prevents overfitting by finding the best tradeoff between simplicity and complexity. The algorithm will calculate and optimize this value if you do not specify a value. If the model skews its predictions in favor of one class, you may choose to rebuild with a manually-entered complexity factor higher than the one calculated by the algorithm.

Active Learning is a methodology, internally implemented, that optimizes the selection of a subset of the support vectors which will maintain accuracy while enhancing the speed of the model. You should not disable this setting.

For the Gaussian case:



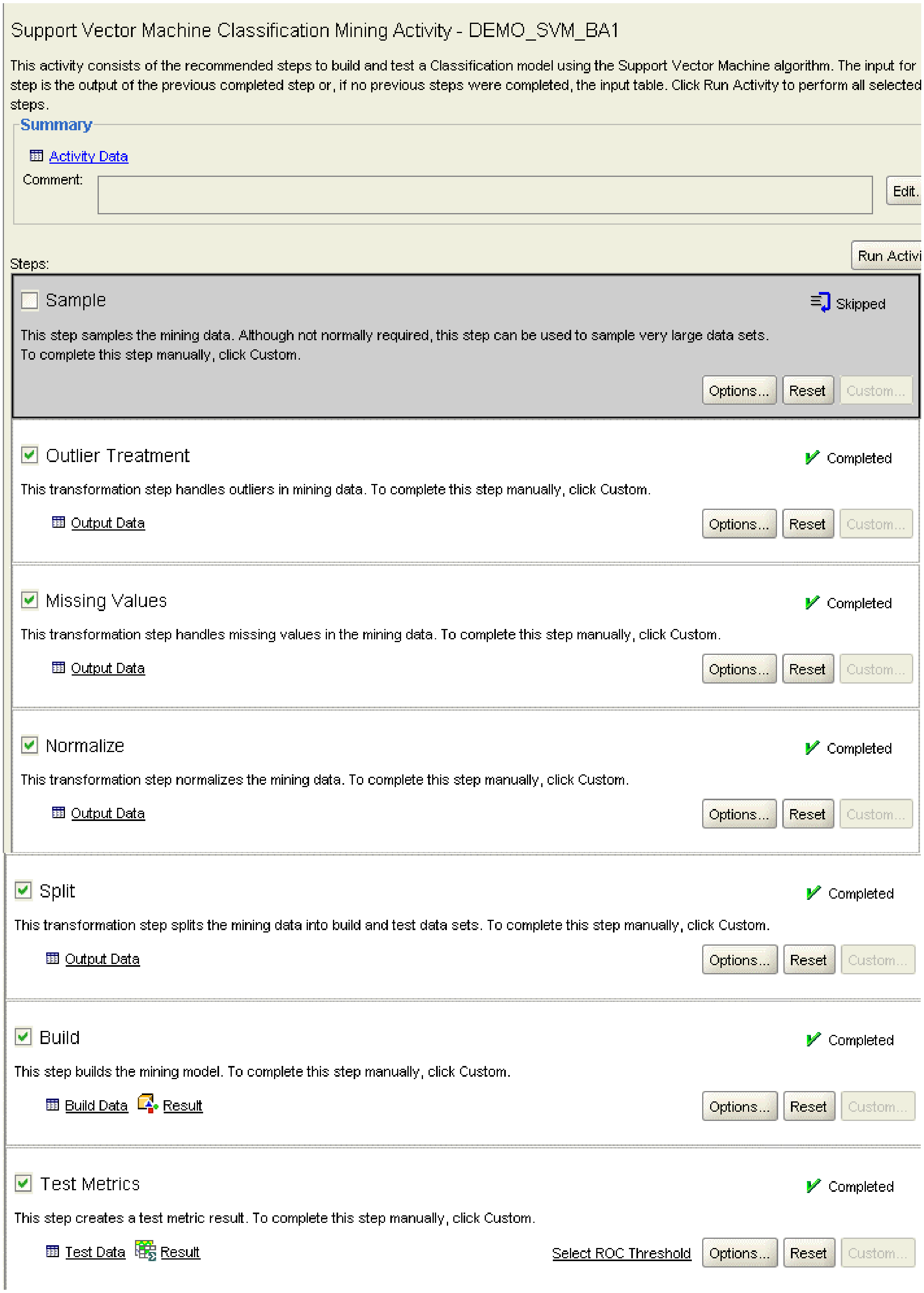
The Tolerance and Complexity settings have the same meaning as in the Linear case.

Active Learning, in addition to increasing performance as in the Linear case, will reduce the size of the Gaussian model; this is an important consideration if memory and temporary disk space are issues.

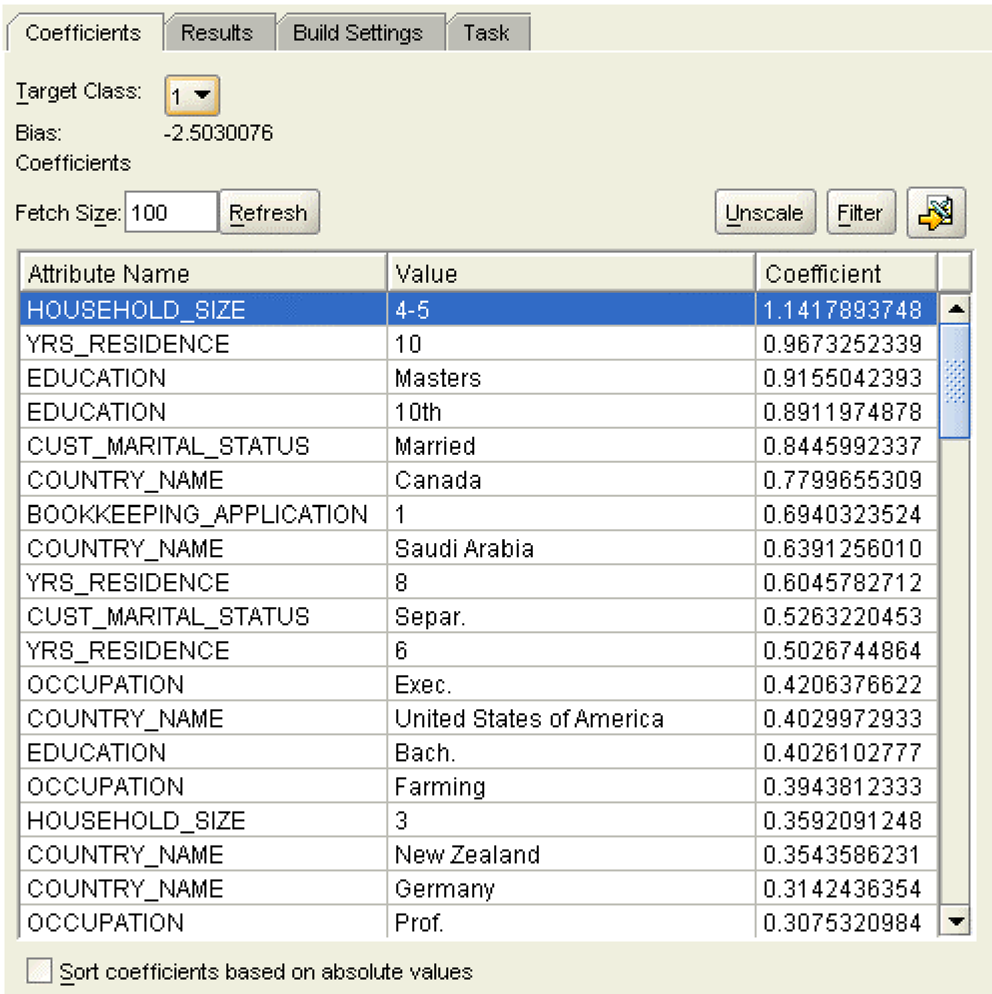
Together with the complexity factor, the number of standard deviations (sigmas) is used to find the happy medium betweem simplicity and complexity. A small value for sigma may cause overfitting, and a large value may cause excess complexity. The algorithm will calculate the ideal value internally.

The Gaussian kernel uses a large amount of memory in its calculations if Active Learning is not enabled. The default cache size is 50 Megabytes and should suffice; if the build operation seems very slow, increasing Cache size may help.

Click OK to return to the final wizard page and click Finish to run the activity.



In the case of an SVM with the Linear kernel, you can click Result in the Build step to see the Coefficients and Offset (Bias) for the model. Attribute Values with high Coefficients (either positive or negative) are the characteristics with strongest influence on the predictions.



Attribute Name	Value	Coefficient
HOUSEHOLD_SIZE	4-5	1.1417893748
YRS_RESIDENCE	10	0.9673252339
EDUCATION	Masters	0.9155042393
EDUCATION	10th	0.8911974878
CUST_MARITAL_STATUS	Married	0.8445992337
COUNTRY_NAME	Canada	0.7799655309
BOOKKEEPING_APPLICATION	1	0.6940323524
COUNTRY_NAME	Saudi Arabia	0.6391256010
YRS_RESIDENCE	8	0.6045782712
CUST_MARITAL_STATUS	Separ.	0.5263220453
YRS_RESIDENCE	6	0.5026744864
OCCUPATION	Exec.	0.4206376622
COUNTRY_NAME	United States of America	0.4029972933
EDUCATION	Bach.	0.4026102777
OCCUPATION	Farming	0.3943812333
HOUSEHOLD_SIZE	3	0.3592091248
COUNTRY_NAME	New Zealand	0.3543586231
COUNTRY_NAME	Germany	0.3142436354
OCCUPATION	Prof.	0.3075320984

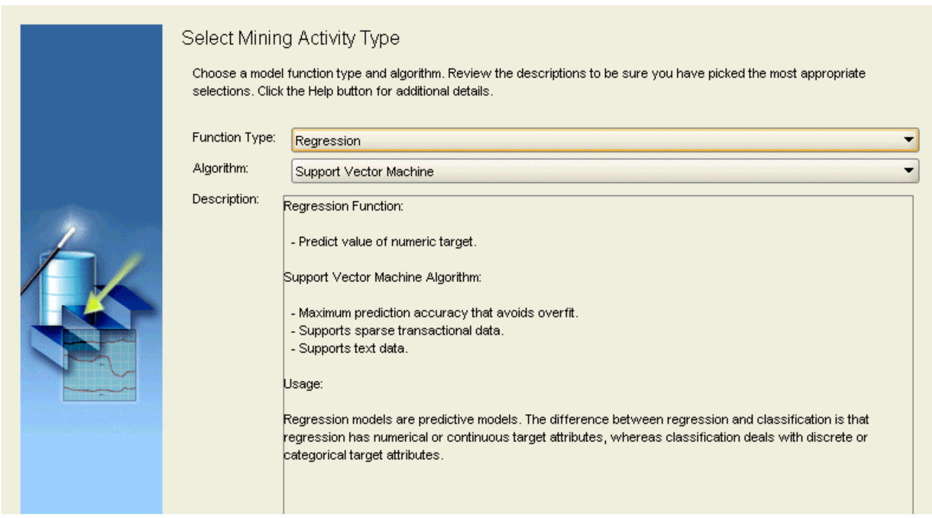
The interpretation of the Test Metrics and the form of the Apply output is similar to that for Naïve Bayes – see Chapter 5 for more detail.

Chapter 9 – Regression: Support Vector Machines

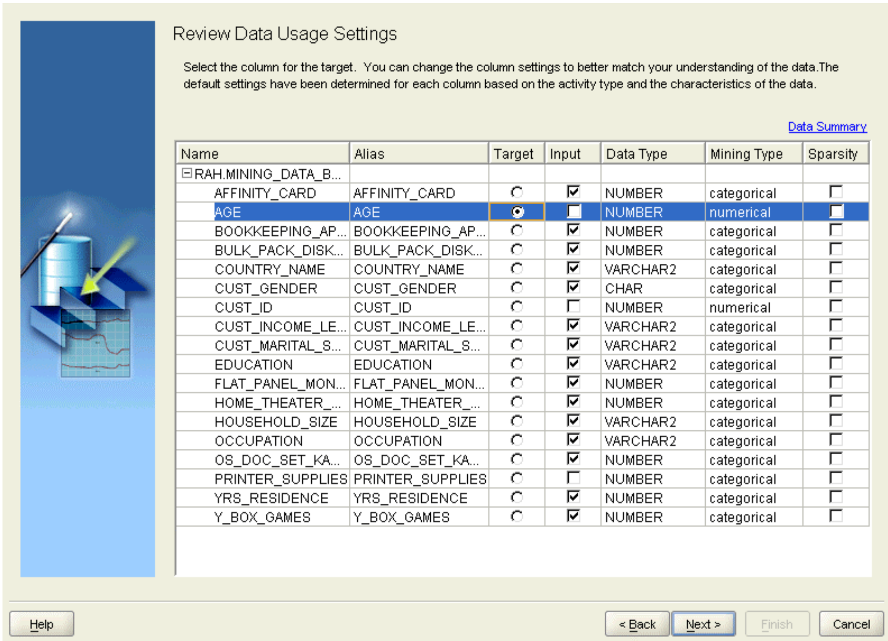
NOTE: Oracle Data Mining 11.1 support two algorithms for Regression: Support Vector Machine and Linear Regression (Generalized Linear Models). See Appendix D for more information.

The SVM algorithm can be used to predict the value of a continuous (floating point) value, usually called a regression problem. To illustrate this feature, use the same tables as for the Classification problem, but select a continuous attribute, AGE, as the target.

Select Build from the Activity pull-down menu to launch the activity. Select Regression as the Functionality; Support Vector Machine is the only choice for the algorithm. Click Next. In Step 2 specify MINING_DATA_BUILD_V as the case table and CUST_ID as the key. Click Next.



In Step 3, specify AGE as the Target.

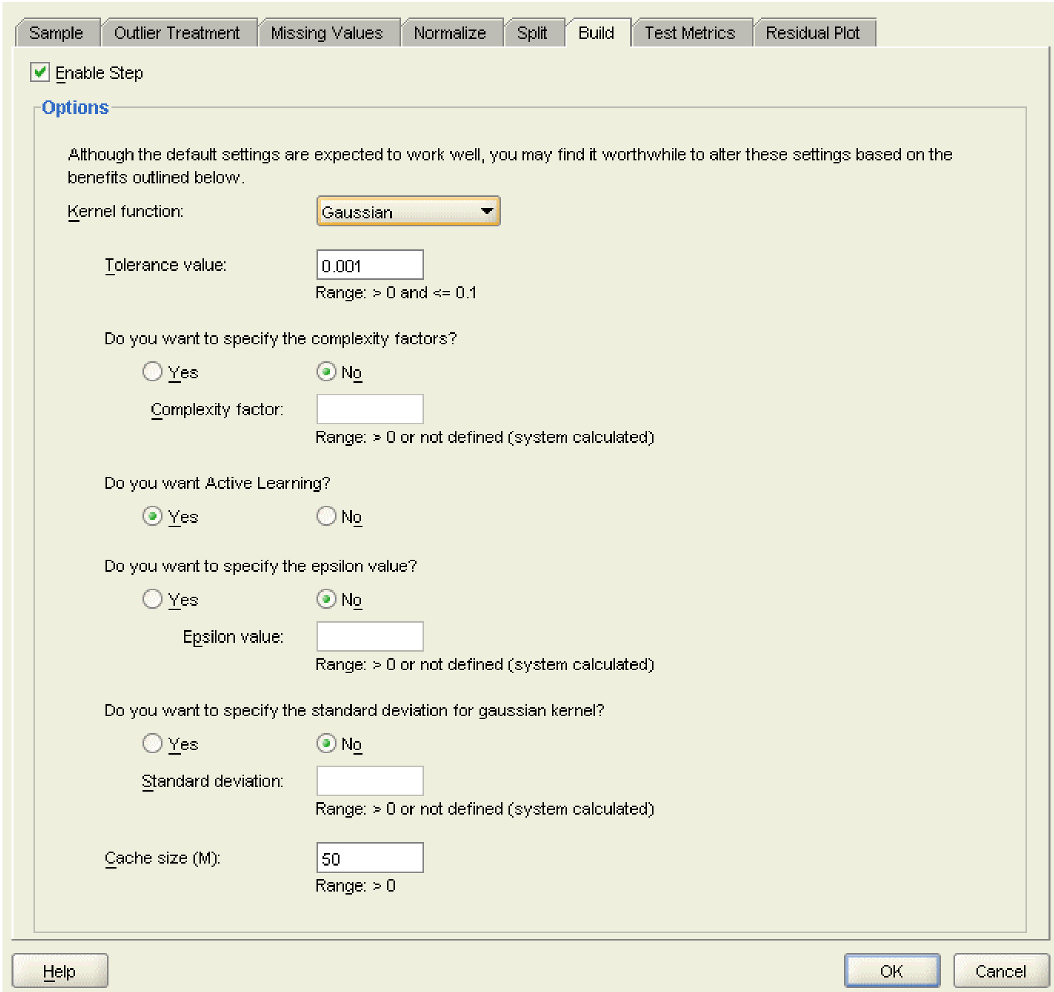


The remaining steps are the same as for previous model build activities; refer to Chapter 5 for more details.

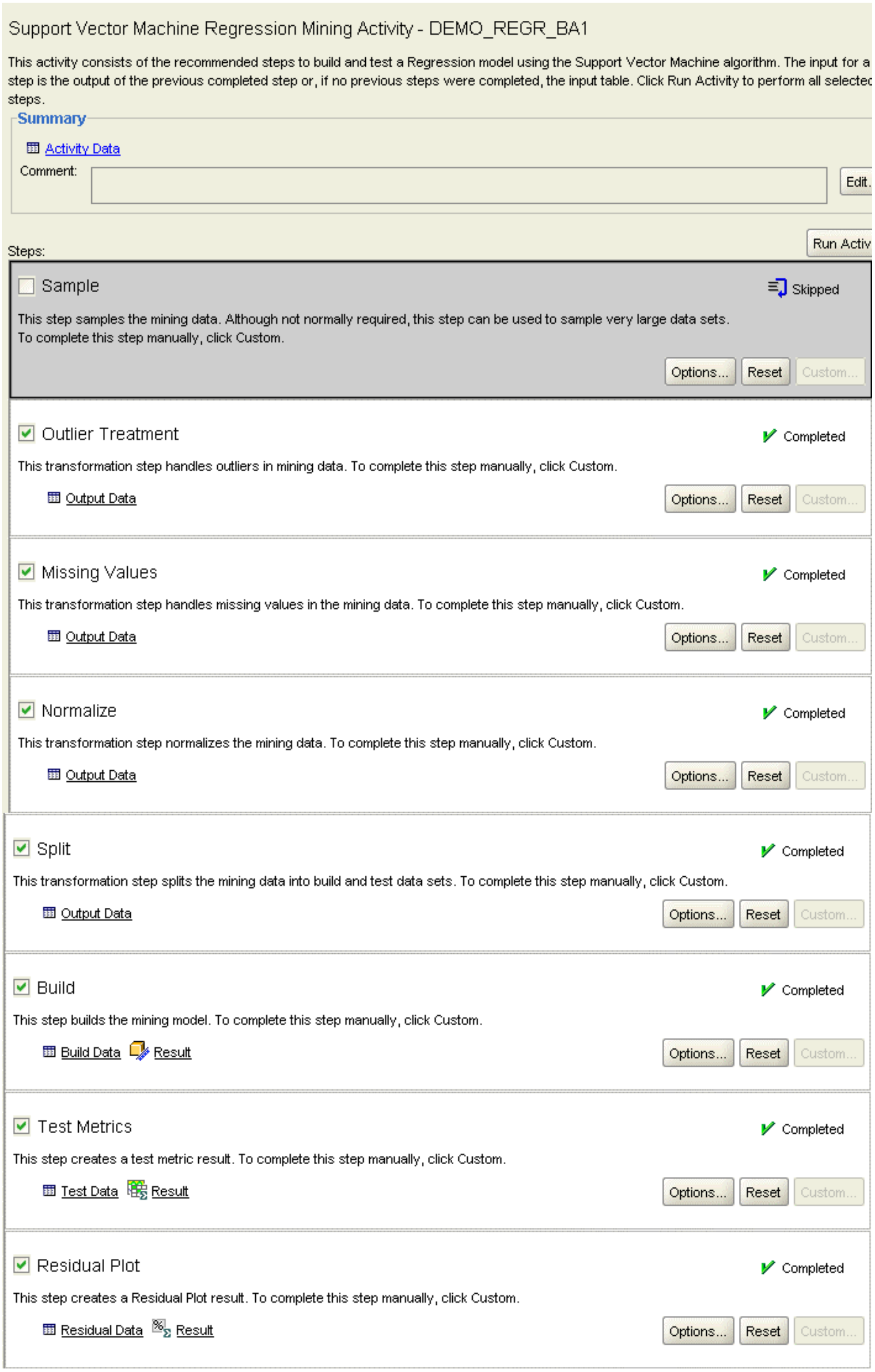
On the final page, click Advanced Settings to modify the default values. The Test Metrics and Residual Plot tabs don't expose any parameters, and all tabs except Build are identical to the tabs for SVM Classification. Refer to Chapter 8 for more details.

The default setting for Build allows the algorithm to select and optimize all parameters, including the choice of kernel. If either Linear or Gaussian kernel is chosen explicitly, the only parameter different from the Classification case (See Chapter 8) is Epsilon Value.

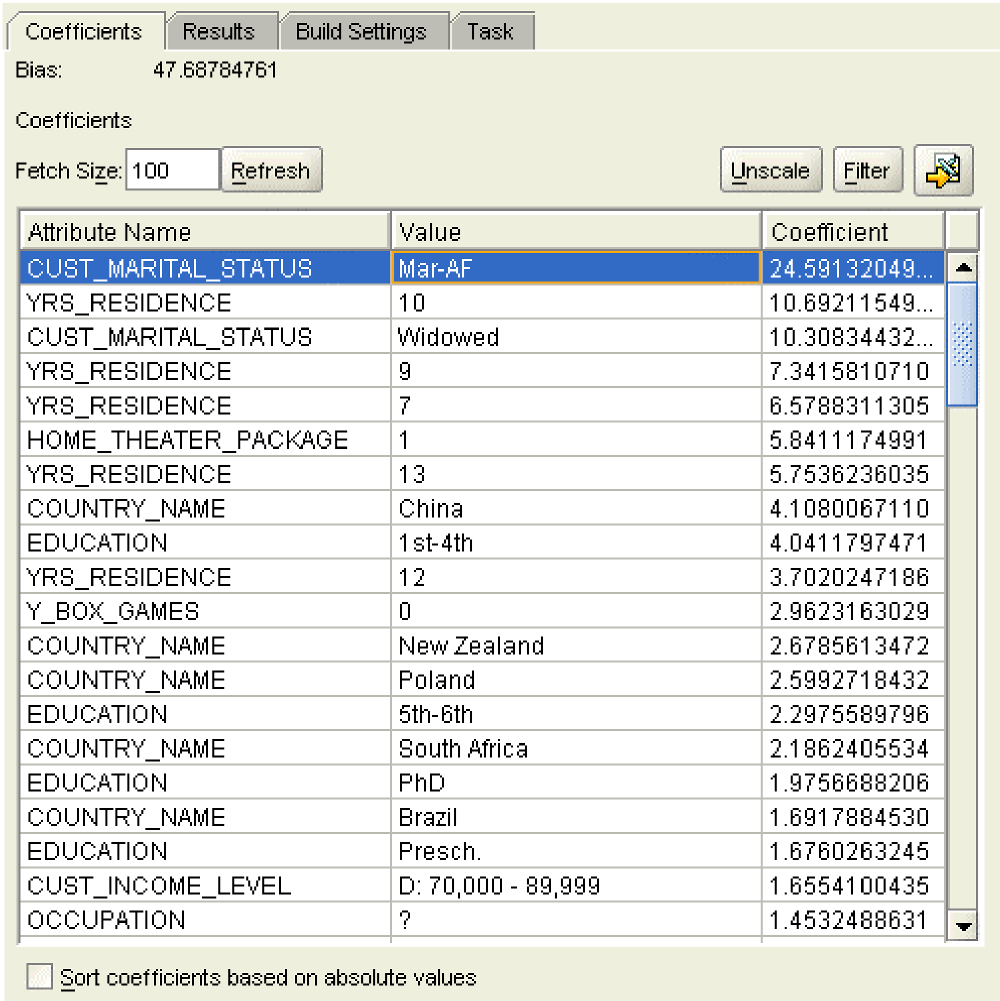
SVM makes a distinction between small errors and large errors; the difference is defined by the epsilon value. The algorithm will calculate and optimize an epsilon value internally, or you can supply a value. If there are very high cardinality categorical attributes, try decreasing epsilon, after an initial execution to determine the system-calculated value.



Click OK to return to the final page of the wizard, and Finish to run the activity.

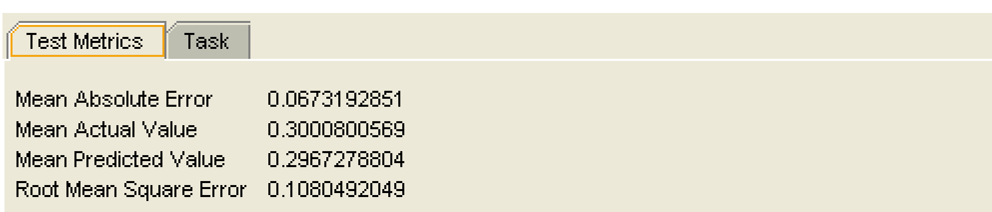


When the activity completes, you can click Result in the Build step to see the Coefficients if the Linear kernel was used.



Attribute Name	Value	Coefficient
CUST_MARITAL_STATUS	Mar-AF	24.59132049...
YRS_RESIDENCE	10	10.69211549...
CUST_MARITAL_STATUS	Widowed	10.30834432...
YRS_RESIDENCE	9	7.3415810710
YRS_RESIDENCE	7	6.5788311305
HOME_THEATER_PACKAGE	1	5.8411174991
YRS_RESIDENCE	13	5.7536236035
COUNTRY_NAME	China	4.1080067110
EDUCATION	1st-4th	4.0411797471
YRS_RESIDENCE	12	3.7020247186
Y_BOX_GAMES	0	2.9623163029
COUNTRY_NAME	New Zealand	2.6785613472
COUNTRY_NAME	Poland	2.5992718432
EDUCATION	5th-6th	2.2975589796
COUNTRY_NAME	South Africa	2.1862405534
EDUCATION	PhD	1.9756688206
COUNTRY_NAME	Brazil	1.6917884530
EDUCATION	Presch.	1.6760263245
CUST_INCOME_LEVEL	D: 70,000 - 89,999	1.6554100435
OCCUPATION	?	1.4532488631

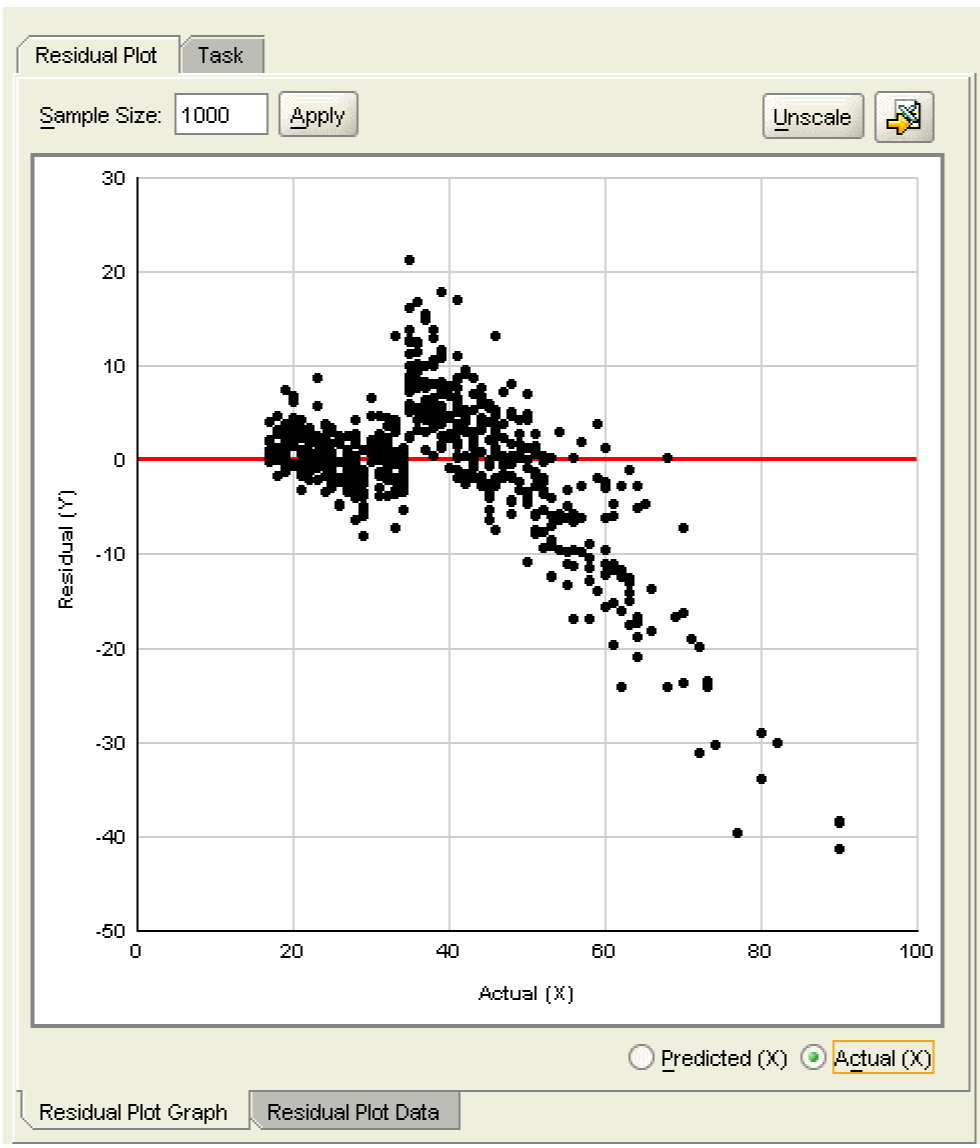
You can click Result in the Test Metrics step to see several measures of accuracy, including Root Mean Square Error (RMSE).



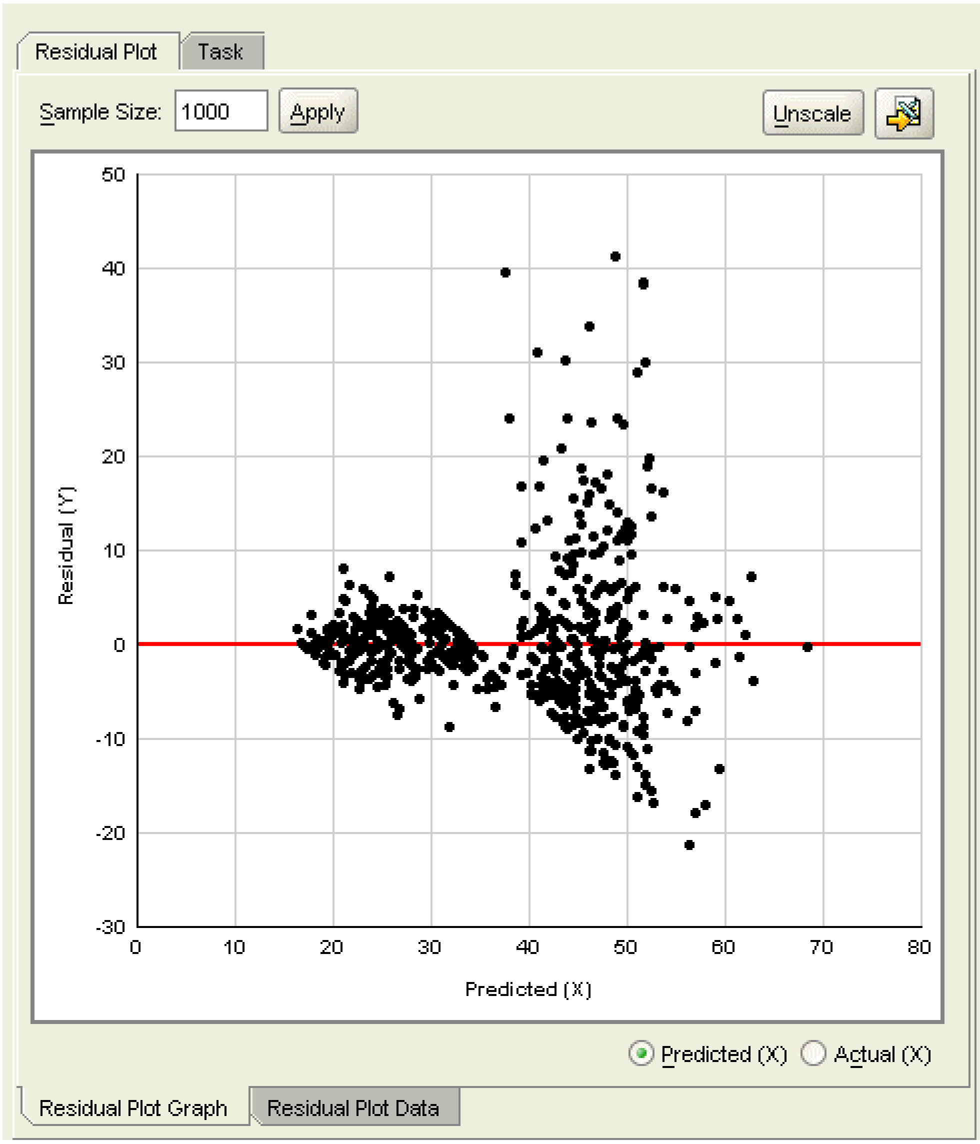
You can click Result in the Residual Plot step to see information about the residuals, that is, an indication of the difference between the actual value (in the Test dataset) and the predicted value. There are two different graphs available by clicking the appropriate radio button: the Predicted value on the X-axis or the Actual value on the X-axis. In each case, a dot on the 0 line means an exact prediction, while other dots represent the error.

The graph below uses the Actual value on the x-axis, answering the question: for what range(s) of actual values is this model likely to be accurate? Clearly, there is a change at about Age=35, with a high degree of accuracy for lower ages. In particular, a dot at AGE = 82 (X-axis) and Error = –30 (Y-axis) represents a case with actual AGE 82 that was predicted to be AGE 52.

One possible tactic, given this test evidence, would be to build two distinct models, one for ages below 35 and one for ages above 35.



The graph shown below uses predicted values on the X-axis, and answers the question: which predictions can I trust the most? The conclusion is similar to that derived above – but from a different viewpoint – in particular, predictions between the ages of 40 and 50 are not to be trusted.



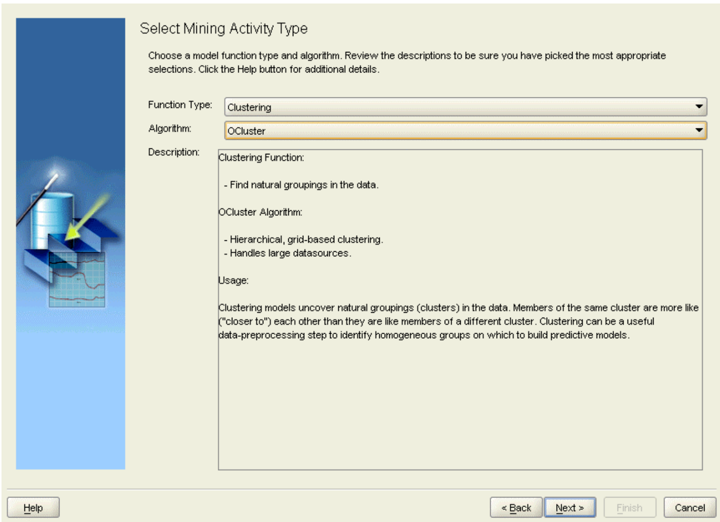
You can click on the Residual Plot Data to see a listing of the actual and predicted values for the Test dataset.

Chapter 10 – Clustering: O-Cluster

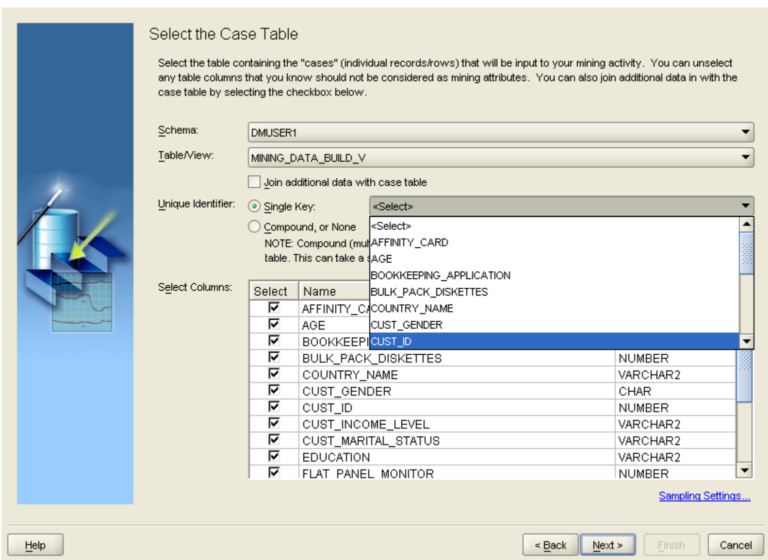
Clustering is used to identify distinct segments of a population and to explain the common characteristics of members of a cluster, and also to determine what distinguishes members of one cluster from members of another cluster.

ODM provides two Clustering algorithms, Enhanced k-means and O-cluster; this chapter will discuss O-cluster.

Choose Build from the Activity pull-down menu, then select Clustering as the Function Type and OCluster as the algorithm, Click Next.



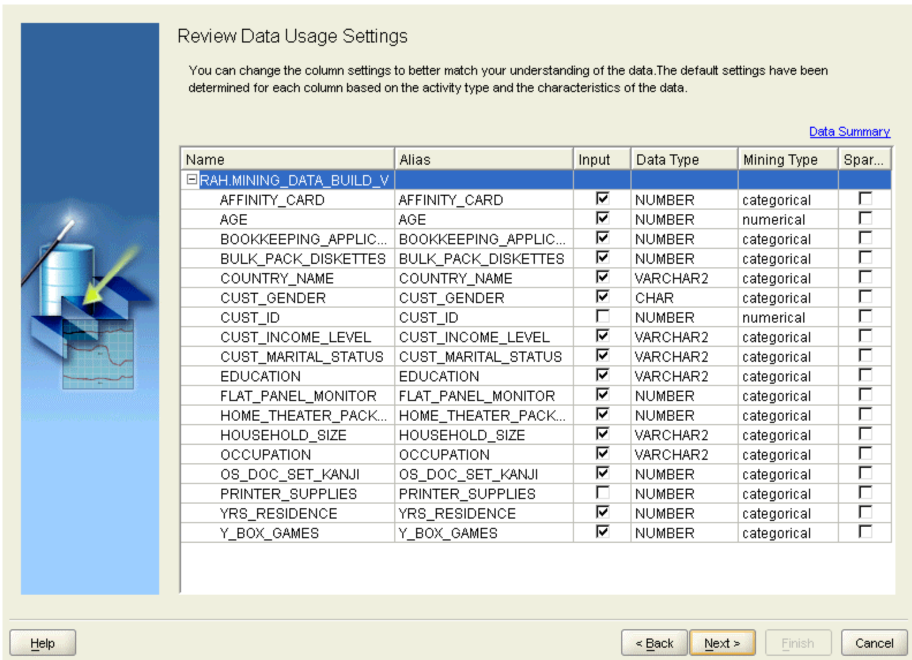
The goal is to segment the customers of the electronics store – select MINING_DATA_BUILD_V as the Case Table. You won't join in additional data – select CUST_ID as key and click Next to continue.



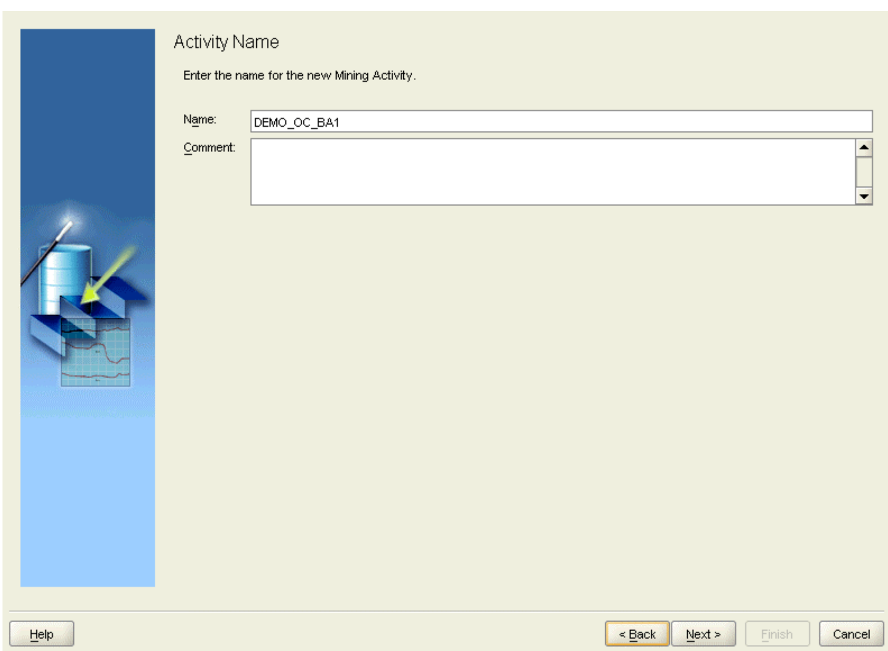
Review the data settings and

- Ensure that "continuous" integer attributes are numerical (for example, AGE)
- Ensure that binary integers are categorical

Click Next



Enter a descriptive name for the activity and click Next.



Click Advanced Settings on the final wizard page to view or modify the default settings. The Sample, Outlier Treatment, and Discretize settings have the same meanings as for Naïve Bayes and Support Vector Machines; refer to Chapters 5 and 8 for more details.

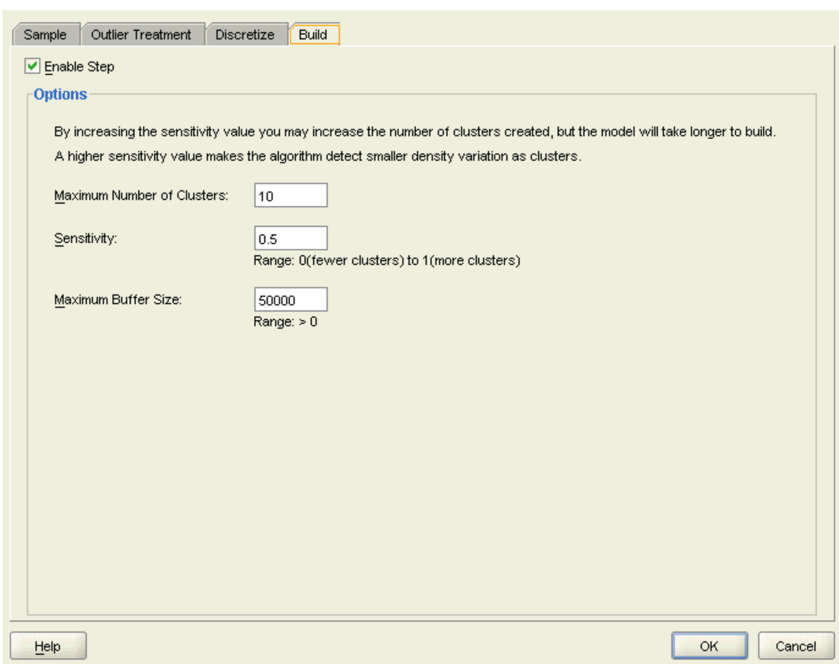
You can change the maximum number of clusters and the Sensitivity.

O-Cluster finds "natural" clusters by identifying areas of density within the data, up to the maximum number entered as a parameter. That is, the algorithm is not forced into defining a user-specified number of clusters, so the cluster membership is more clearly defined.

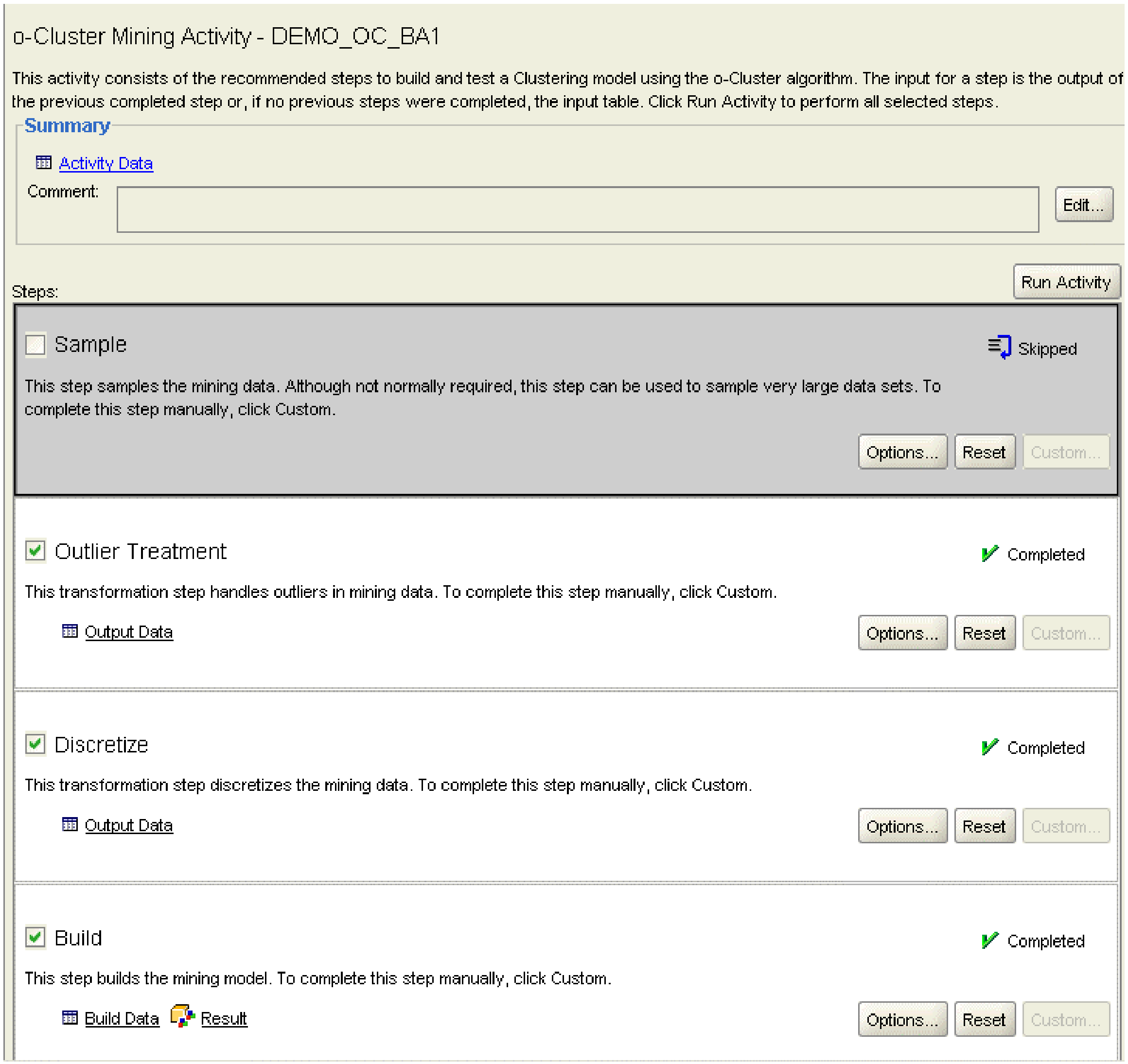
The Sensitivity setting determines how sensitive the algorithm is to differences in the characteristics of the population. O-cluster determines areas of density by looking for a "valley" separating two "hills" of density in the distribution curve of an attribute. A lower sensitivity requires a deeper valley; a higher sensitivity allows a shallow valley to define differences in density. Thus, a higher sensitivity value usually leads to a higher number of clusters.

If the build operation is very slow, you can increase the Maximum Buffer Size in an attempt to improve performance.

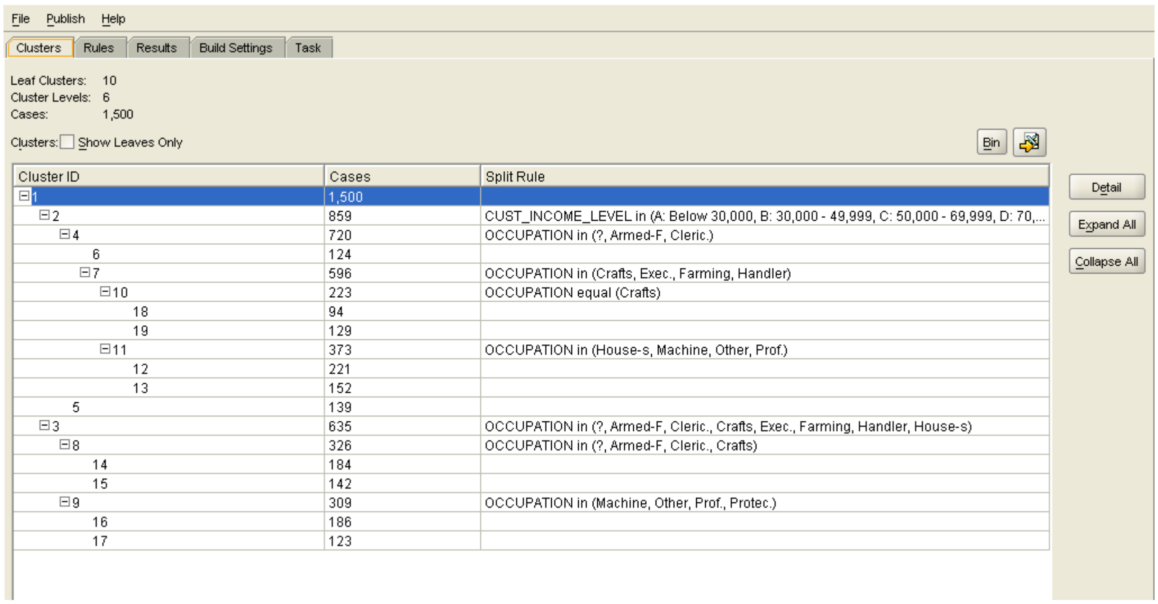
When done, click OK to return to the wizard final page, then Finish to launch the activity.



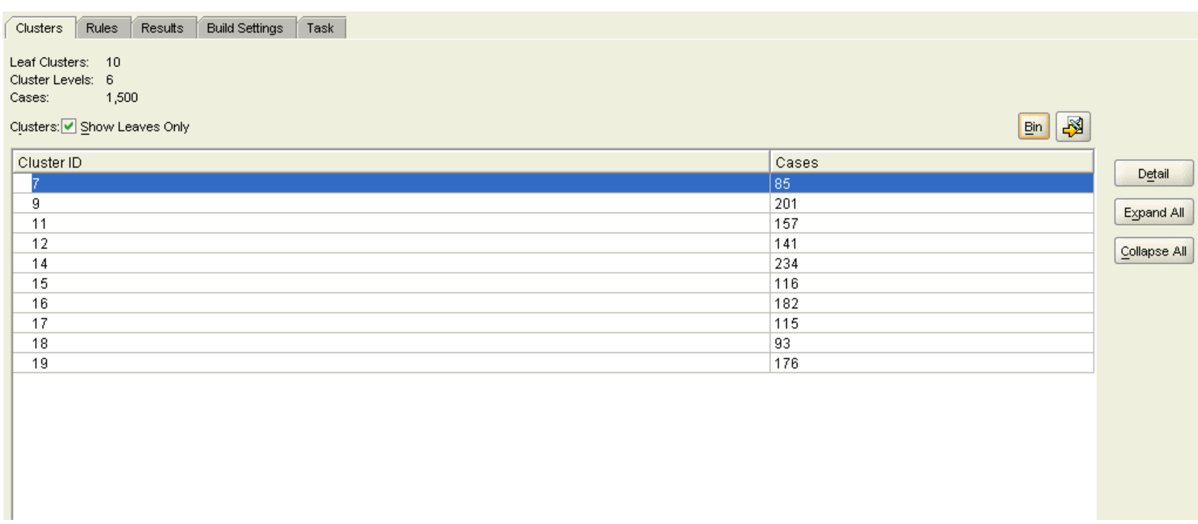
When the activity has completed, click Result in the Build step to investigate the model.



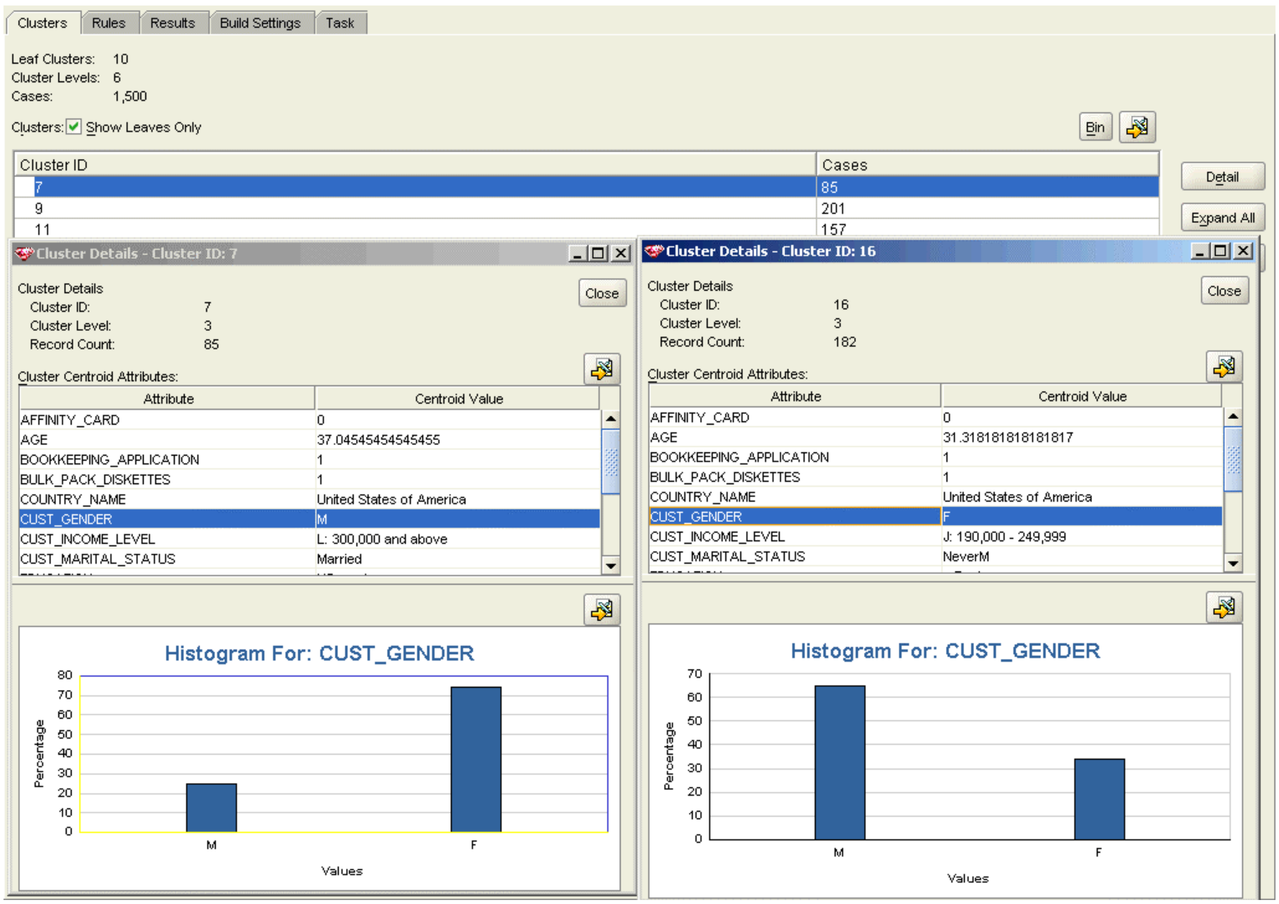
All clusters are shown in the first display, even intermediate clusters, so you can see how and why the segments were created in the iterative process. In the example shown below cluster 2 was created from cluster 1 (the entire population) based on Cust_Income_Level; cluster 4 was created from cluster 2 based on Occupation. (Your display may differ due to the small size of the dataset and the random sampling process)



To see the final clustering click the Show Leaves Only checkbox.



Highlight a cluster and click Detail to see a histogram of attributes for the members of the cluster. You can view the histograms for more than one cluster at a time, so you can compare the characteristics separating one cluster from another. In the example below (yours may differ), cluster 7 is predominantly Female while cluster 16 is predominantly Male. Note that the centroid does not necessarily indicate the value with the highest distribution; it is found by a calculation similar to that for a physical center of gravity.

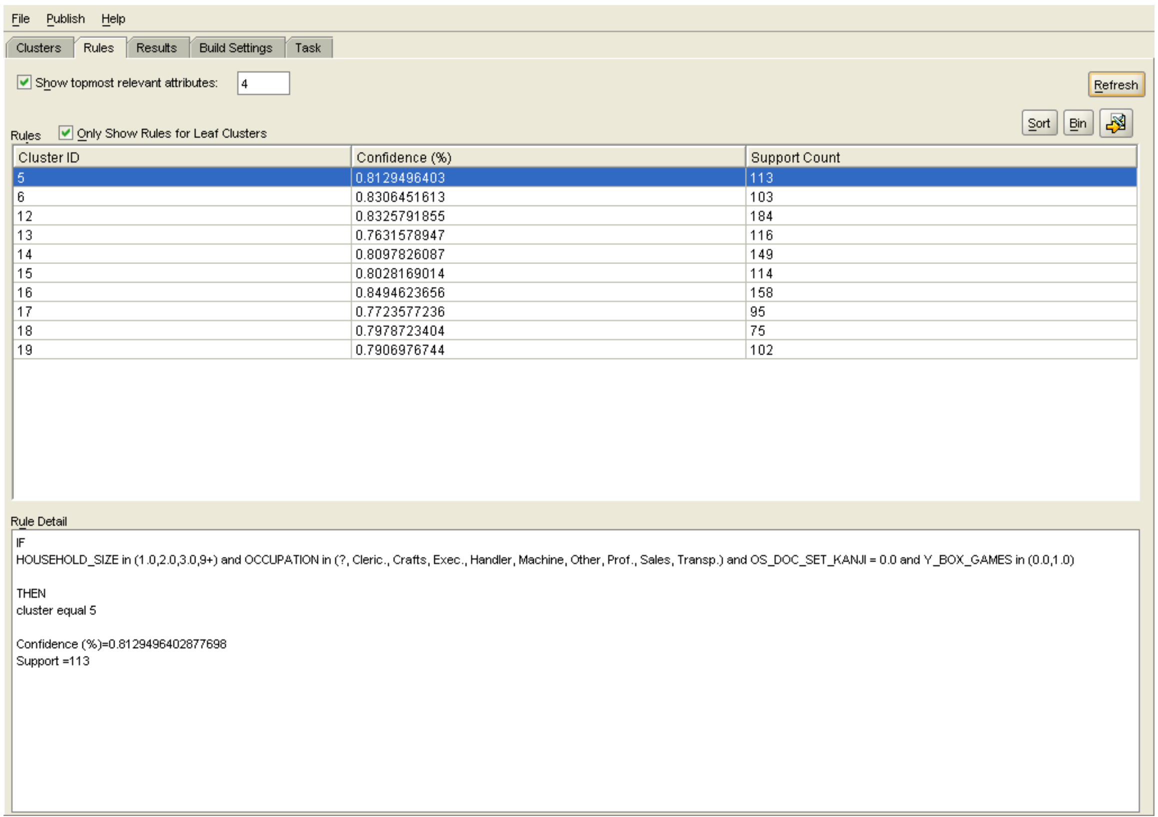


Click the Rules tab and highlight a cluster to see the rules defined by the cluster.

Click the checkbox Only Show Rules for Leaf Clusters to see the final clustering.

Click the checkbox Show Topmost Relevant Attributes to include only the most important factors in the rule for cluster membership.

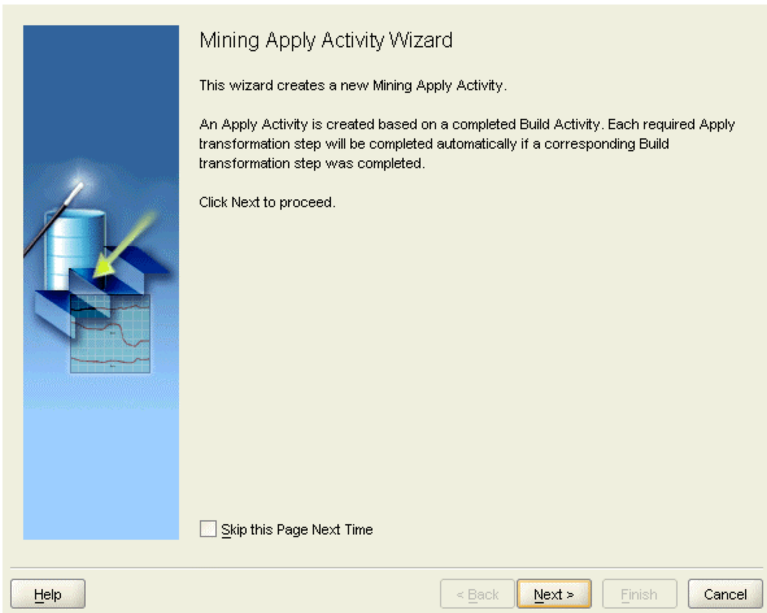
Confidence is a measure of the density of the cluster and Support is the number of cases from the input dataset determined to be in the cluster.



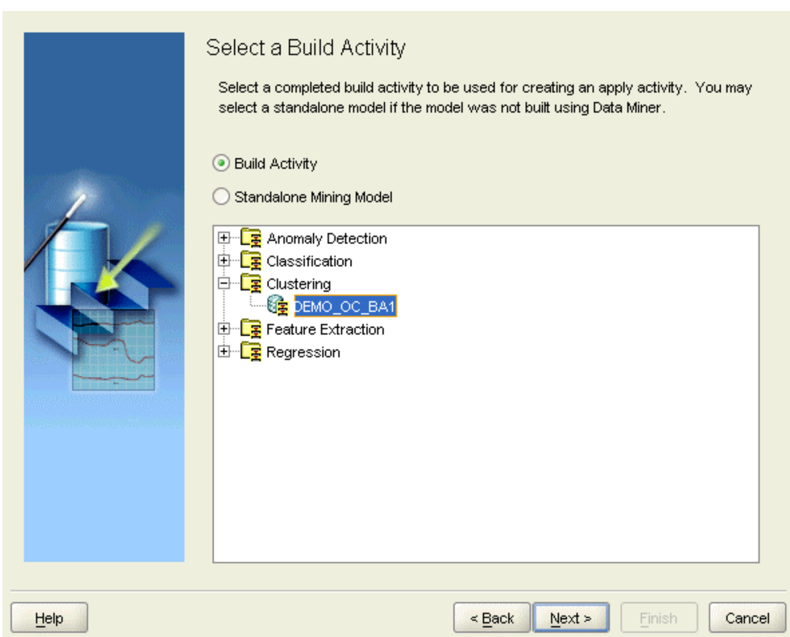
Applying a Clustering Model

Suppose you have created a clustering model that segments your customers into homogeneous groups. You can apply that model to new customers to determine likely segment membership.

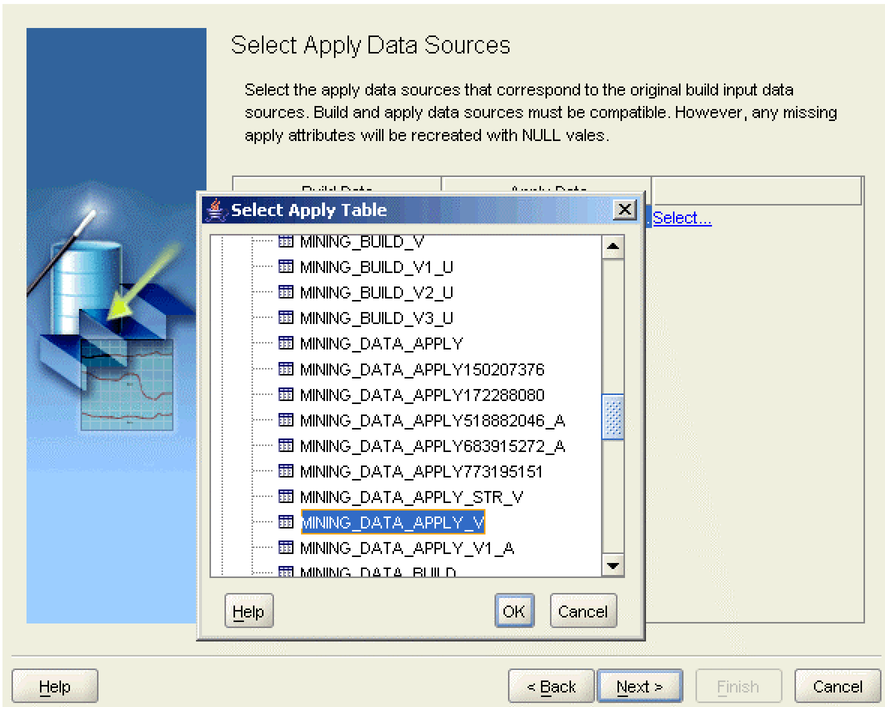
Choose Apply from the Activity pull-down menu and click Next.



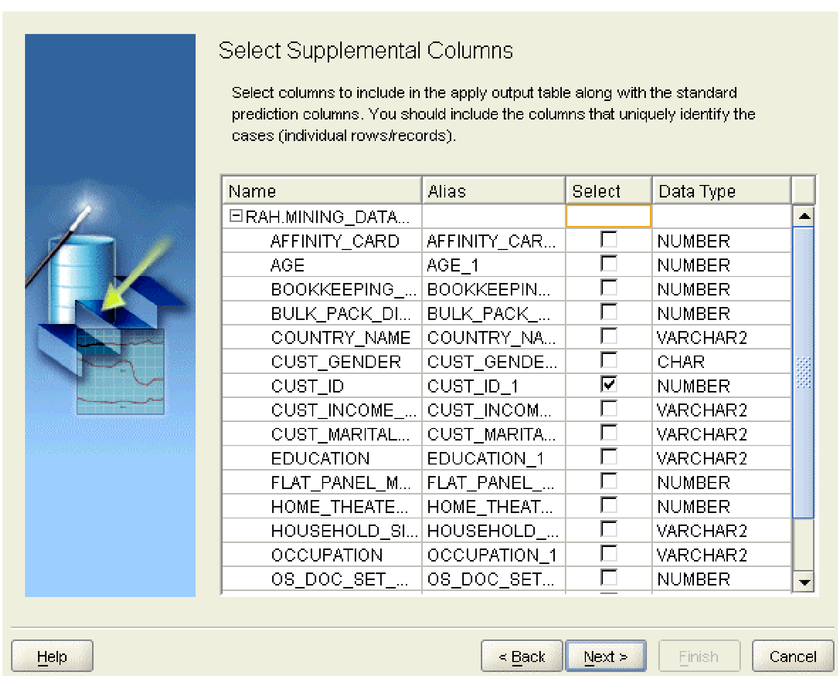
Select the Build activity that was used to create the clustering model; all metadata from the build steps will be passed to the Apply activity. Click Next.



Click Select and browse to the table/view that will be scored by the model. The input data must be in the same format as the case table in the build activity. Click Next.



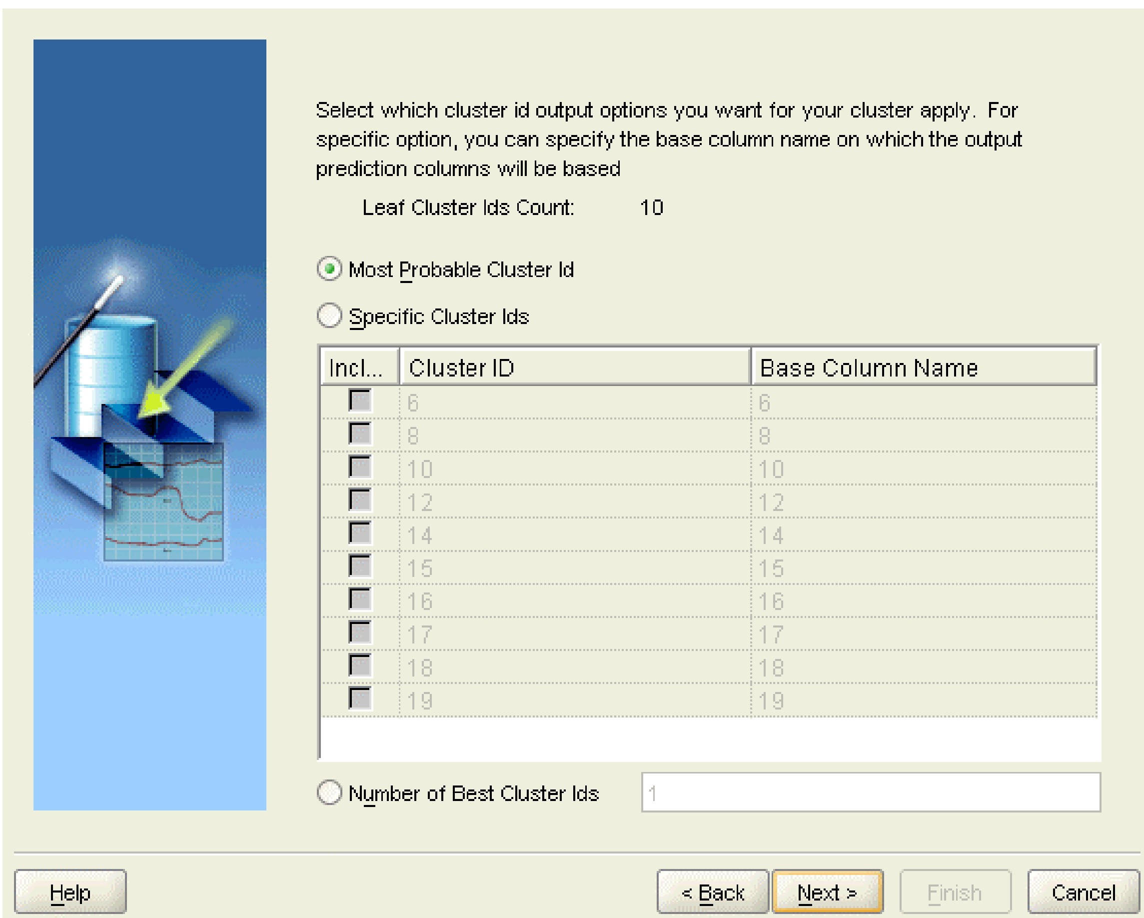
You will need an identifier for each record, and you can add other information such as name and phone number if available. Click Next to continue.



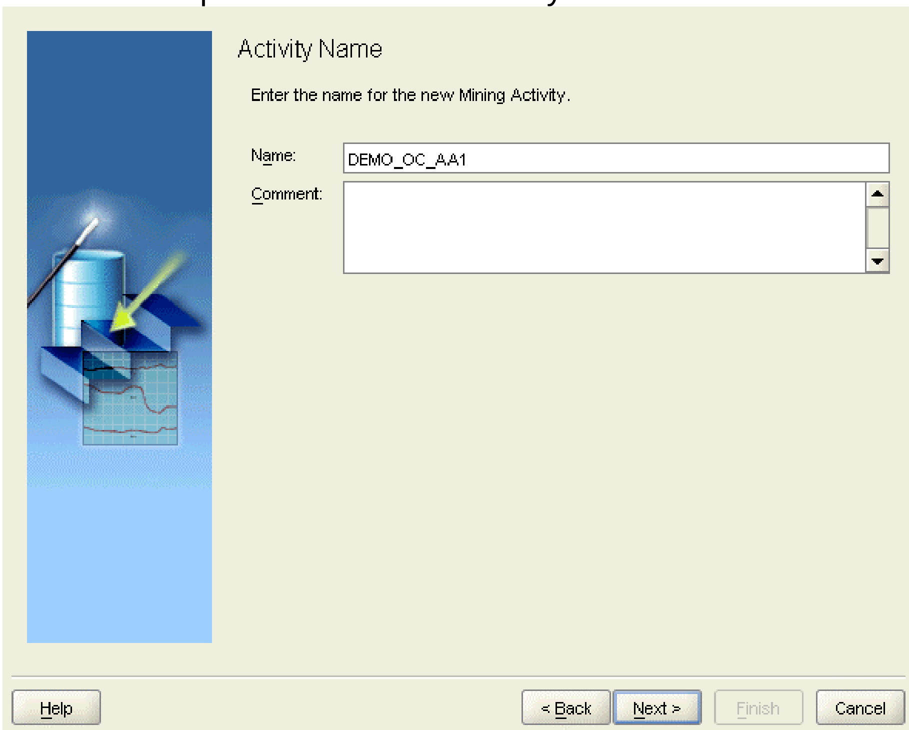
You have a choice of output information to be included in the records that are scored by the model.

The rules that were displayed in the model build activity are not necessarily exhaustive of the data space. For example, the model will be asked to associate a new customer with an existing segment, even if the new customer doesn't conform exactly to the rules for any cluster. Thus a probability of membership in each cluster is assigned to a record. You may want the apply output to show only the cluster with highest probability of membership for a given record (this is the default as shown below). Otherwise, you can include the probabilities for any number of clusters, either by specifying particular clusters, or by indicating the number of clusters ranked by probability. Check the appropriate radio button and click Next.

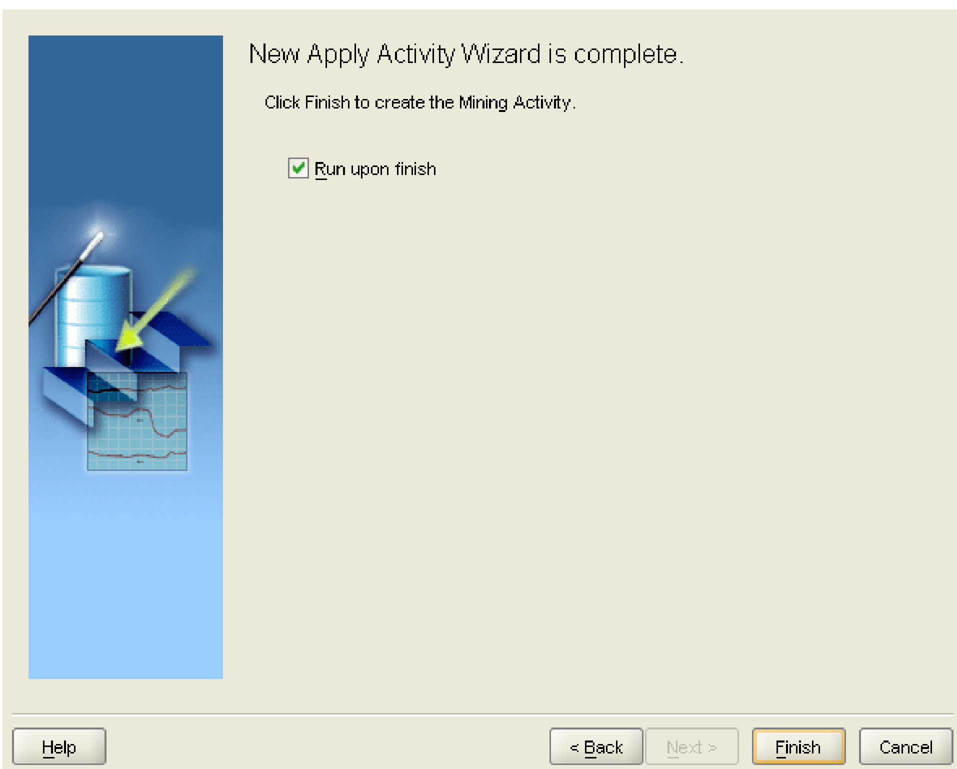
Both types of formats will be shown in the final display later in this chapter.



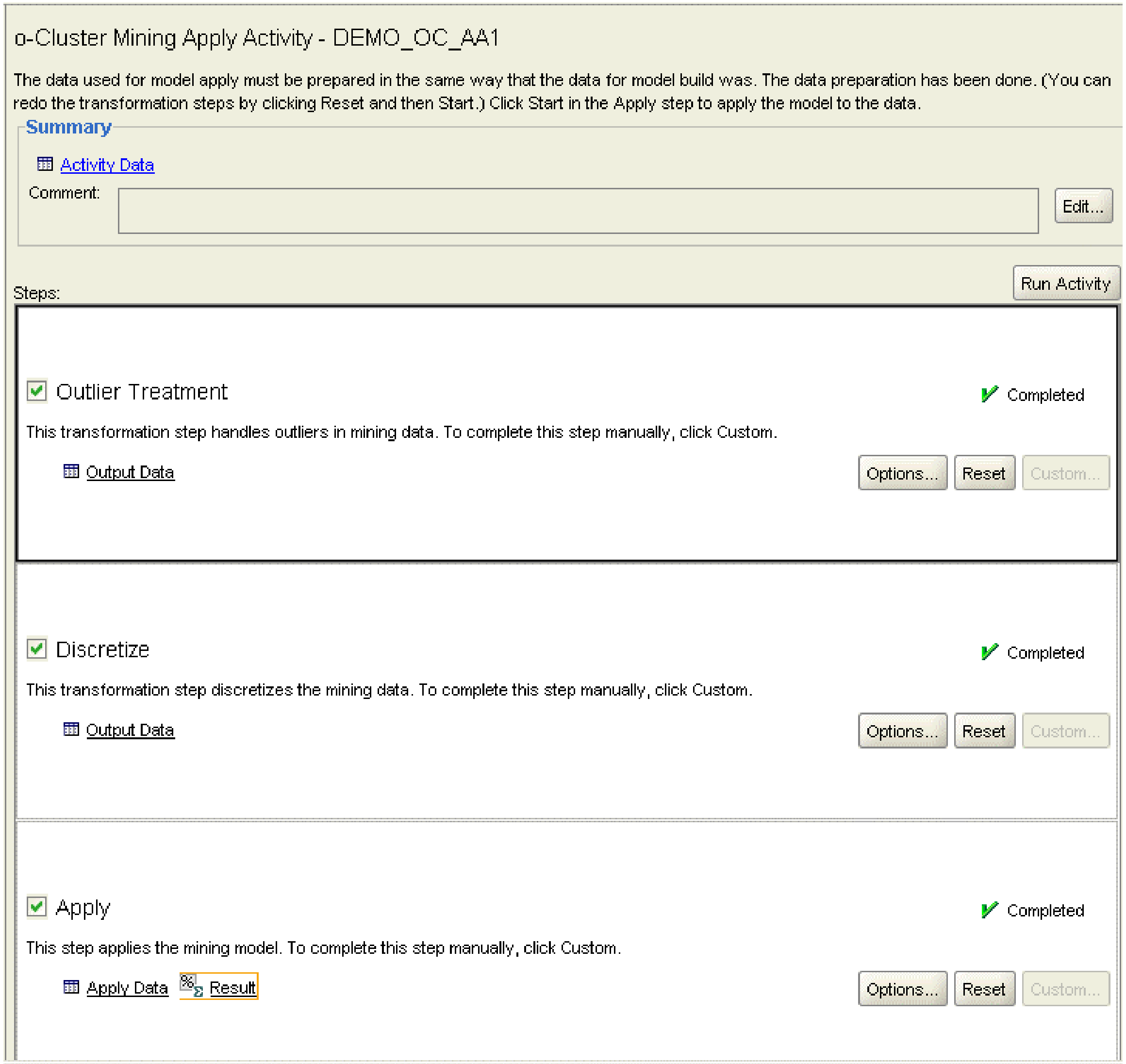
Enter a descriptive name for the activity and click next.



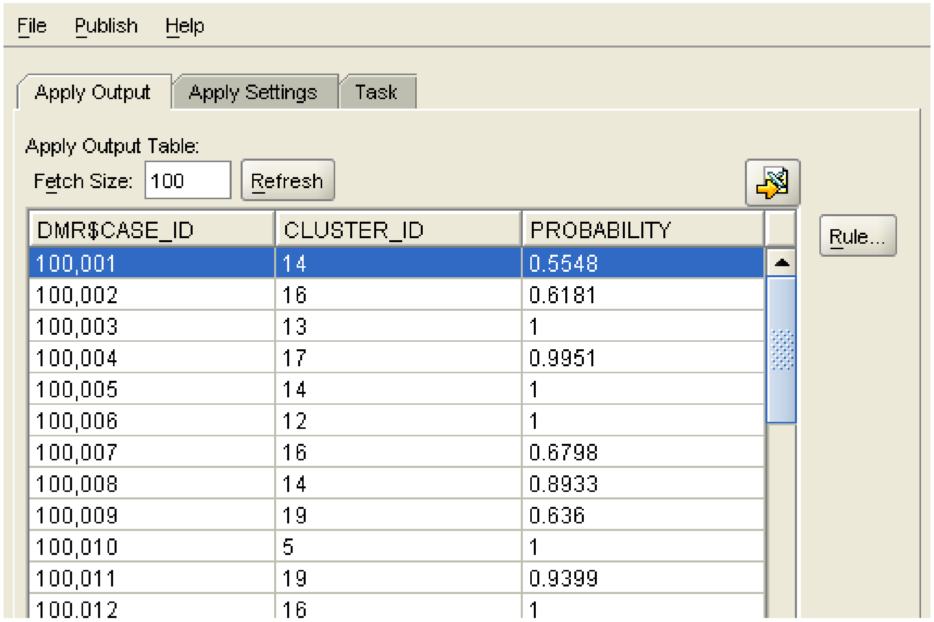
There are no Advanced Settings for the Apply activity. Click Finish to launch the activity.



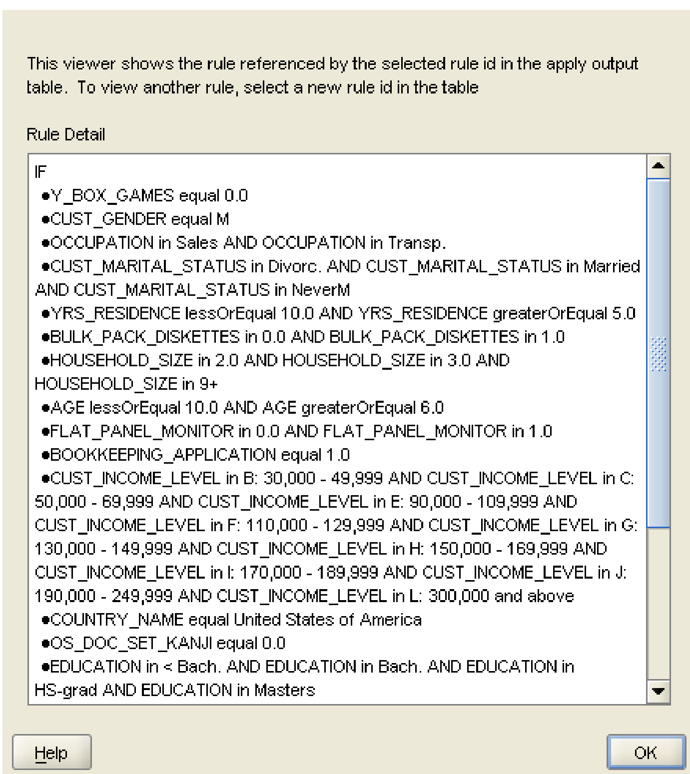
When the activity has completed, click Result in the Apply step to view the output.



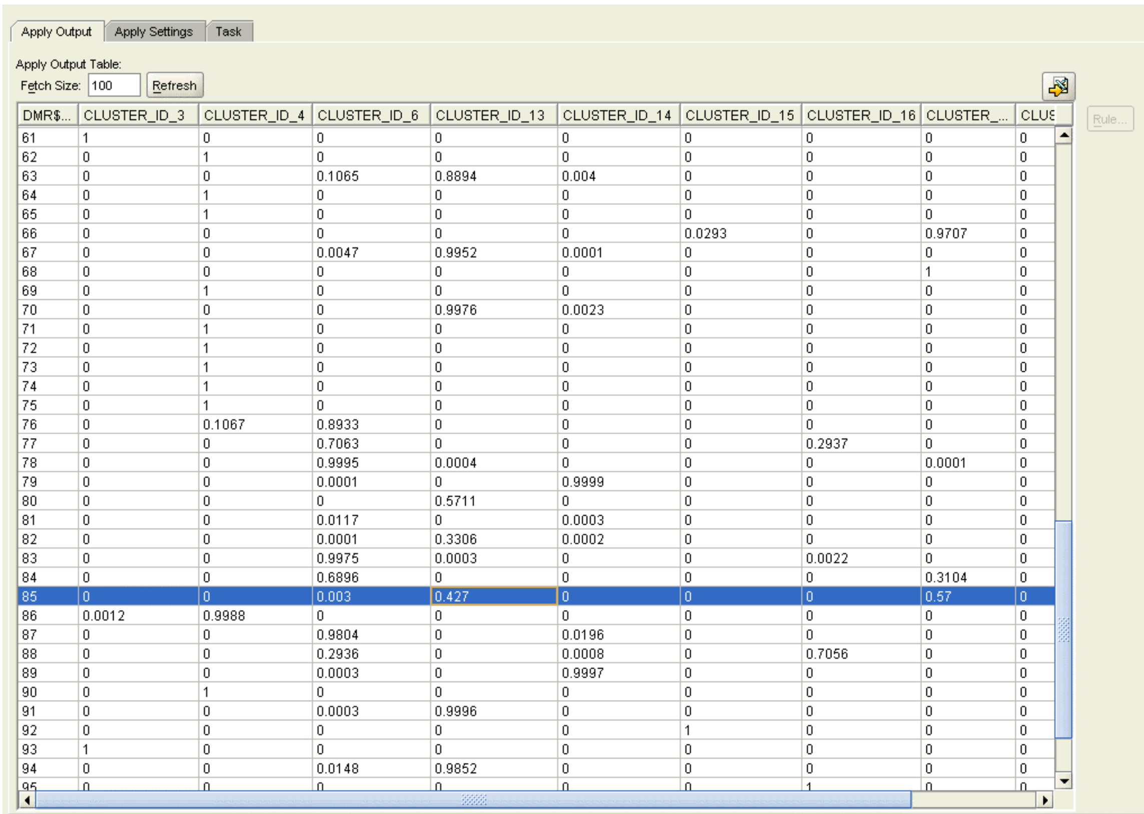
If you chose to include only the most likely cluster membership for each record, you see a display as below:



You can highlight a record and click Rule to see the cluster definition for the cluster with the highest probability. As noted above, the record may not conform exactly to the rule.



If you chose to specify by Cluster_ID more than one cluster in each record, you will see a result like the one below. Note that in some cases, as in the record highlighted below, there is not a very high probability that the record belongs in any defined cluster. This may be an example of a rare or unusual case.



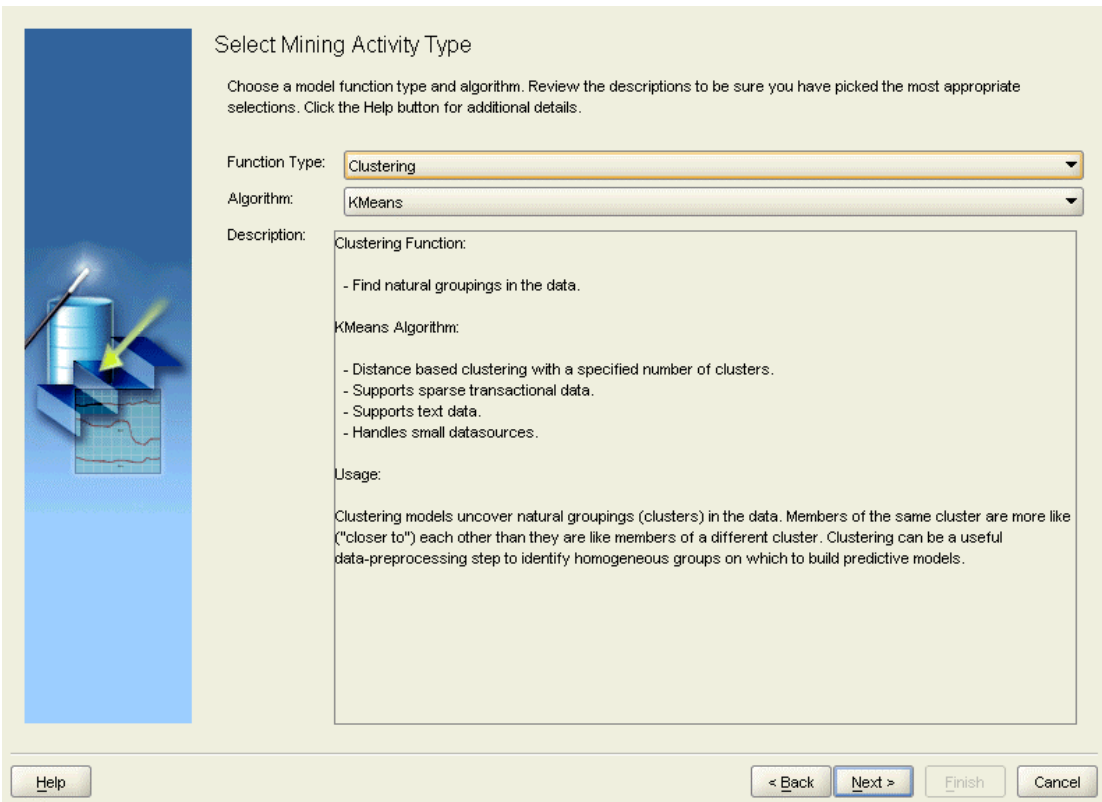
DMR$...	CLUSTER_ID_3	CLUSTER_ID_4	CLUSTER_ID_6	CLUSTER_ID_13	CLUSTER_ID_14	CLUSTER_ID_15	CLUSTER_ID_16	CLUSTER_...	CLUS
61	1	0	0	0	0	0	0	0	0
62	0	1	0	0	0	0	0	0	0
63	0	0	0.1065	0.8894	0.004	0	0	0	0
64	0	1	0	0	0	0	0	0	0
65	0	1	0	0	0	0	0	0	0
66	0	0	0	0	0	0.0293	0	0.9707	0
67	0	0	0.0047	0.9952	0.0001	0	0	0	0
68	0	0	0	0	0	0	0	1	0
69	0	1	0	0	0	0	0	0	0
70	0	0	0	0.9976	0.0023	0	0	0	0
71	0	1	0	0	0	0	0	0	0
72	0	1	0	0	0	0	0	0	0
73	0	1	0	0	0	0	0	0	0
74	0	1	0	0	0	0	0	0	0
75	0	1	0	0	0	0	0	0	0
76	0	0.1067	0.8933	0	0	0	0	0	0
77	0	0	0.7063	0	0	0	0.2937	0	0
78	0	0	0.9995	0.0004	0	0	0	0.0001	0
79	0	0	0.0001	0	0.9999	0	0	0	0
80	0	0	0	0.5711	0	0	0	0	0
81	0	0	0.0117	0	0.0003	0	0	0	0
82	0	0	0.0001	0.3306	0.0002	0	0	0	0
83	0	0	0.9975	0.0003	0	0	0.0022	0	0
84	0	0	0.6896	0	0	0	0	0.3104	0
85	0	0	0.003	0.427	0	0	0	0.57	0
86	0.0012	0.9988	0	0	0	0	0	0	0
87	0	0	0.9804	0	0.0196	0	0	0	0
88	0	0	0.2936	0	0.0008	0	0.7056	0	0
89	0	0	0.0003	0	0.9997	0	0	0	0
90	0	1	0	0	0	0	0	0	0
91	0	0	0.0003	0.9996	0	0	0	0	0
92	0	0	0	0	0	1	0	0	0
93	1	0	0	0	0	0	0	0	0
94	0	0	0.0148	0.9852	0	0	0	0	0
95	0	0	0	0	0	0	1	0	0

Chapter 11 – Clustering: k-Means

Clustering is used to identify distinct segments of a population and to explain the common characteristics of members of a cluster, and also to determine what distinguishes members of one cluster from members of another cluster.

ODM provides two Clustering algorithms, Enhanced k-means and O-cluster; this chapter will discuss k-means.

Choose Build from the Activity pull-down menu, then select Clustering as the Function Type and KMeans as the algorithm, Click Next.



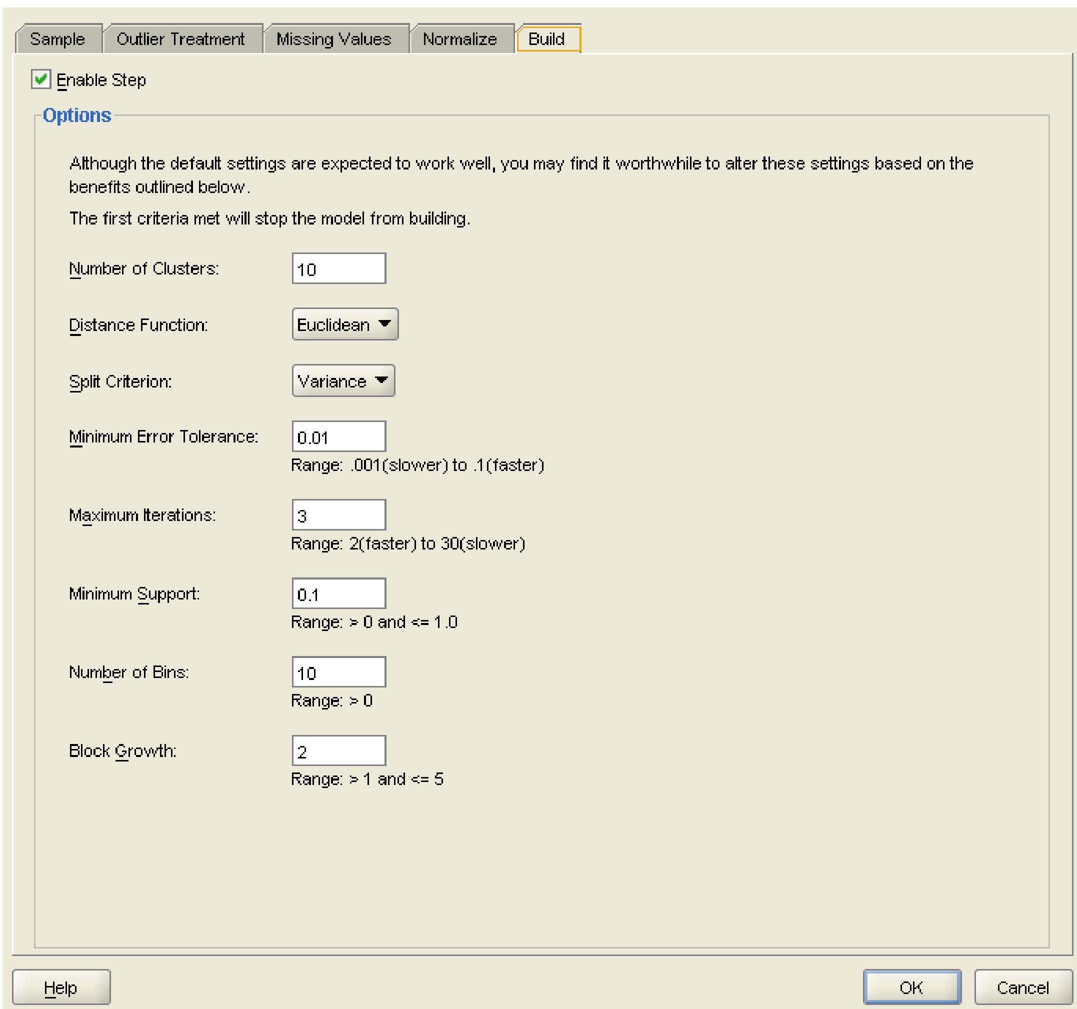
The goal is to segment the customers of the electronics store – select MINING_DATA_BUILD_V as the Case Table. You won't join in additional data – select CUST_ID as key and click Next to continue.

The wizard steps are identical to those for O-cluster until the final step. Refer to Chapter 10 for more details.

On the final wizard page, click Advanced Settings to view or modify default values. All settings except Build have the same meaning as for O-cluster. See Chapter 10 for explanations.

K-means uses a distance metric to define the clusters, based on the centroid (center of gravity) of each cluster. When a new cluster is split from an existing one (that is, a new centroid is defined), each record is assigned to the cluster whose centroid is closest to the record. ODM's version of k-means goes beyond the classical implementation by defining a hierarchical parent-child relationship of clusters.

The parameter settings are explained below the screen display.



The k-means algorithm creates the number of clusters specified by the user (except in the unusual case in which the number of records is less than the number of requested clusters).

There are two distance metrics: Euclidean (default) and Cosine (appropriate if the data has been normalized).

There are two methods of determining which cluster to split to get a new one: Variance (default – split the cluster that produces the largest variance; that is a new cluster that is most different from the original) and Size (split the largest).

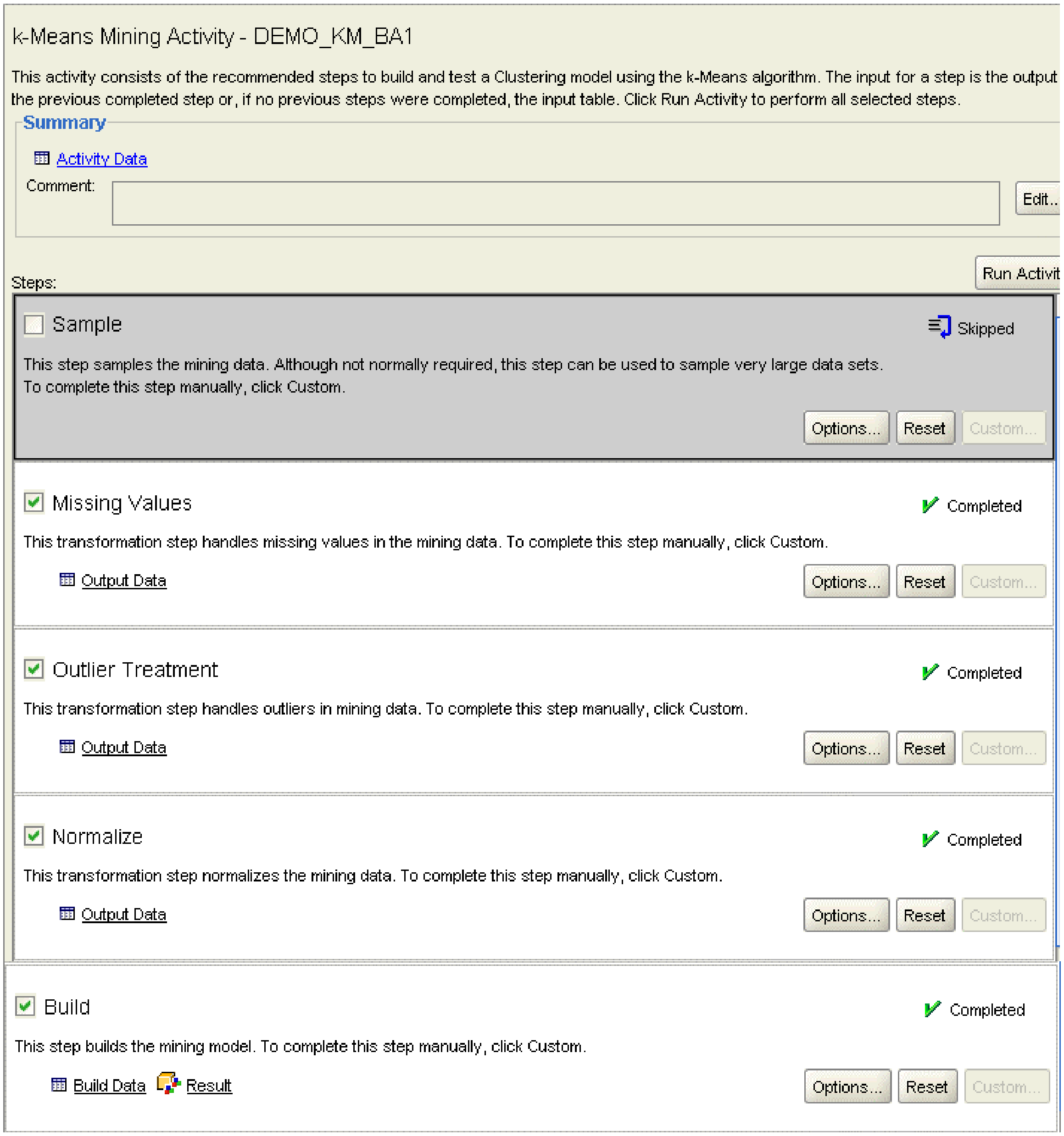
Minimum Error Tolerance and Maximum Iterations determine how the parent-child hierarchy of clusters is formed. Increasing the tolerance or lowering the iteration maximum will cause the model to be built faster, but possibly with more poorly-defined clusters.

Minimum Support applies to an individual attribute: the attribute will be included in the rule describing a cluster only if the number of non-NULL values for that cluster in the Build data exceeds the fraction entered.

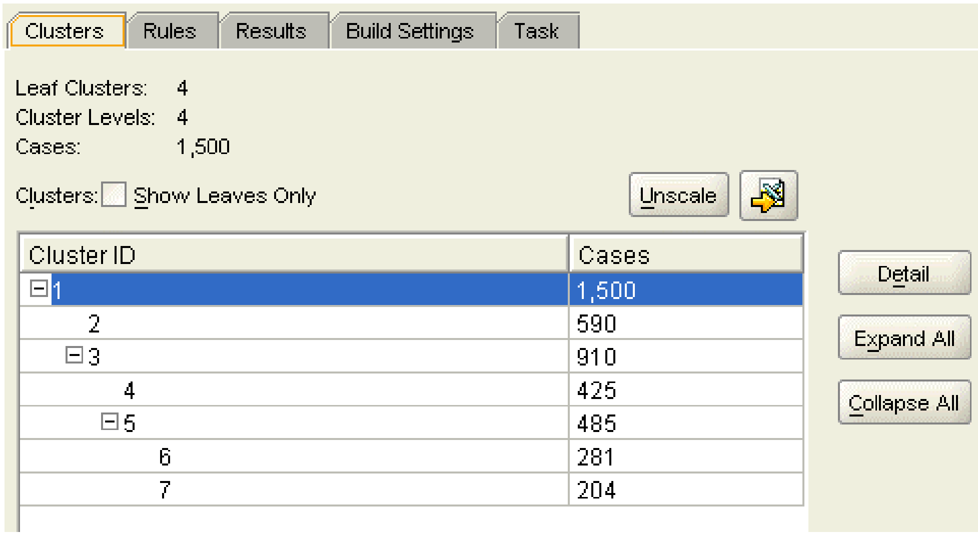
Number of Bins is the number of bars shown in the cluster histogram for each attribute.

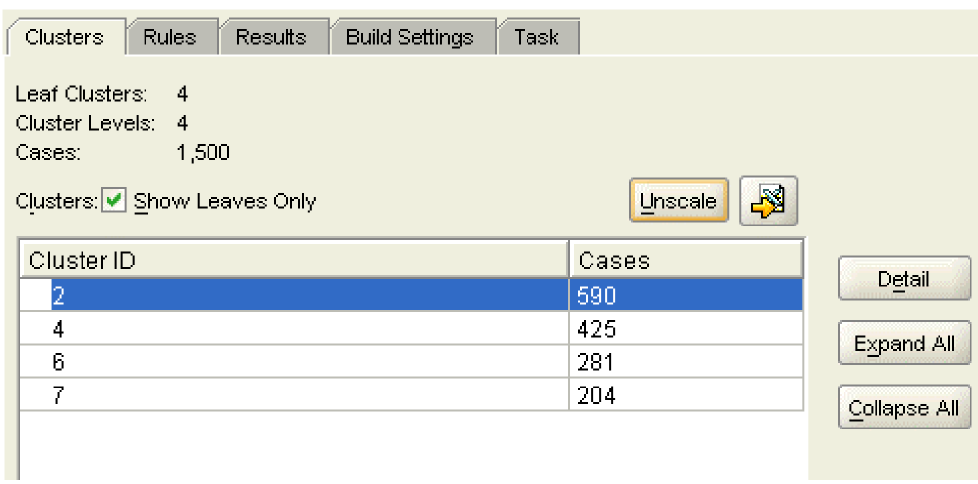
Block Growth is a factor related to memory usage during the Build process.

When the activity has completed, you can click Result in the Build step to see the model.

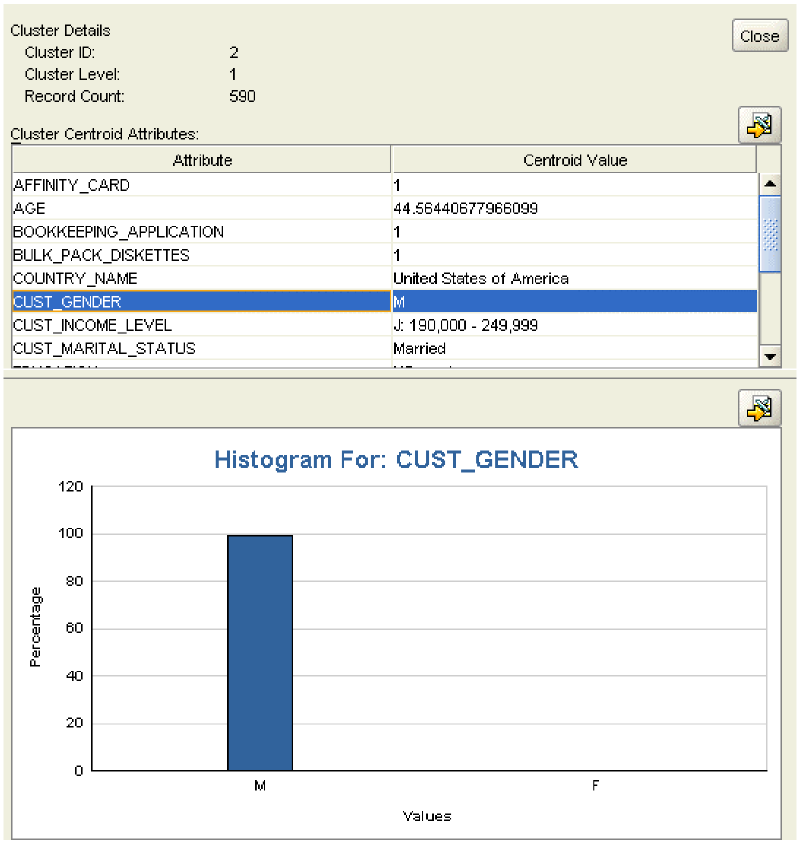


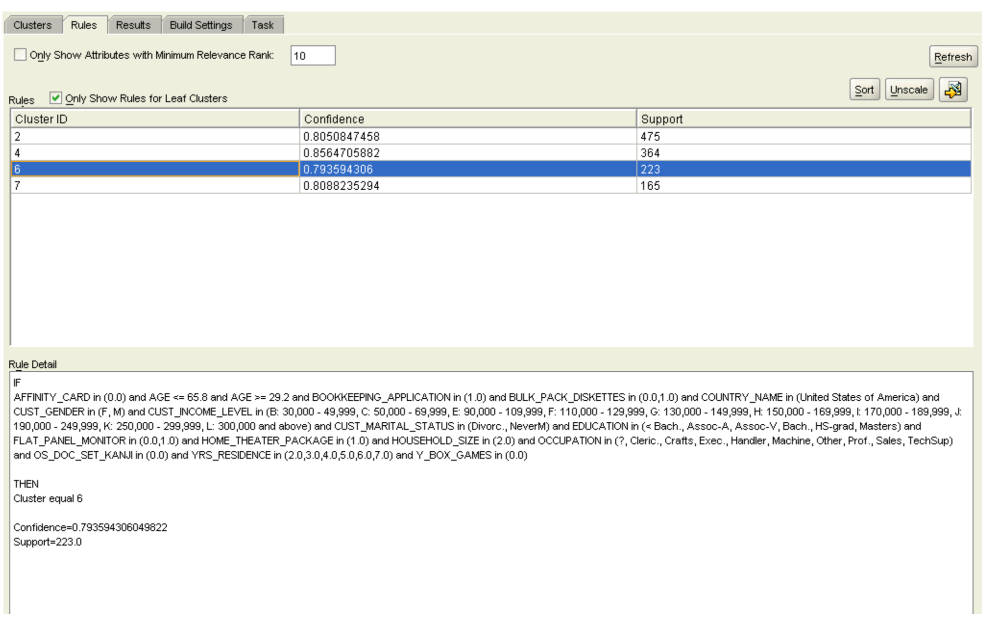
As with the O-cluster model, you can view all splits and also the final clustering.





You can click a cluster and view histograms of its attributes, and you can click the Rules tab to display the rules for a highlighted cluster.





Chapter 12 – Anomaly Detection

Normally, the building of a Classification model requires existing data containing sufficiently many cases in each class. For example, a model predicting high-value versus low-value customers must be built using data containing records of both types of customers who have been determined (by some business rule) to be either high or low value customers.

However, in some cases only one class of individuals has been defined, or one class is extremely rare.

Some examples are:

1. An automobile retailer knows purchasing, financial, and demographic information about people who have bought cars, but nothing about those who have not bought cars. How can potential car buyers be identified?

2. A law enforcement agency compiles many facts about illegal activities, but nothing about legitimate activities. How can suspicious activity be flagged?

3. A taxing authority processes millions of tax forms knowing that a very small number are submitted by tax cheats. How can the cheaters be found?

In the first two cases only one class is known; in the third case, the abnormal records may have been identified by manual means, but there are so few of them that a “normal” classification model cannot be built.

If there are enough of the “rare” records that a stratified sample can be created that is sufficiently rich in information to build a classification model, then the classification model should be built.

It is important to note that solving a “One-class” classification problem is difficult, and the goal of Anomaly Detection is to provide some useful information where no information was previously attainable.

The goal is to create a “profile” of the known class, and to apply that profile to the general population for the purpose of identifying individuals who are “different” from the profile in some way.

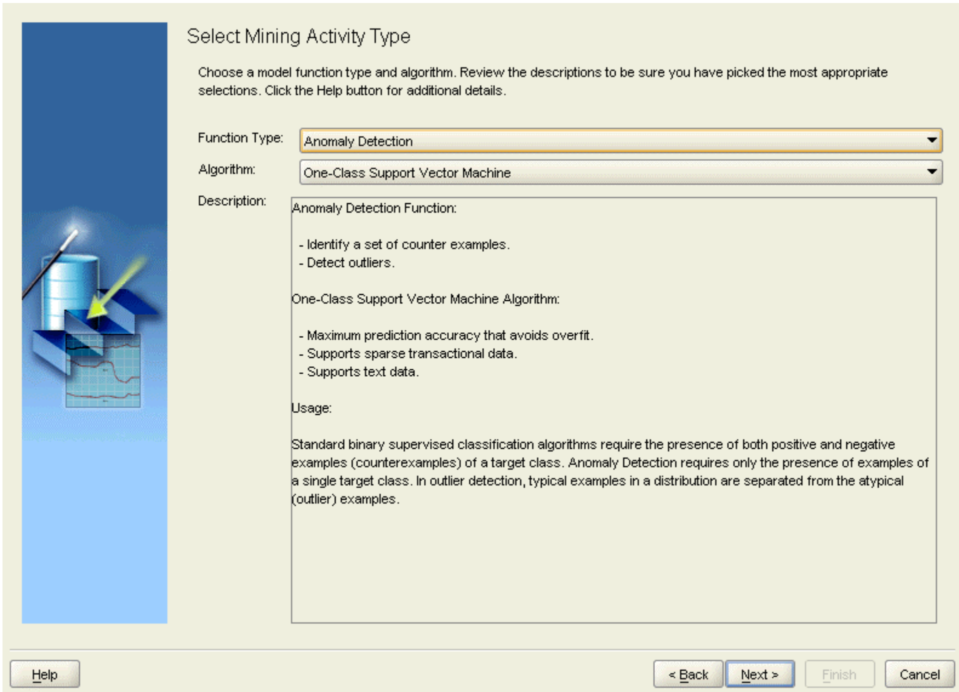
The dataset that will be used to illustrate the methodology has been derived from the marketing data used in the Classification examples. The tables are contained in the dump file in the Supplemental_Data file packaged with this tutorial.

Attributes

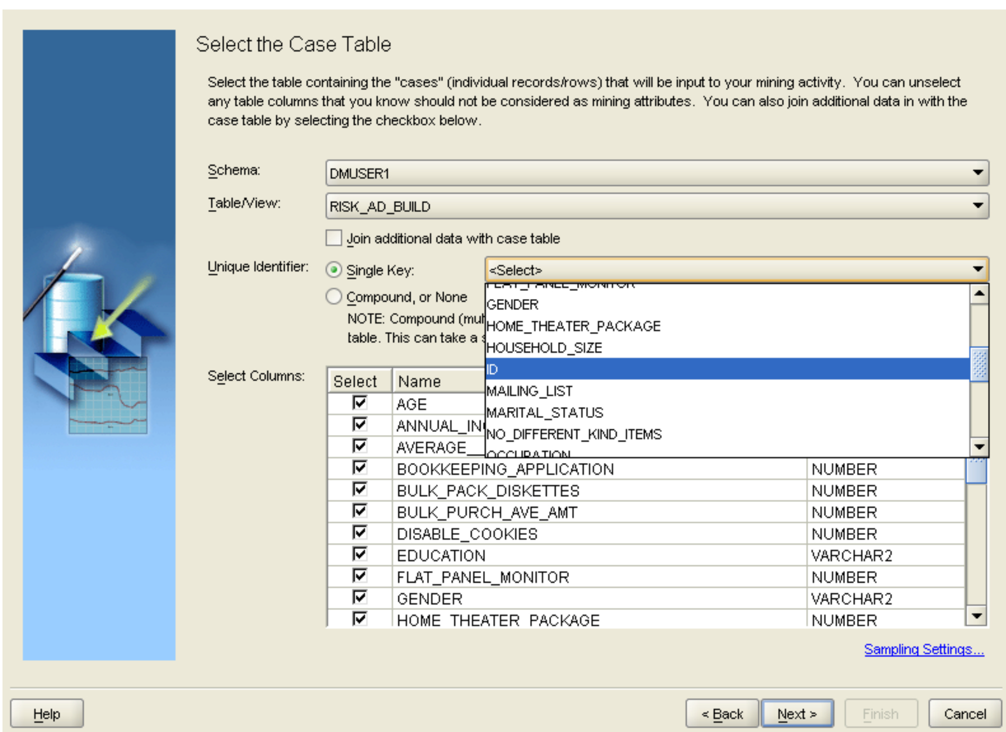
PK	Name	Type	Size
	WORKCLASS	VARCHAR2	21
	EDUCATION	VARCHAR2	21
	MARITAL_STATUS	VARCHAR2	21
	OCCUPATION	VARCHAR2	21
	HOUSEHOLD_SIZE	VARCHAR2	21
	TOP_REASON_FOR_...	VARCHAR2	21
	GENDER	VARCHAR2	18
	SHIPPING_ADDRESS_...	VARCHAR2	21
	AGE	NUMBER	22
	ANNUAL_INCOME	NUMBER	22
	WKS_SINCE_LAST_P...	NUMBER	22
	AVERAGE___ITEMS_...	NUMBER	22
	NO_DIFFERENT_KIND...	NUMBER	22
	BULK_PURCH_AVE_...	NUMBER	22
	YRS_RESIDENCE	NUMBER	22
	DISABLE_COOKIES	NUMBER	22
	PROMO_RESPOND	NUMBER	22
	MAILING_LIST	NUMBER	22
	SR_CITIZEN	NUMBER	22
	BULK_PACK_DISKET...	NUMBER	22
	FLAT_PANEL_MONIT...	NUMBER	22
	HOME_THEATER_PA...	NUMBER	22
	BOOKKEEPING_APPLI...	NUMBER	22
	PRINTER_SUPPLIES	NUMBER	22
	Y_BOX_GAMES	NUMBER	22
	OS_DOC_SET_KANJI	NUMBER	22
	PETS	NUMBER	22
	ID	NUMBER	10
	RISK	NUMBER	22

The AFFINITY_CARD column that was used as the target in the marketing examples has been changed to RISK, representing “suspicious” cases. The data has been transformed so that the build data RISK_AD_BUILD consists only of records with RISK = 0, representing no risk. The test data RISK_AD_TEST has a few records with RISK = 1, representing the unusual cases that the Anomaly Detection model will try to find.

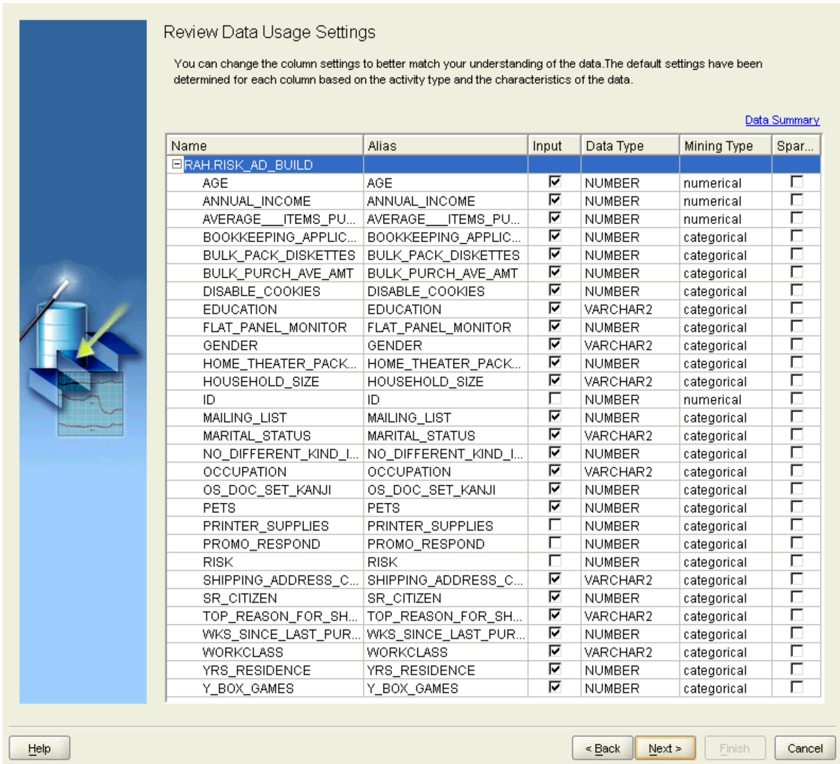
Choose Build on the Activity pull-down menu and select Anomaly Detection (there is only one algorithm: SVM). Click Next.



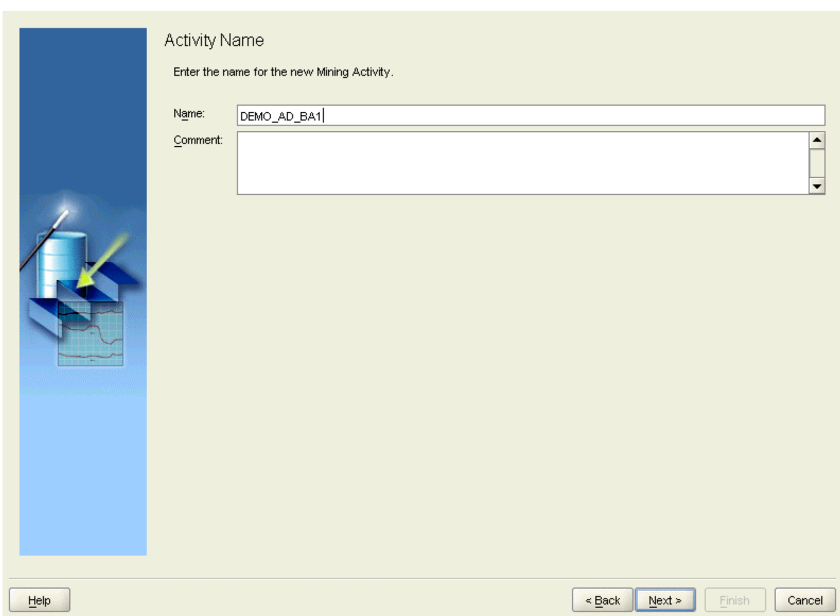
The modified source data RISK_AD_BUILD is selected, and the attribute ID is designated as the unique identifier. Click Next



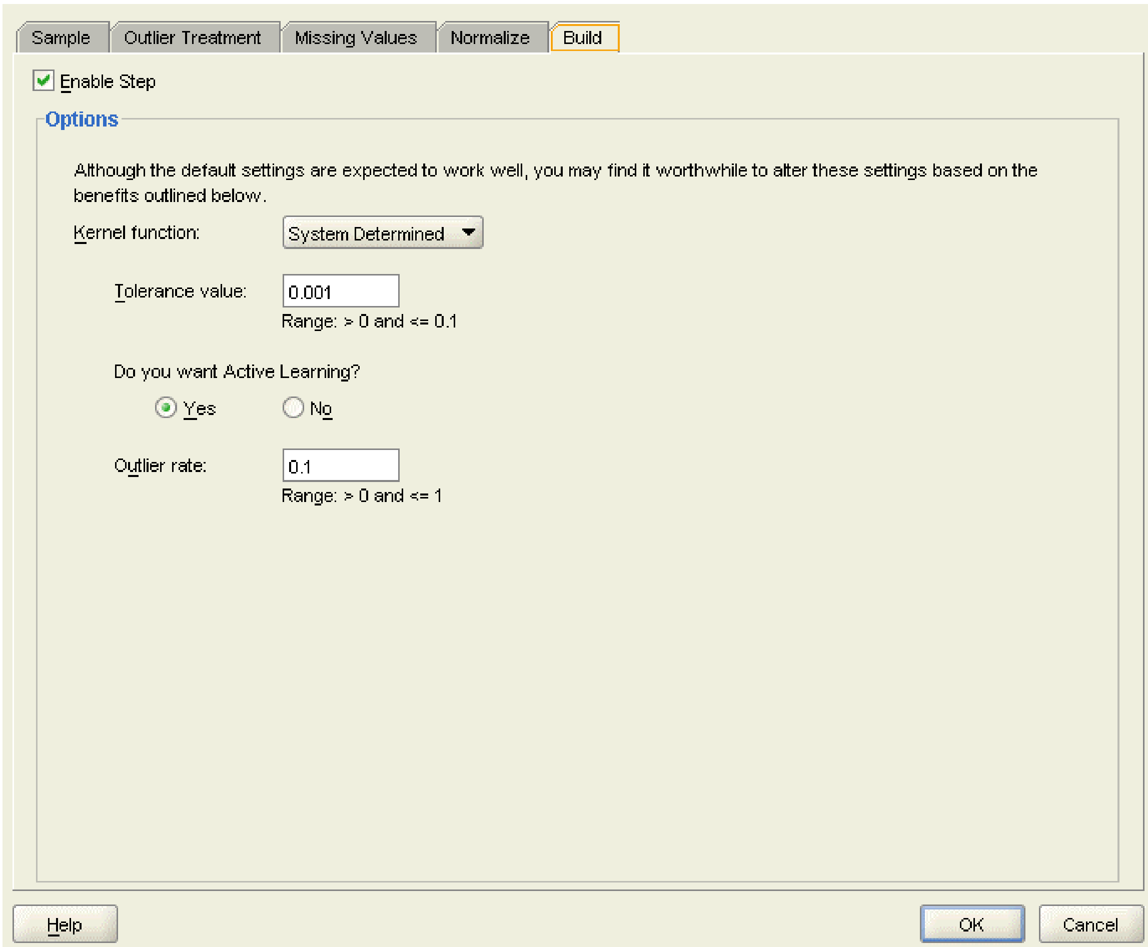
Notice that RISK has been automatically eliminated from the Build process because it has the constant value 0. Click Next.



Enter a descriptive name for the activity and click Next to proceed to the final wizard page.

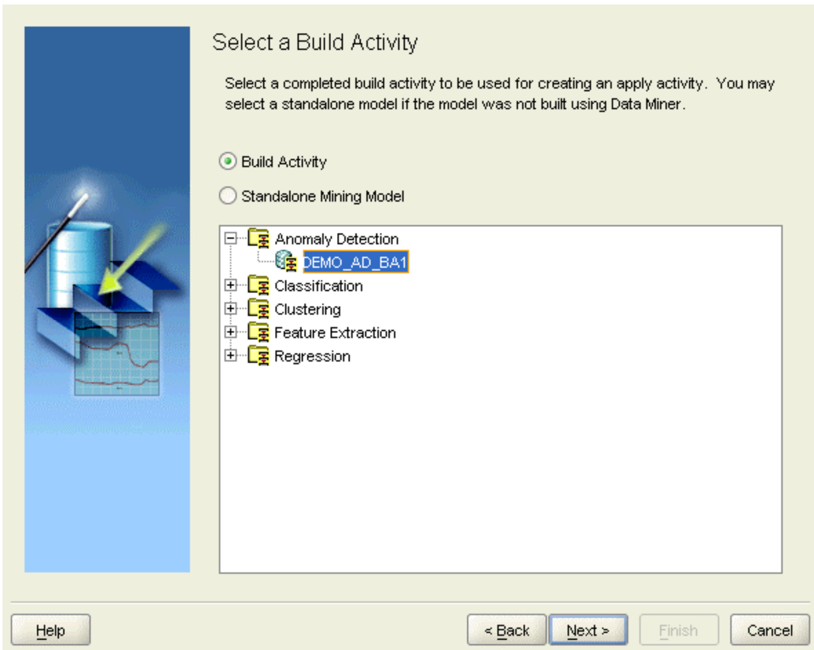


Click Advanced Settings on the final page to view or modify the default settings. The data preparation settings have the same meaning as for SVM Classification; see Chapter 8 for further discussion. The default Build setting lets the algorithm choose the kernel type; you may specify Linear or Gaussian.

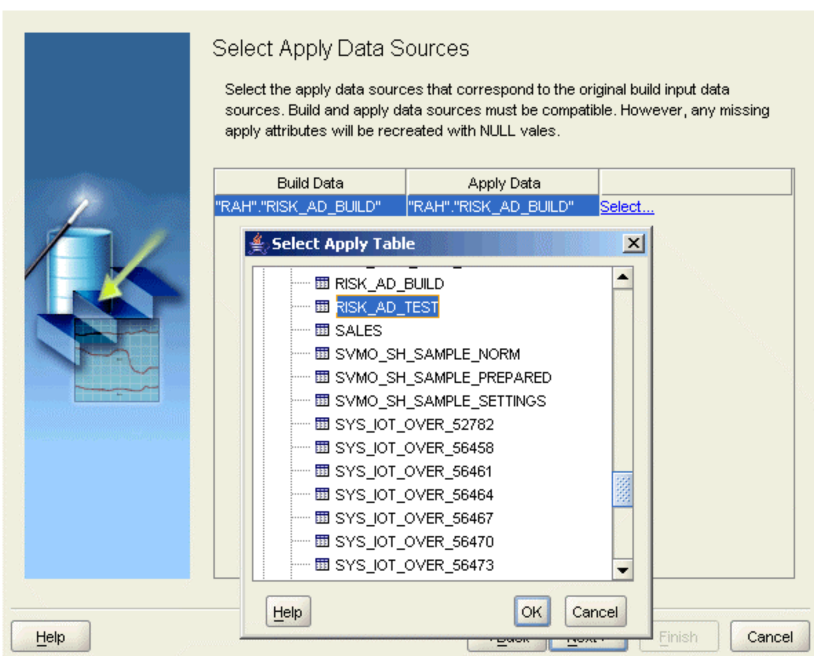


Tolerance Value and Active Learning have the same meaning as for SVM Classification; refer to Chapter 8 for more detail. However, a new parameter not seen in previous SVM examples is exposed – Outlier Rate. If you have some knowledge that the number of “suspicious” cases is a certain percentage of your population, you can set the Outlier Rate to that percentage, and the model will identify approximately that many “rare” cases when applied to the general population. The default is 10%; this may be high for most Anomaly Detection problems, but you may want to see the initial results before modifying this value. Click OK to return to the wizard and click Next to execute the activity.

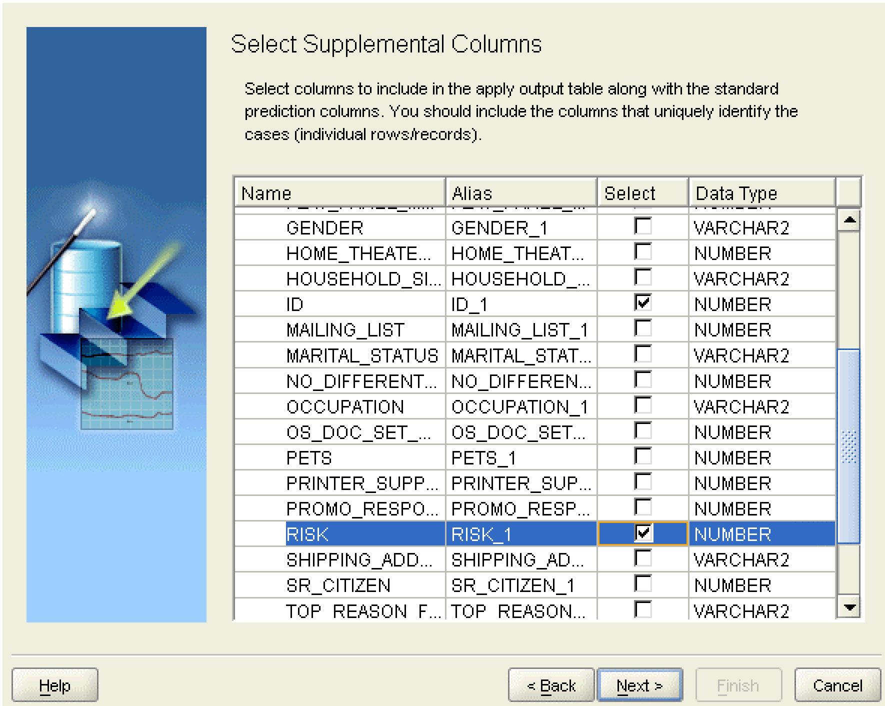
When the activity has been completed, you can test the model if you have holdout data containing some known “rare” cases. The RISK_AD_TEST data has such rare cases, so an Apply activity will use this data and compare the model’s predictions of “suspicious” to the known cases. Choose Apply from the Activity pull-down menu and highlight the Anomaly Detection Build activity. Click Next.



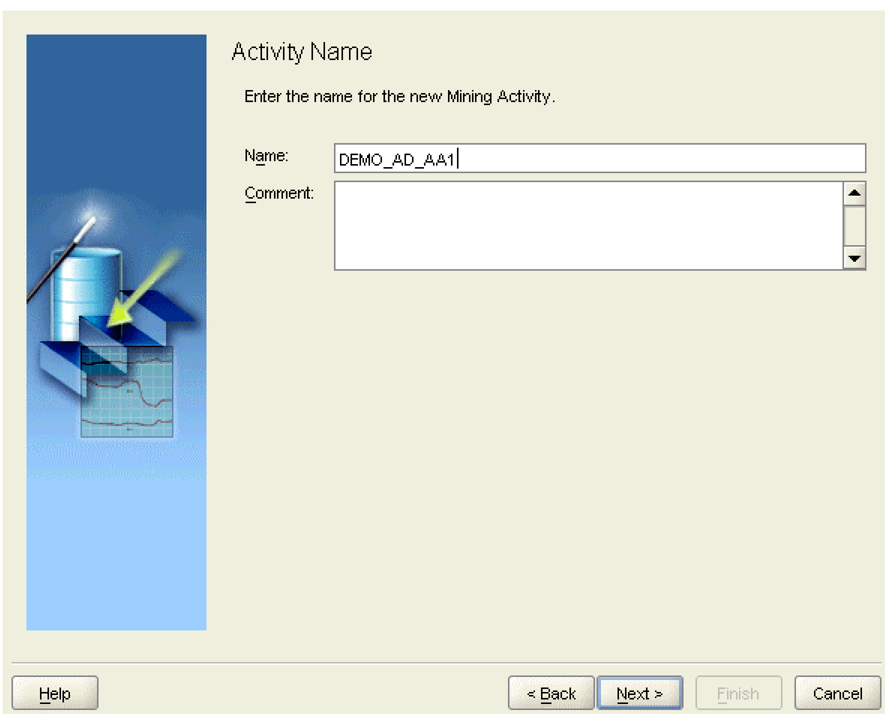
The RISK holdout sample with some known “suspicious” cases is selected. Click OK, then Next.



In order to compare the results, include the RISK column, which has automatically been given the Alias RISK_1, and click Next.



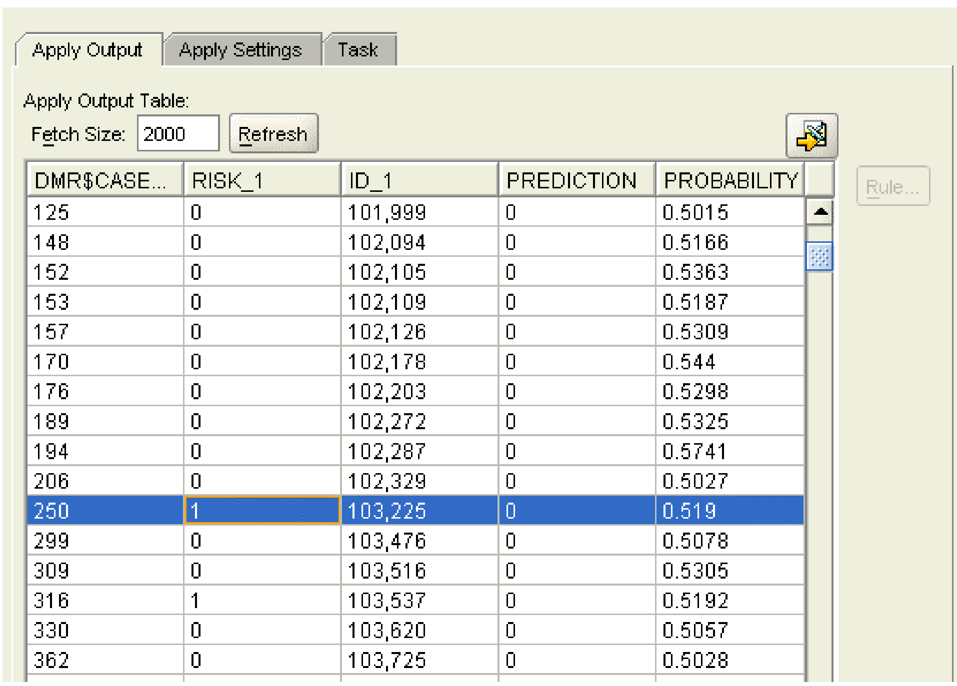
Enter a descriptive name and click Next to proceed to the final page of the wizard.



Click Finish on the final wizard page to execute the Activity. When the activity has been completed, click Result in the Apply step to see the output table. The Prediction column in the Anomaly Detection output table always has value 1 for a case determined to be “normal” and 0 for a “suspicious” case. Since the RISK column in the input data had value 0 for Low Risk and 1 for High Risk, the correctly predicted cases are those in which RISK_1 =1 and PREDICTION = 0.

For a display that is easier to interpret, click the column header PREDICTION to order the cases and show all suspicious cases at the top of the list.

Cases with PREDICTION = 0 will be investigated; in a problem as difficult as this, it should be considered a successful solution if 10% of those investigations result in the desired outcome.



DMR$CASE...	RISK_1	ID_1	PREDICTION	PROBABILITY
125	0	101,999	0	0.5015
148	0	102,094	0	0.5166
152	0	102,105	0	0.5363
153	0	102,109	0	0.5187
157	0	102,126	0	0.5309
170	0	102,178	0	0.544
176	0	102,203	0	0.5298
189	0	102,272	0	0.5325
194	0	102,287	0	0.5741
206	0	102,329	0	0.5027
250	1	103,225	0	0.519
299	0	103,476	0	0.5078
309	0	103,516	0	0.5305
316	1	103,537	0	0.5192
330	0	103,620	0	0.5057
362	0	103,725	0	0.5028

Chapter 13 - Association Rules

The Association Rules (AR) algorithm predicts the probability of co-occurrence among a given set of attribute values. The most well-known case of AR is Market Basket analysis, which predicts items occurring together in a market checkout session.

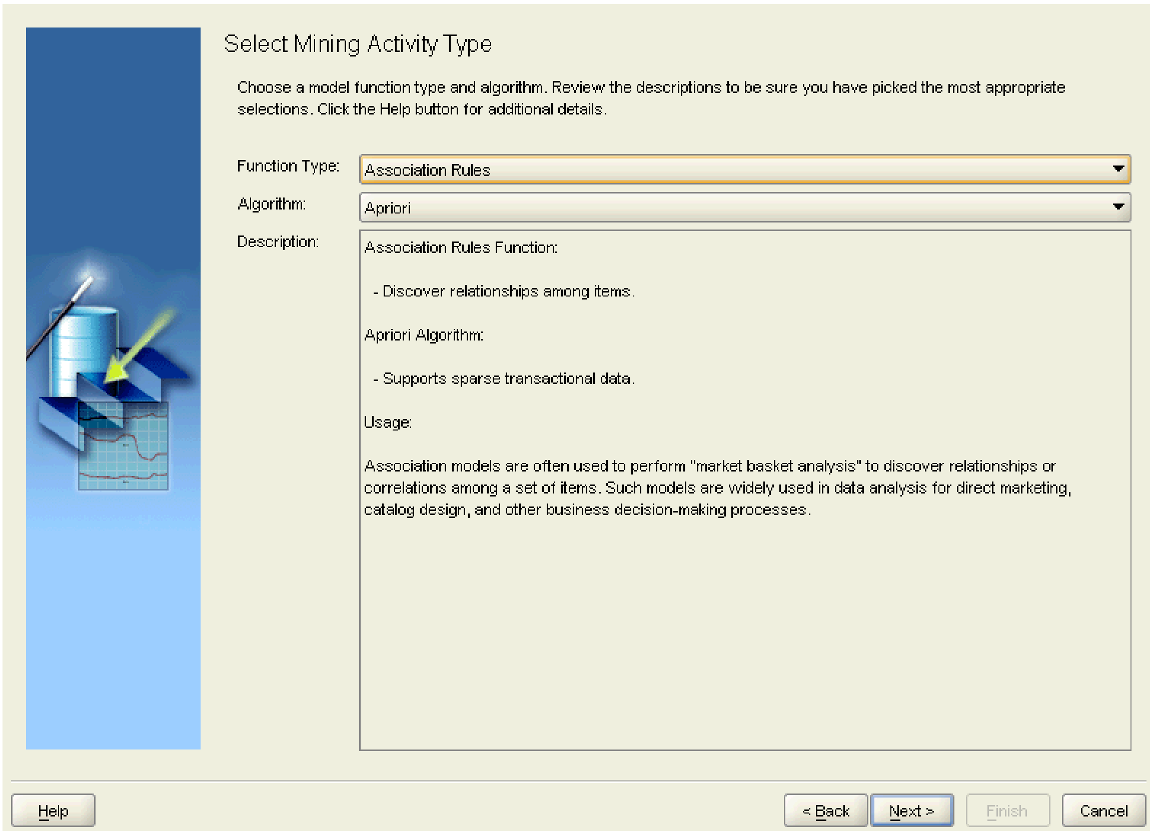
For the purpose of assistance with product placement in the stores, ODM's Association Rules feature will be used to measure the affinity between products.

If your ODM user was created and configured according to the instructions in Appendix A, you have access to the SH schema. The point-of-sale information in the SALES table will be used to illustrate Market Basket Analysis. The columns CUST_ID and PROD_ID are required to define the items purchased by a given customer. A single transaction (that is, a purchasing session resulting in a unique market basket) is identified by a combination of CUST_ID and TIME_ID.

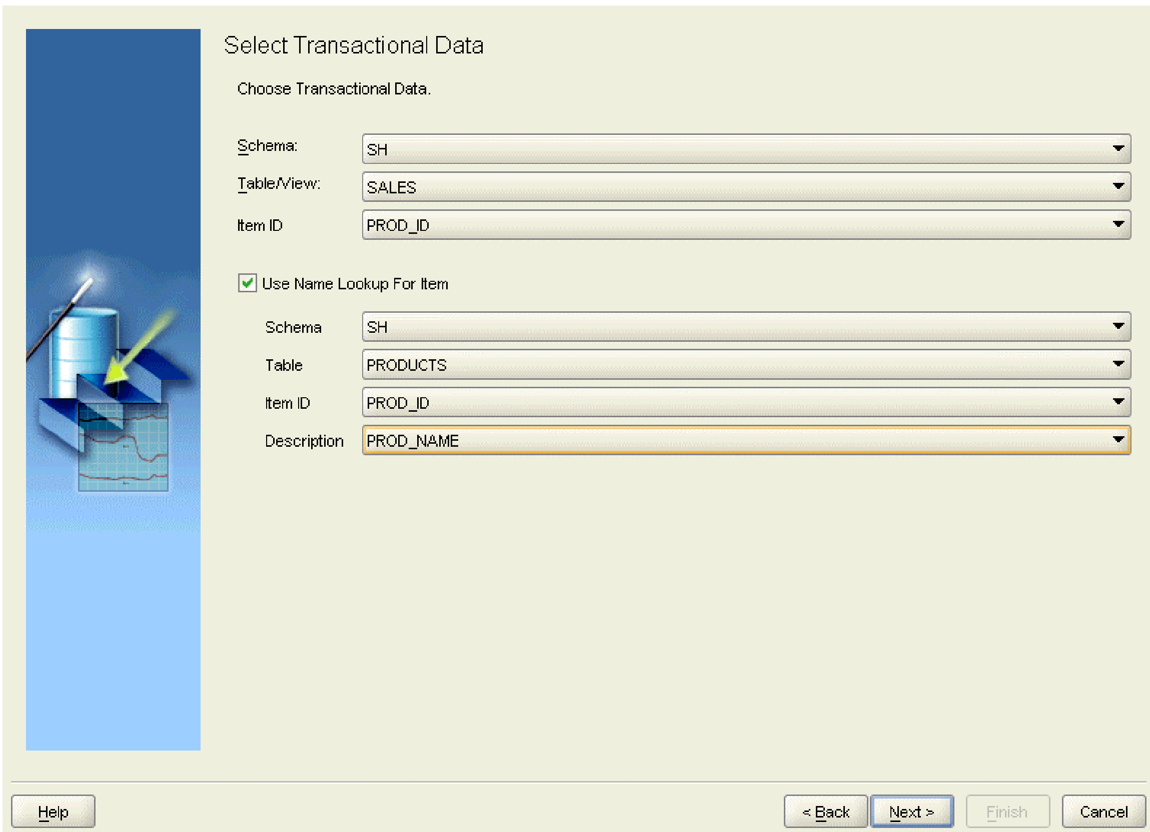
The Association Rules algorithm requires that data be in this "transactional" format, with one item being represented in each row.

PROD_ID	CUST_ID	TIME_ID	CHANNEL_ID	PROMO_ID	QUANTITY_SOLD	AMOUNT_S...
13	987	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	1,660	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	1,762	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	1,843	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	1,948	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	2,273	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	2,380	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	2,683	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	2,865	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	4,663	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	5,203	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	5,321	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	5,590	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	6,277	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	6,859	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	8,540	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	9,076	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	12,099	1998-01-10 00:00:00.0	3	999	1	1,232.16003...

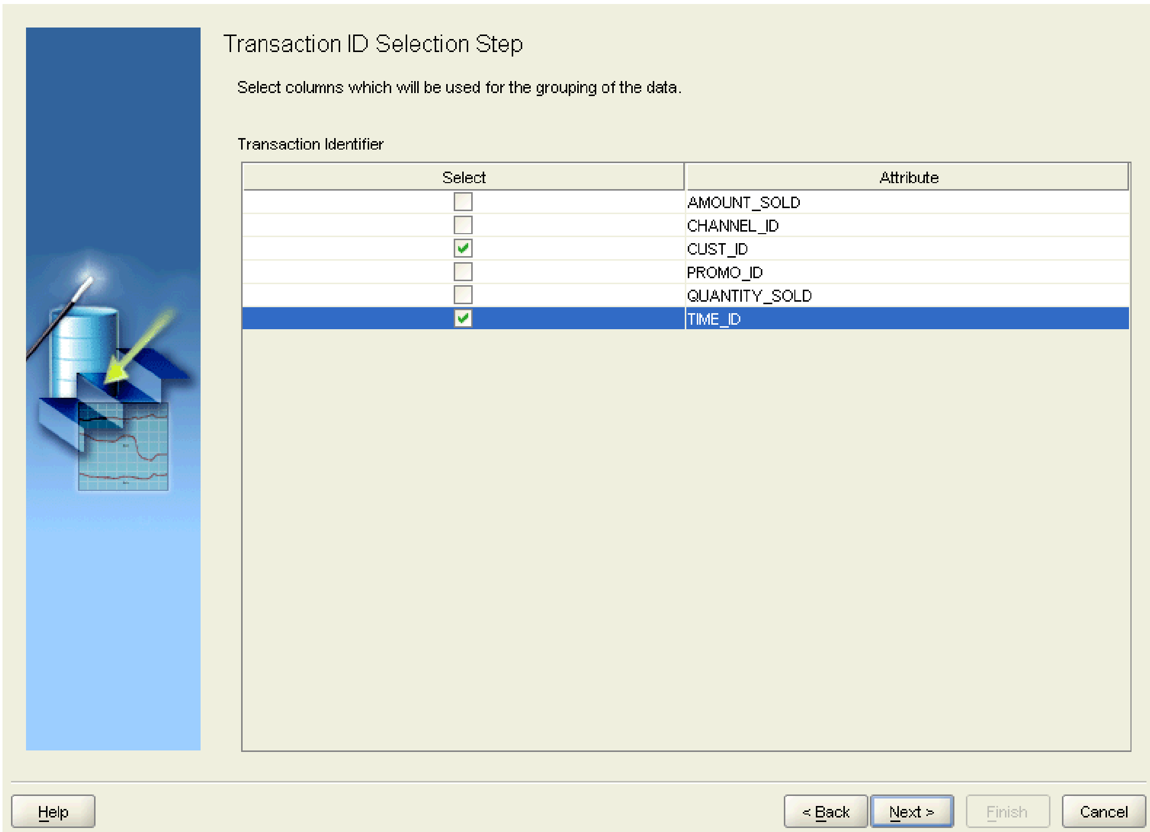
Select Build from the Activity pull-down menu to launch the activity, and select Association Rules from the Function Type pull-down menu. Click Next.



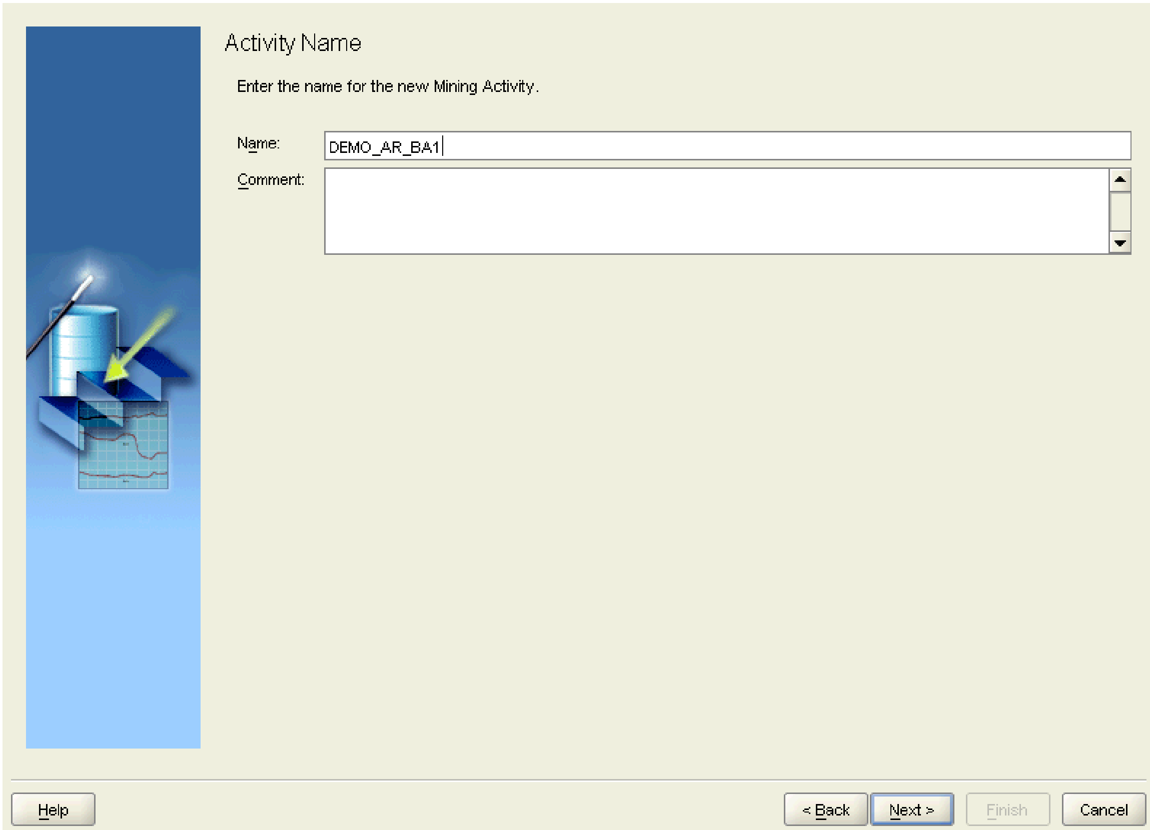
The transactions (market baskets) are contained in the SH.SALES table; select PROD_ID as the identifier for the items purchased. However, the products are identified only by an item code; the names of the products are in the SH.PRODUCTS table, so click the checkbox indicating that there is a Name Lookup table, and select the table, the item identifier (PROD_ID), and the column containing the item description (PROD_NAME). Click Next to continue.



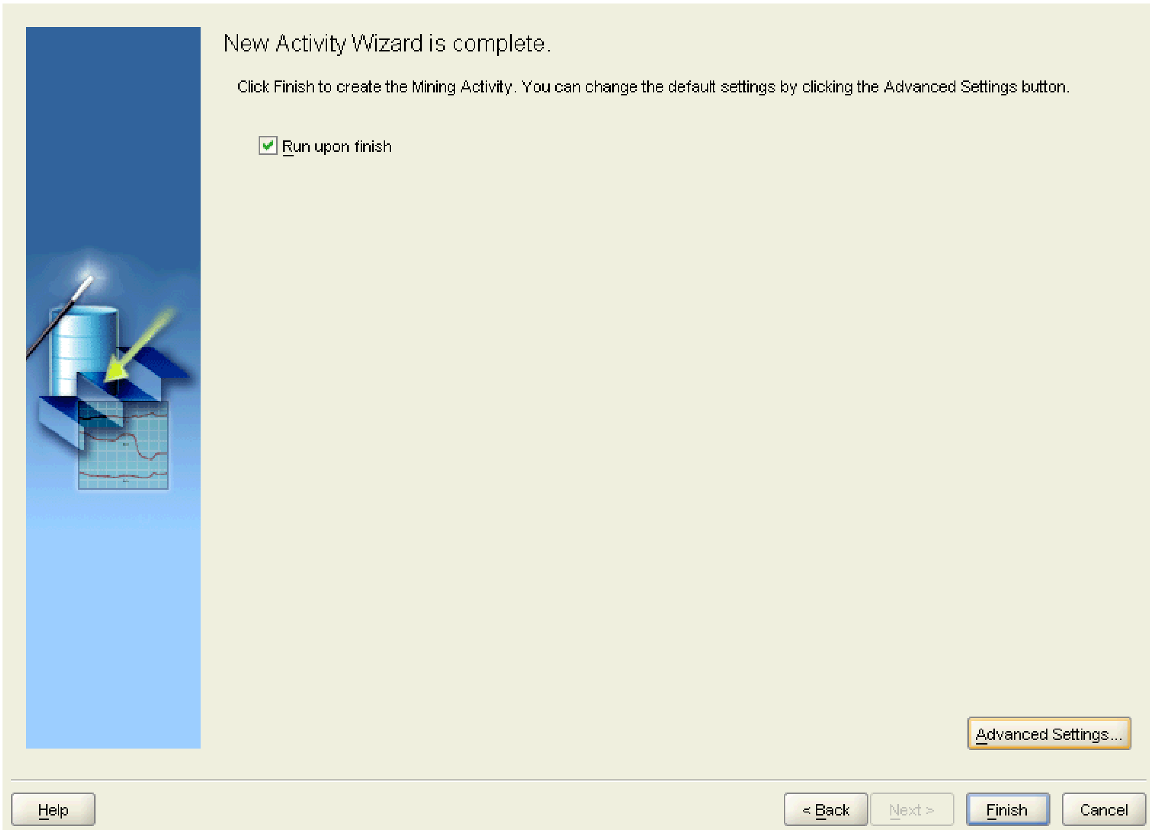
As noted previously, two columns are required to identify a single market basket. Click the checkboxes for CUST_ID and TIME_ID, then click Next.



Enter a name for the activity and click Next.



On the final page of the wizard, click Advanced Settings to see the parameters available for Association Rules.



Click the Build tab.

Each association rule is in the form:

If Product A and Product B and Product C … then Product X

The items on the left are called antecedents; the item on the right is the consequent.

The Length of the rule is the total number of items in the rule; for example, the rule: If Milk and Bread then Eggs has length = 3.

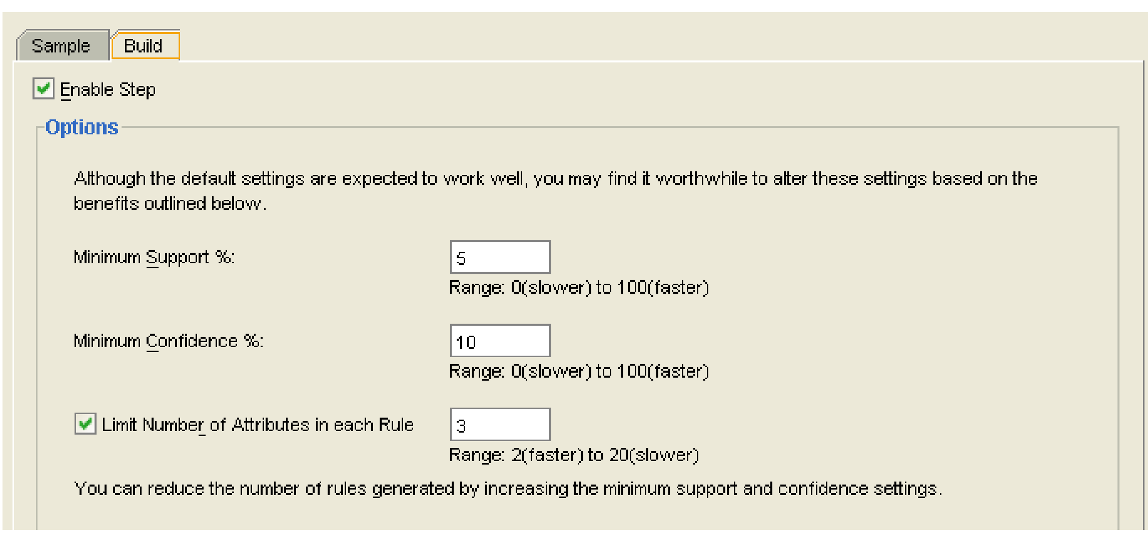
The Support for the rule is the percentage of baskets containing the items in the rule. In the example, Support is the percentage of all baskets containing the three items milk, bread and eggs.

The Confidence for the rule is the percentage of baskets containing the item(s) in the antecedents that also contain the consequent. In the example, consider only baskets containing milk and bread and calculate the percentage of those baskets that contain eggs.

Suppose that 100 market baskets are observed; suppose that 20 of those contain milk and bread, and 2 of those 20 contain eggs. Then the Support for the example rule is 2% (2 of 100), while the Confidence is 10% (2 of 20). Typically, the values for Confidence are much higher than those for Support.

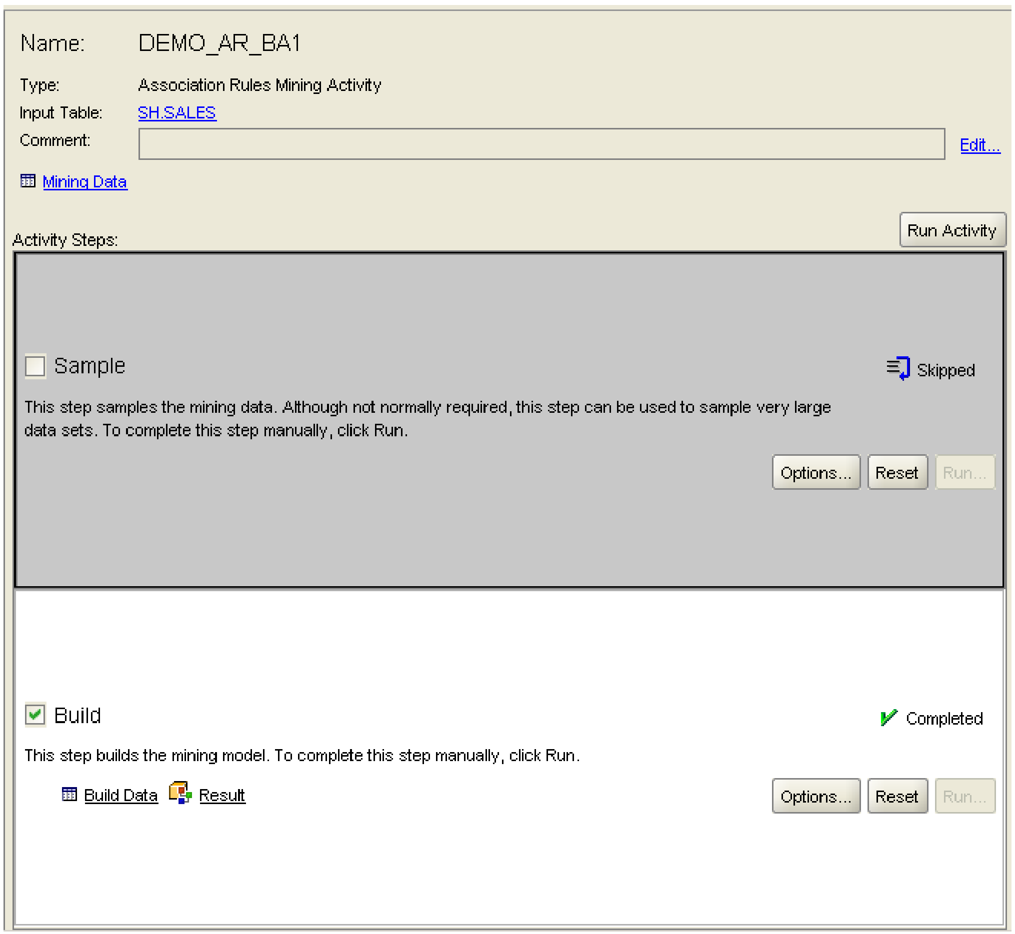
Setting minimums for Confidence and Support prevents the algorithm from wasting time and resources counting very rare cases. If either of these minimums is set too high, it is possible that no rules will be found; if they are set too low, there's a danger of exhausting system resources before the algorithm completes. Thus it is preferable to begin with the fairly high default values and then experiment with lower values.

Similarly, an increase in the maximum length increases the number of rules considerably, so begin with the low default and then increase slowly.

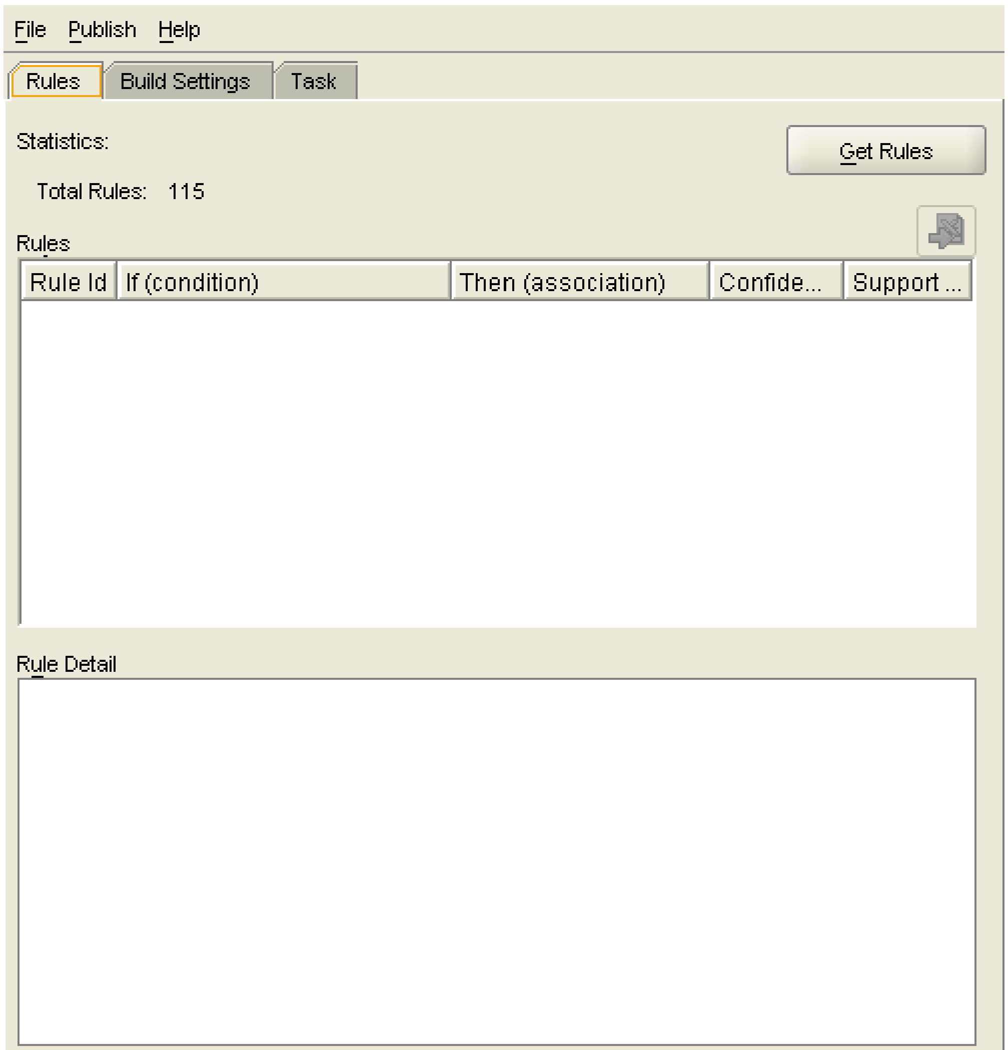


Click OK to return to the last page of the wizard and click Finish to run the activity.

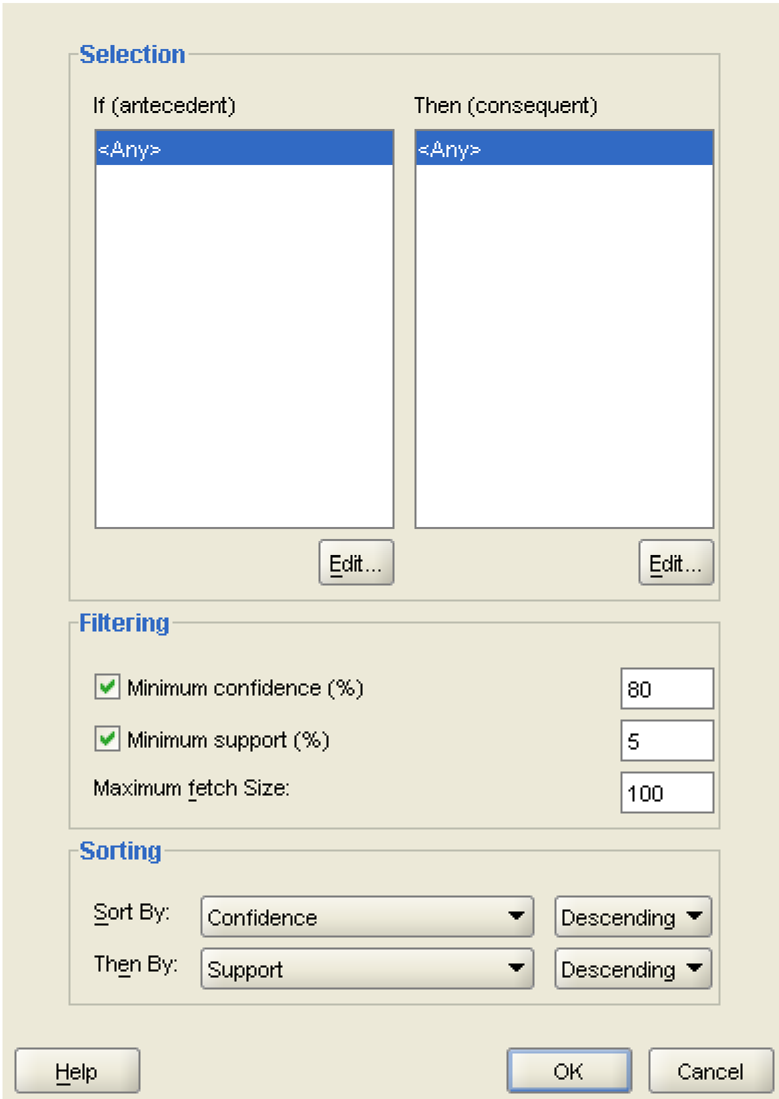
When the activity completes, click Result in the Build step to access the rules defined by the model.



In this example, 115 rules were defined using the default parameter settings. No rules are displayed initially, since there may be many thousands of rules in the model, and you may be interested only in a subset containing particular products. Therefore, you must request rules to see them.



Click Get Rules to initialize a dialog box for defining the rules to display.



There are several ways to select the rules for display:

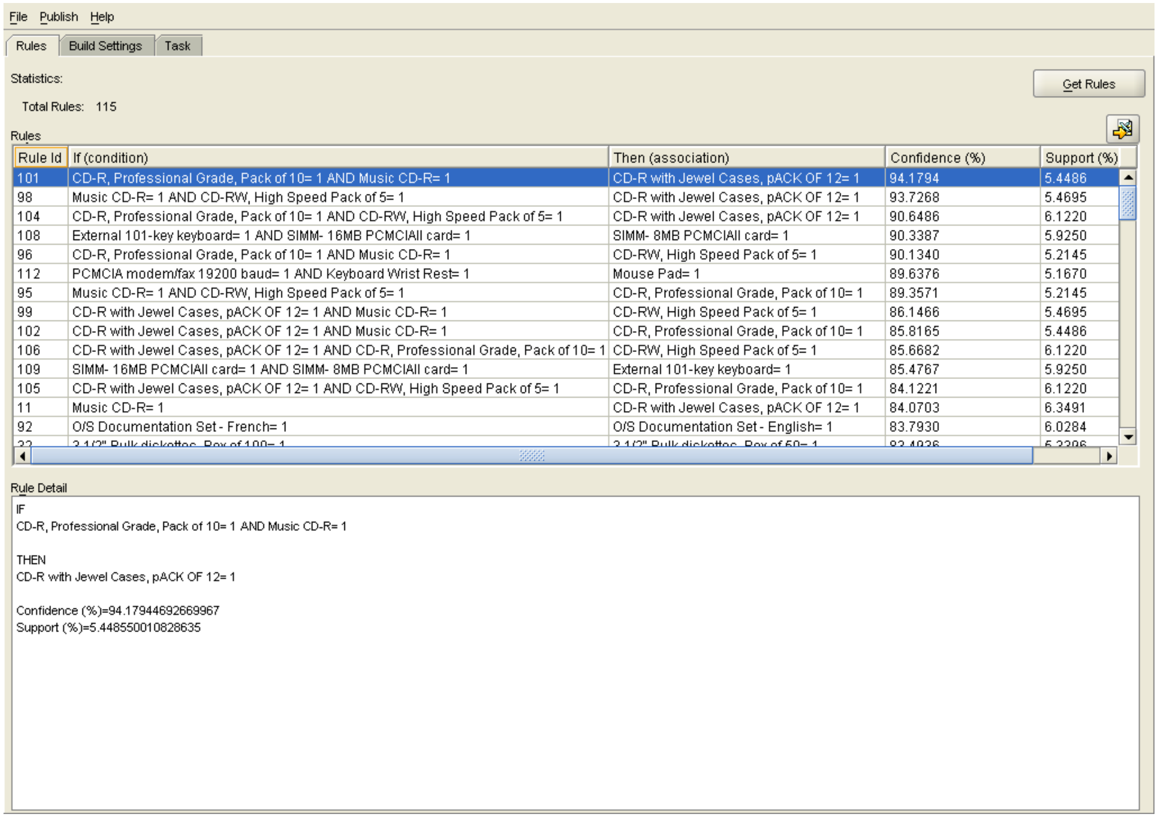
You can limit the number by adjusting the value in the Fetch box.

You can adjust the Minimum confidence and Minimum support.

You can limit the items shown in either the Antecedent or the Consequent by editing the list in either case.

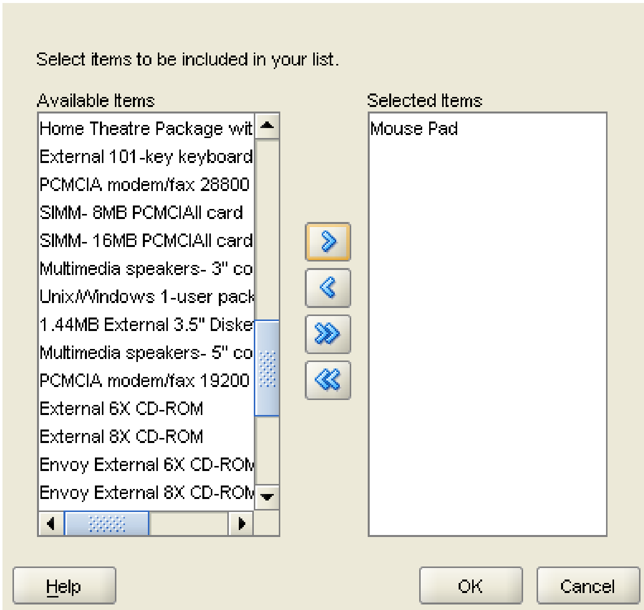
You can elect to sort the display by either Support or by Confidence.

Suppose that <Any> items are selected for both antecedent and consequent, and other settings are left with the default values. Click OK.

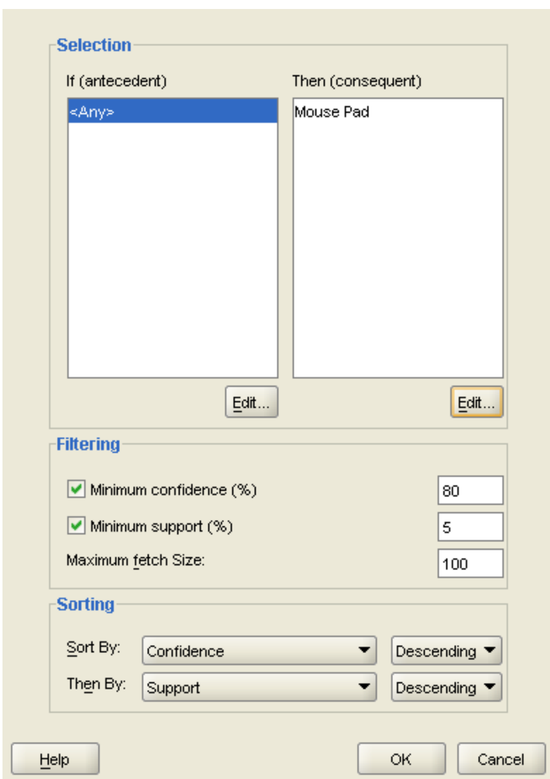


Now suppose that you want to see only those products associated with Mouse Pad. Click Get Rules and click Edit under the Consequent box to initialize a dialog box:

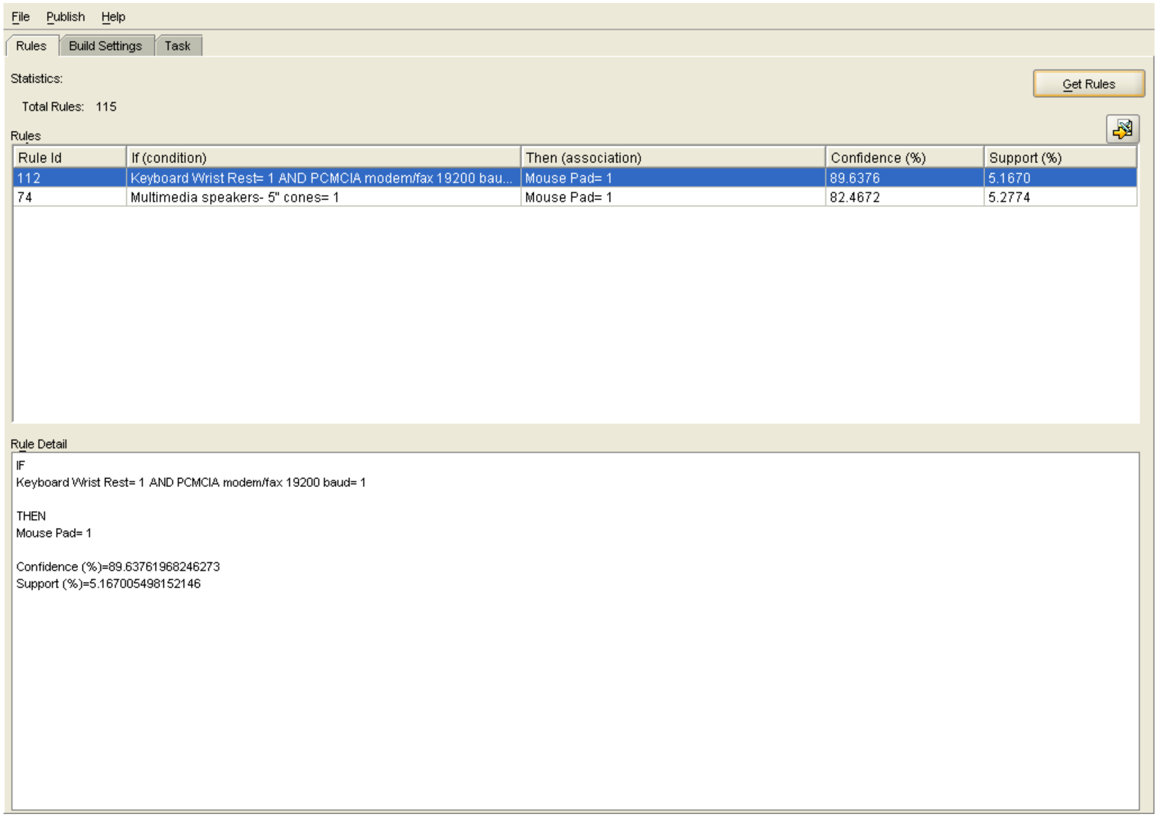
Move one item into the Selected window by highlighting that item and clicking >, or move more than one item by highlighting and clicking >>.



Click OK to conclude selection.



Click OK to display the rules having Mouse Pad as consequent.



Any rule that is highlighted in the grid is displayed in the Rule Detail window below.

You can sort the list by Support by clicking the header of the Support column, and you can reverse the order of the values by clicking the header again.

The notation “=1” is merely an indication that the item is present in the market basket.

You can save the rules to a spreadsheet or to a delimited text file by clicking on the floppy disk icon at the upper right corner of the grid; choose format, click OK, and specify file location.

You can save the rules into a database table that is “Discoverer-ready” by selecting Publish to Discoverer in the Publish pull-down menu.

Chapter 14 – Deployment

When the model that best solves the business problem has been chosen, then the solution must be put into the hands of the people who can improve the business by taking some action based on the results of the data mining. The deployment can take several forms:

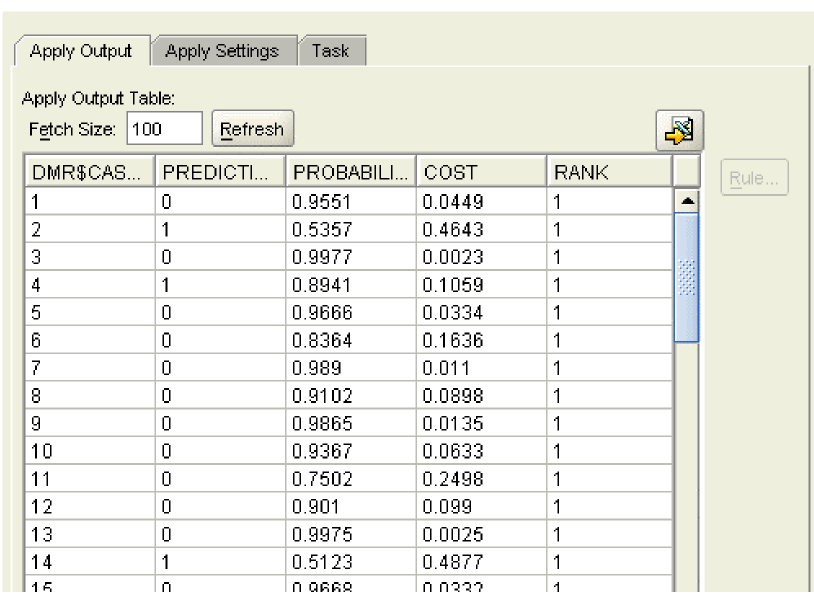
- Saving a scored list as a text file or spreadsheet
- Publishing a result into Oracle Discoverer
- Exporting a model to another Oracle database instance for scoring

Saving a scored list as a text file or spreadsheet

Any data in an object viewer whose display contains the icon

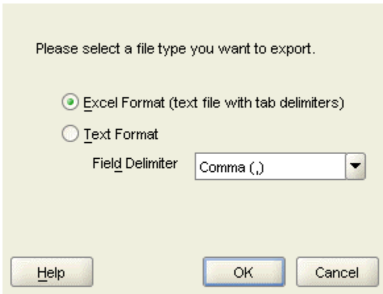
can be saved into a text file or a spreadsheet.

Suppose that an Apply result is displayed, and this information must be saved into a spreadsheet for transmission to account managers. First, enter the number of records to be saved in Fetch Size and click Refresh.

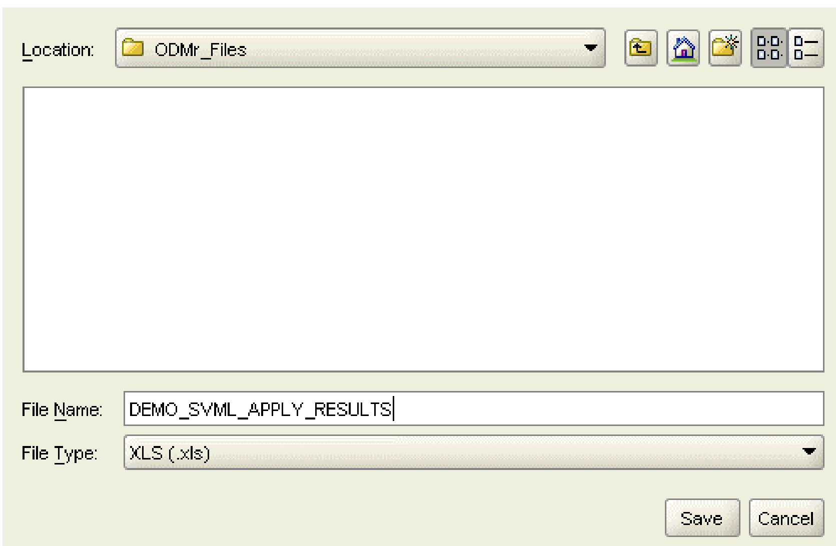


DMR$CAS...	PREDICTI...	PROBABILI...	COST	RANK
1	0	0.9551	0.0449	1
2	1	0.5357	0.4643	1
3	0	0.9977	0.0023	1
4	1	0.8941	0.1059	1
5	0	0.9666	0.0334	1
6	0	0.8364	0.1636	1
7	0	0.989	0.011	1
8	0	0.9102	0.0898	1
9	0	0.9865	0.0135	1
10	0	0.9367	0.0633	1
11	0	0.7502	0.2498	1
12	0	0.901	0.099	1
13	0	0.9975	0.0025	1
14	1	0.5123	0.4877	1
15	0	0.9668	0.0332	1

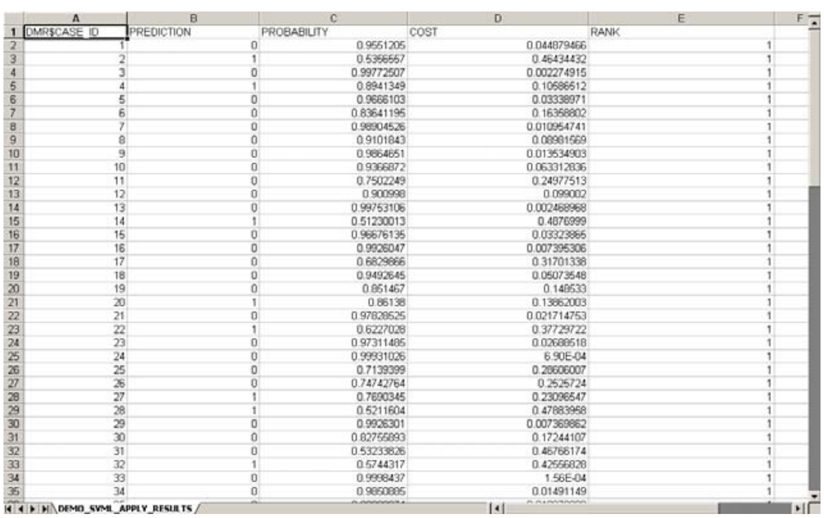
Click the icon to launch the wizard. Select the appropriate radio button for the desired format and click OK.



Select a storage location and enter a name for the spreadsheet file.



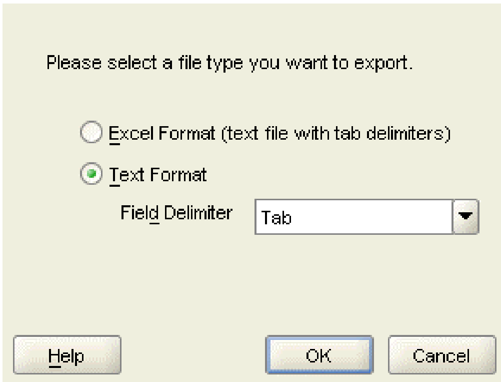
You can open the spreadsheet to see the result.



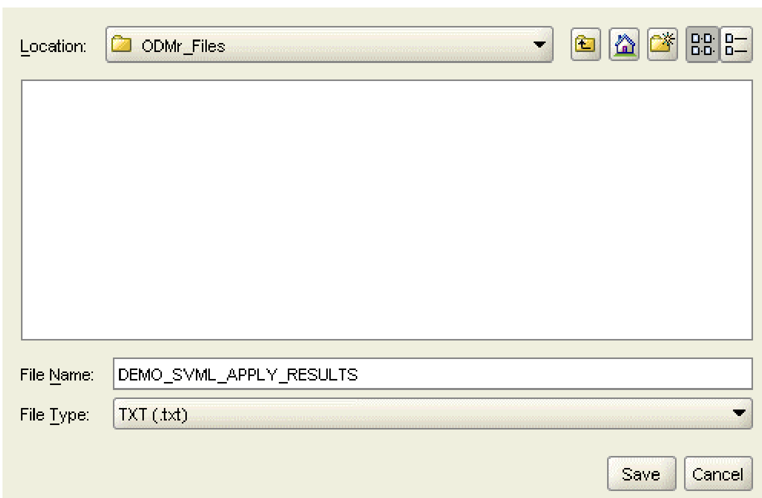
DMR$CASE_ID	PREDICTION	PROBABILITY	COST	RANK
1	0	0.9551205	0.044879466	1
2	1	0.5366557	0.46434432	1
3	0	0.99772507	0.002274915	1
4	1	0.8941349	0.10586612	1
5	0	0.9666103	0.03338971	1
6	0	0.83641195	0.16358802	1
7	0	0.98904526	0.010954741	1
8	0	0.9101843	0.08981569	1
9	0	0.9864651	0.013534903	1
10	0	0.9366872	0.063312836	1
11	0	0.7502249	0.24977513	1
12	0	0.900998	0.099002	1
13	0	0.99753106	0.002468968	1
14	1	0.51230013	0.4876999	1
15	0	0.96676135	0.03323865	1
16	0	0.9926047	0.007395306	1
17	0	0.6829866	0.31701338	1
18	0	0.9492645	0.05073548	1
19	0	0.851467	0.148533	1
20	1	0.86138	0.13862003	1
21	0	0.97828525	0.021714753	1
22	1	0.6227028	0.37729722	1
23	0	0.97311485	0.02688518	1
24	0	0.99931026	6.90E-04	1
25	0	0.7139399	0.28606007	1
26	0	0.74742764	0.2525724	1
27	1	0.7690345	0.23096547	1
28	1	0.5211604	0.47883958	1
29	0	0.9926301	0.007369862	1
30	0	0.82755893	0.17244107	1
31	0	0.53233826	0.46766174	1
32	1	0.5744317	0.42556828	1
33	0	0.9998437	1.56E-04	1
34	0	0.9850885	0.01491149	1

DEMO_SVML_APPLY_RESULTS

If a Tab-delimited text file is required, launch the wizard and make the selections, then click OK.



Enter a name and folder.



Open the file to see the result.

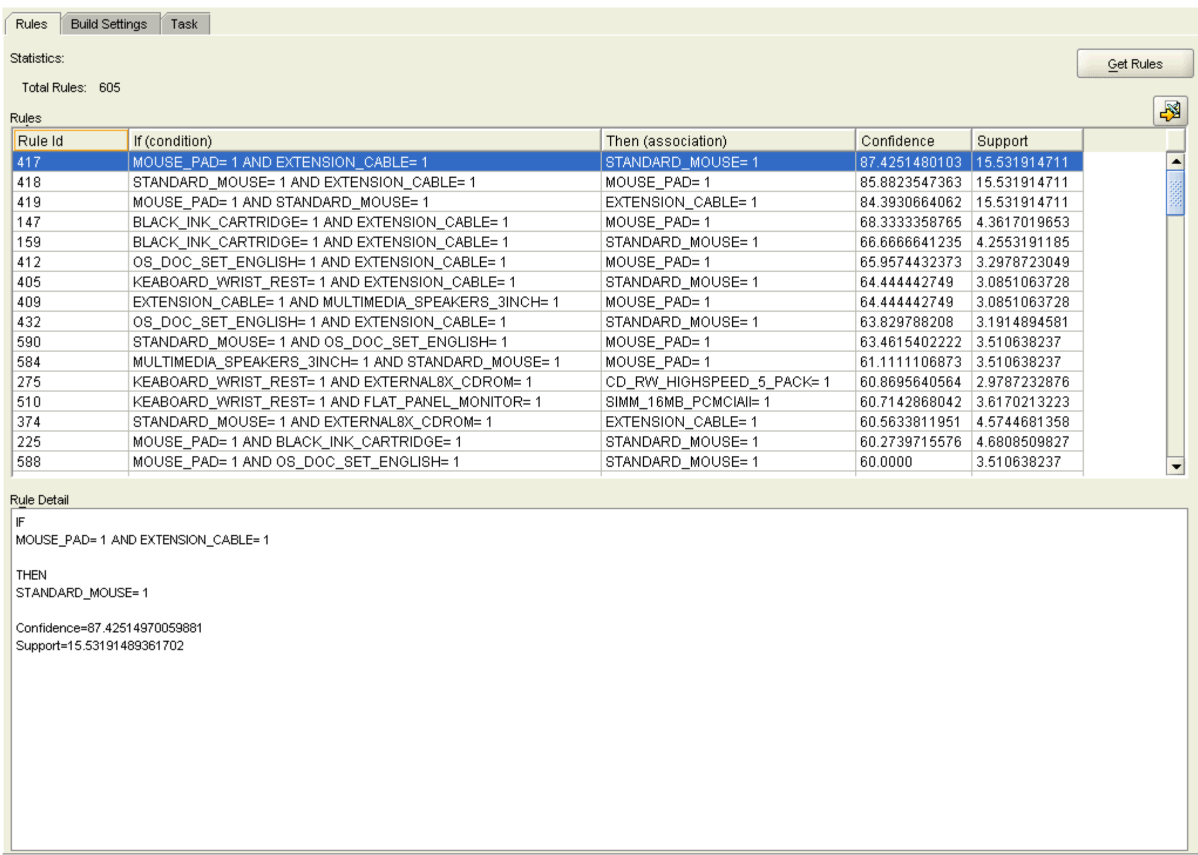
DMR$CASE_ID	PREDICTION	PROBABILITY	COST	RANK
1.0	0	0.9551205	0.044879466	1.0
2.0	1	0.5356557	0.46434432	1.0
3.0	0	0.99772507	0.0022749149	1.0
4.0	1	0.8941349	0.10586512	1.0
5.0	0	0.9666103	0.03338971	1.0
6.0	0	0.83641195	0.16358802	1.0
7.0	0	0.98904526	0.010954741	1.0
8.0	0	0.9101843	0.08981569	1.0
9.0	0	0.9864651	0.013534903	1.0
10.0	0	0.9366872	0.063312836	1.0
11.0	0	0.7502249	0.24977513	1.0
12.0	0	0.900998	0.099002	1.0
13.0	0	0.99753106	0.0024689676	1.0
14.0	1	0.51230013	0.4876999	1.0
15.0	0	0.96676135	0.03323865	1.0
16.0	0	0.9926047	0.007395306	1.0
17.0	0	0.6829866	0.31701338	1.0
18.0	0	0.9492645	0.05073548	1.0
19.0	0	0.851467	0.148533	1.0

Publishing a result into Oracle BI Discoverer

Oracle Data Miner includes a wizard to prepare objects created by ODM to be accessible to an Oracle Discoverer End User Layer (EUL) via Discoverer Gateway.

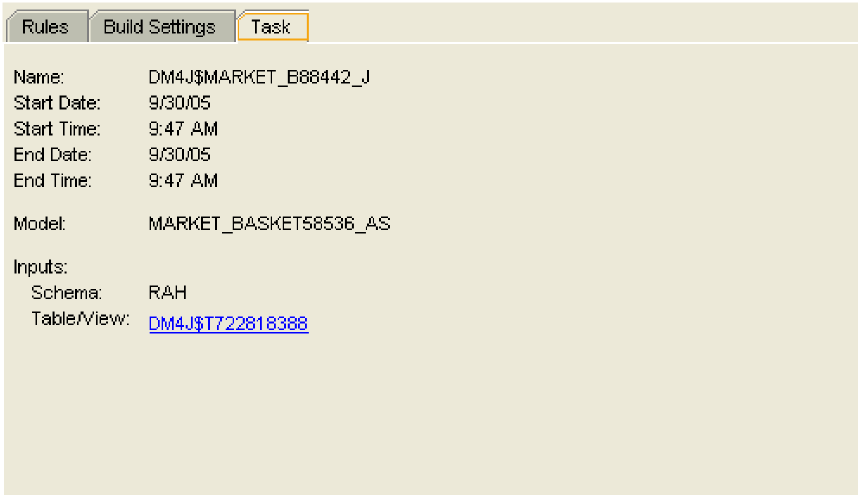
Some data mining objects are complex; the publishing wizard creates simple relational tables that can be added to a business area of an EUL.

For example, the market basket rules displayed as a result in the Association Rules build activity are actually part of the model definition, not a distinct table. They can be displayed through the Model Viewer:

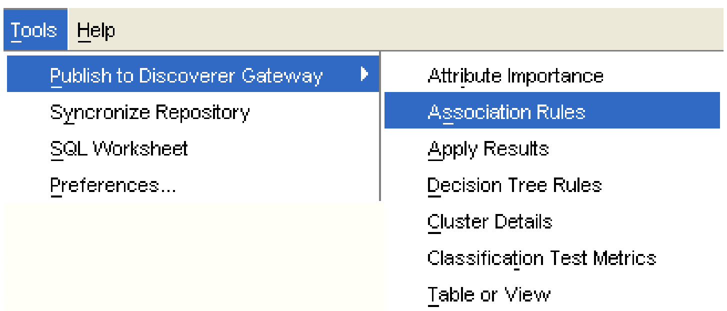


Rule Id	If (condition)	Then (association)	Confidence	Support
417	MOUSE_PAD= 1 AND EXTENSION_CABLE= 1	STANDARD_MOUSE= 1	87.4251480103	15.531914711
418	STANDARD_MOUSE= 1 AND EXTENSION_CABLE= 1	MOUSE_PAD= 1	85.8823547363	15.531914711
419	MOUSE_PAD= 1 AND STANDARD_MOUSE= 1	EXTENSION_CABLE= 1	84.3930664062	15.531914711
147	BLACK_INK_CARTRIDGE= 1 AND EXTENSION_CABLE= 1	MOUSE_PAD= 1	68.3333358765	4.3617019653
159	BLACK_INK_CARTRIDGE= 1 AND EXTENSION_CABLE= 1	STANDARD_MOUSE= 1	66.6666641235	4.2553191185
412	OS_DOC_SET_ENGLISH= 1 AND EXTENSION_CABLE= 1	MOUSE_PAD= 1	65.9574432373	3.2978723049
405	KEABOARD_WRIST_REST= 1 AND EXTENSION_CABLE= 1	STANDARD_MOUSE= 1	64.444442749	3.0851063728
409	EXTENSION_CABLE= 1 AND MULTIMEDIA_SPEAKERS_3INCH= 1	MOUSE_PAD= 1	64.444442749	3.0851063728
432	OS_DOC_SET_ENGLISH= 1 AND EXTENSION_CABLE= 1	STANDARD_MOUSE= 1	63.829788208	3.1914894581
590	STANDARD_MOUSE= 1 AND OS_DOC_SET_ENGLISH= 1	MOUSE_PAD= 1	63.4615402222	3.510638237
584	MULTIMEDIA_SPEAKERS_3INCH= 1 AND STANDARD_MOUSE= 1	MOUSE_PAD= 1	61.1111106873	3.510638237
275	KEABOARD_WRIST_REST= 1 AND EXTERNAL8X_CDROM= 1	CD_RW_HIGHSPEED_5_PACK= 1	60.8695640564	2.9787232876
510	KEABOARD_WRIST_REST= 1 AND FLAT_PANEL_MONITOR= 1	SIMM_16MB_PCMCIAII= 1	60.7142868042	3.6170213223
374	STANDARD_MOUSE= 1 AND EXTERNAL8X_CDROM= 1	EXTENSION_CABLE= 1	60.5633811951	4.5744681358
225	MOUSE_PAD= 1 AND BLACK_INK_CARTRIDGE= 1	STANDARD_MOUSE= 1	60.2739715576	4.6808509827
588	MOUSE_PAD= 1 AND OS_DOC_SET_ENGLISH= 1	STANDARD_MOUSE= 1	60.0000	3.510638237

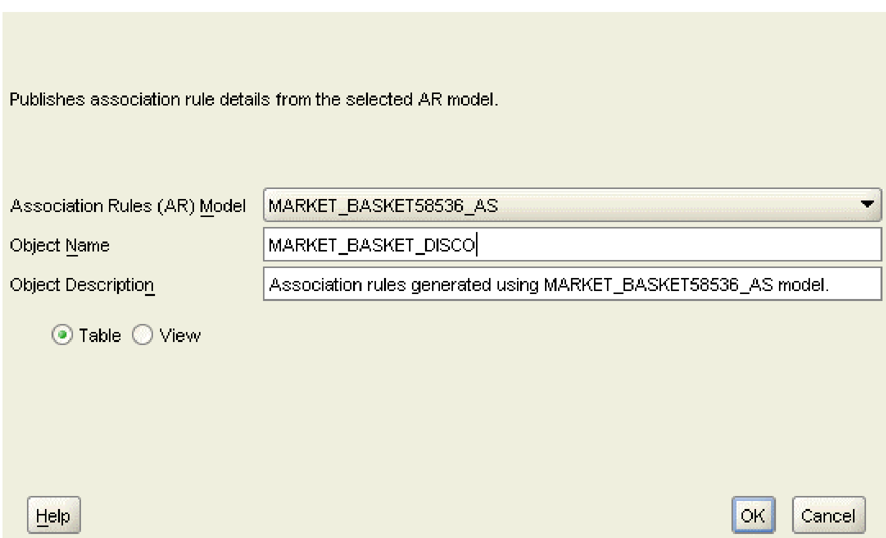
To publish these Association Rules to Discoverer, click the Build Result link in the activity, then click the Task tab to determine the full model name.



Next, initialize the Publish to Discoverer wizard.



Select the model name as shown in the Task details and enter a name for the object to be published. Select Table or View and click OK.



The resulting object is shown in the Navigation tree.

The table can be displayed like any table in the schema.

RULE_ID	RULE_ANTECEDENT_ITEMS	RULE_CONSEQUENT_ITEMS	RULE_SUPPORT	RULE_CONFIDENCE	RULE_LENGTH
417	MOUSE_PAD, EXTENSION_CABLE	STANDARD_MOUSE	0.1553191543	0.8742514849	2
418	STANDARD_MOUSE, EXTENSION_CABLE	MOUSE_PAD	0.1553191543	0.8588235378	2
419	MOUSE_PAD, STANDARD_MOUSE	EXTENSION_CABLE	0.1553191543	0.8439306617	2
147	BLACK_INK_CARTRIDGE, EXTENSION_CABLE	MOUSE_PAD	0.0436170213	0.6833333373	2
159	BLACK_INK_CARTRIDGE, EXTENSION_CABLE	STANDARD_MOUSE	0.0425531901	0.6666666865	2
412	OS_DOC_SET_ENGLISH, EXTENSION_CABLE	MOUSE_PAD	0.0329787247	0.6595744491	2
405	KEABOARD_WRIST_REST, EXTENSION_CABLE	STANDARD_MOUSE	0.0308510642	0.6444444656	2
409	EXTENSION_CABLE, MULTIMEDIA_SPEAKERS_3INCH	MOUSE_PAD	0.0308510642	0.6444444656	2
432	OS_DOC_SET_ENGLISH, EXTENSION_CABLE	STANDARD_MOUSE	0.0319148935	0.6382978559	2
590	STANDARD_MOUSE, OS_DOC_SET_ENGLISH	MOUSE_PAD	0.0351063833	0.6346153617	2
584	MULTIMEDIA_SPEAKERS_3INCH, STANDARD_MOUSE	MOUSE_PAD	0.0351063833	0.6111111045	2
275	KEABOARD_WRIST_REST, EXTERNAL8X_CDROM	CD_RW_HIGHSPEED_5_PACK	0.029787235	0.6086956263	2
510	KEABOARD_WRIST_REST, FLAT_PANEL_MONITOR	SIMM_16MB_PCMCIAII	0.0361702144	0.6071428657	2
374	STANDARD_MOUSE, EXTERNAL8X_CDROM	EXTENSION_CABLE	0.0457446799	0.6056337953	2
225	MOUSE_PAD, BLACK_INK_CARTRIDGE	STANDARD_MOUSE	0.046808511	0.6027397513	2
588	MOUSE_PAD, OS_DOC_SET_ENGLISH	STANDARD_MOUSE	0.0351063833	0.6000000238	2
481	STANDARD_MOUSE, EXTERNAL8X_CDROM	MOUSE_PAD	0.0446808524	0.5915492773	2
390	FLAT_PANEL_MONITOR, EXTENSION_CABLE	STANDARD_MOUSE	0.0531914905	0.5882353187	2
226	STANDARD_MOUSE, BLACK_INK_CARTRIDGE	MOUSE_PAD	0.046808511	0.5866666436	2
53	EXTENSION_CABLE	STANDARD_MOUSE	0.1808510572	0.5802047849	1
426	EXTENSION_CABLE, MULTIMEDIA_SPEAKERS_3INCH	STANDARD_MOUSE	0.0276595745	0.5777778029	2
280	CD_RW_HIGHSPEED_5_PACK, MULTIMEDIA_SPEAKERS_3I...	EXTERNAL8X_CDROM	0.0319148935	0.5769230723	2
434	STANDARD_MOUSE, OS_DOC_SET_ENGLISH	EXTENSION_CABLE	0.0319148935	0.5769230723	2
387	FLAT_PANEL_MONITOR, EXTENSION_CABLE	SIMM_16MB_PCMCIAII	0.0521276593	0.5764706135	2
525	MOUSE_PAD, FLAT_PANEL_MONITOR	STANDARD_MOUSE	0.0542553179	0.5730336905	2
98	STANDARD_MOUSE	MOUSE_PAD	0.1840425581	0.5728476644	1
360	EXTERNAL8X_CDROM, EXTENSION_CABLE	MOUSE_PAD	0.046808511	0.571428597	2

The fact that this table is in a storage location separate from the other tables in the schema makes it easy for Oracle Discoverer Gateway to pick the table and add it to an End User Layer.

Exporting a model to another Oracle database instance for scoring

You may develop models in one Oracle Enterprise Edition database, but you may want to apply the model to data in a different (production) database. Oracle 10*g* Release 2 and Oracle 11*g* Release 1 provide native import and export of all ODM models, using Oracle Data Pump Technology, for the purpose of moving a model from one database to another.

NOTE: Whatever transformations are used to prepare the source data for the building of the model must be repeated exactly in the production environment before the model can be used to score new data.

When a DBA exports and imports an entire database or an entire schema using Oracle Data Pump, then any data mining models contained in the database or schema are transferred.

You can export an individual model or several models using the Oracle Data Mining API at the command line level. There is no wizard in the Oracle Data Miner GUI to accomplish such a transfer.

The export operation creates a file in a folder that must exist prior to the export; it is referenced in the PL/SQL export function as a directory object, that is a logical name in the database that is mapped to the operating system file structure. Similarly, the database into which the model is imported must also have a directory object referencing the storage location of the file created by the export function.

Moreover, the tablespace name for the exporting schema must match the tablespace name for the importing schema. Only sysdba can create a new tablespace if that is necessary, so for practical reasons it makes sense for sysdba to check the tablespaces on both databases, create the directory objects, and grant to the ordinary user DMUSER the permission to write to and read from the directory objects.

Suppose that the folder C:\ODMr_Files exists. Then the following sequence gives DMUSER permission to create a directory object linked to C:\ODMR_Files to hold the files associated with exporting a model.

```
C:\>sqlplus sys/oracle as sysdba

SQL*Plus: Release 10.2.0.1.0 - Production on Fri Sep
30 11:16:58 2005

Copyright (c) 1982, 2005, Oracle.  All rights
reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release
10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> GRANT CREATE ANY DIRECTORY TO DMUSER;

Grant succeeded.

SQL>
```

Now DMUSER can create the needed directory.

```
C:\>sqlplus dmuser/dmuser

SQL*Plus: Release 10.2.0.1.0 – Production on Fri Sep
30 11:40:09 2005

Copyright © 1982, 2005, Oracle.  All rights reserved.


Connected to:
Oracle Database 10g Enterprise Edition Release
10.2.0.1.0 – Production
With the Partitioning, OLAP and Data Mining options

SQL> CREATE OR REPLACE DIRECTORY model_dump AS
'C:\ODMr_Files';

Directory created.

SQL>
```

Now sysdba grants directory access to DMUSER.

```
C:\>sqlplus sys/oracle as sysdba

SQL*Plus: Release 10.2.0.1.0 - Production on Fri Sep
30 12:01:00 2005

Copyright (c) 1982, 2005, Oracle.  All rights
reserved.


Connected to:
Oracle Database 10g Enterprise Edition Release
10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> GRANT READ, WRITE ON DIRECTORY model_dump TO
dmuser;

Grant succeeded.

SQL>
```

Suppose that DMUSER has created a Decision Tree model MINING_DATA_B4762_DT and wishes to export the model to another Oracle 10g R2 database. On the SQLPLUS command line, DMUSER must execute the EXPORT_MODEL function with arguments specifying the name of the dumpfile to be created, the directory object, and the model name,

```
SQL> EXECUTE DBMS_DATA_MINING.EXPORT_MODEL('DT3.DMP',
'model_dump', 'name = ''MINING_DATA_B4762_DT''');

PL/SQL procedure successfully completed.

SQL>
```

Note: The model name is surrounded by two single quotes, not double quotes.

Now copy the file DT3.DMP to the existing directory NEW_DIR on the destination server.

Importing the model to the new database

Assuming that sysdba has granted permission to the user on the destination database to create and read from the directory NEW_DIR, the model can be imported and used by executing the following command.

```
SQL> exec dbms_data_mining.import_model('DT3.DMP',
'NEW_DIR');
```

Since no model name is entered as an argument, all models in the dumpfile are imported.

The model is now available for use in the new environment. Recall that in order to apply the model to data, the data must be prepared in exactly the same way that the source data for building the model was prepared.

Chapter 15 – From ad hoc Data Mining to Data Mining Application

Build and apply a model using mining activities; deploy the code in an application

1. Create and Run the Mining Activities

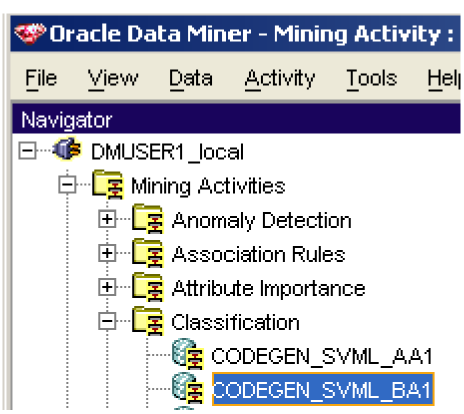
When you complete any Activity in the Oracle Data Miner GUI, the activity automatically generates PL/SQL code that can captured as a package and re-run to repeat the operations.

This example shows how to create a PL/SQL package from a Classification Apply Activity, then how to execute the code in the database to create a new result.

The goal is to illustrate how to create an application that can execute a previously-created model against several types of input: a table, the output of a SQL query, or a single-record dataset.

First, as shown in Chapter 8 of the Tutorial, create a Classification build activity using the Linear SVM algorithm, input data MINING_DATA_BUILD_V, and Target AFFINITY_CARD. Then create an apply activity with input data MINING_DATA_APPLY_V. To observe the same results as shown in the example below, choose Supplemental Attributes AGE and CUST_MARITAL_STATUS, and select as Apply Option: Specific Target Value 1.

In the examples shown below these activities are named CODEGEN_SVML_BA1 and CODEGEN_SVML_AA1 and are in the DMUSER1 schema in the database ORA10GR2.



Note: The code generated by Oracle Data Miner 10.2 Activities can be accessed and tested using either JDeveloper (any recent version) or SQL Developer (1.0, but not more recent). The instructions for downloading and configuring the required code generation extension for either JDeveloper or SQL Developer can be found at

http://www.oracle.com/technology/products/bi/odm/odminer.html

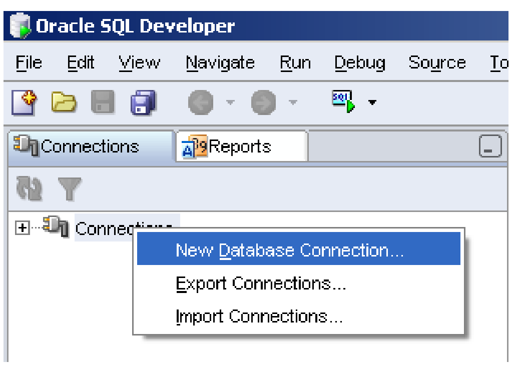
in the Downloads section for Oracle Data Miner 10.2.

The example below illustrates using SQL Developer; the steps for JDeveloper are similar unless otherwise noted.

2. Launch SQL Developer and Create a Database Connection

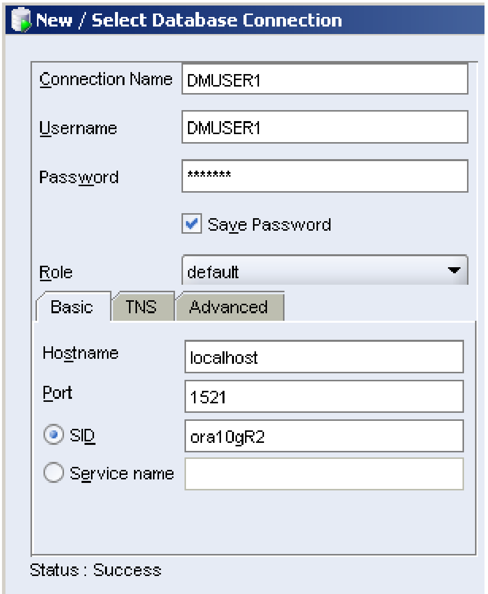
Launch SQL Developer and select View → Connections. If there is already a connection to the DMUSER1 schema in ORA10GR2, then skip to Step 3.

To create a new connection, in the Connections frame right-click Connections and select New Database Connection to initialize the New/Select Database Connection wizard. (In JDeveloper, click the Connections tab, right-click Database and select New Database Connection)



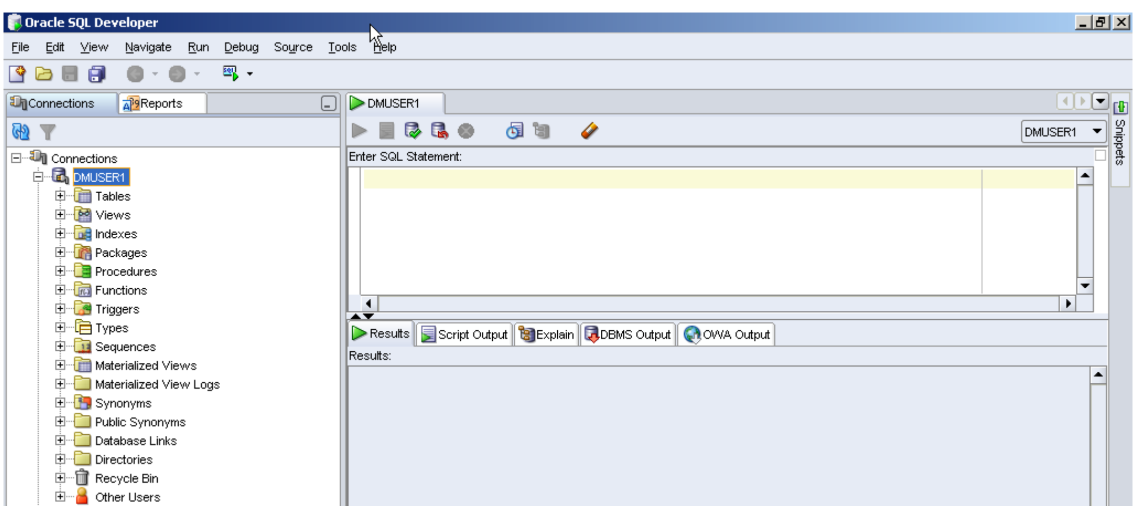
Enter a name for the connection, and the Username and Password for the schema. Enter the full system name or IP address (or the word "localhost" if SQL Developer is on the same system as the database), and the Port and SID (or Service Name) for the database. You must check the Save Password box for the SQL Developer process to succeed. ("Deploy Password" in JDeveloper)

When you click the TEST button you should see Status: Success under the dialog box. Then click CONNECT and the name of the new connection will appear in the Connections tree. (in JDeveloper, click Finish; then Connect manually)

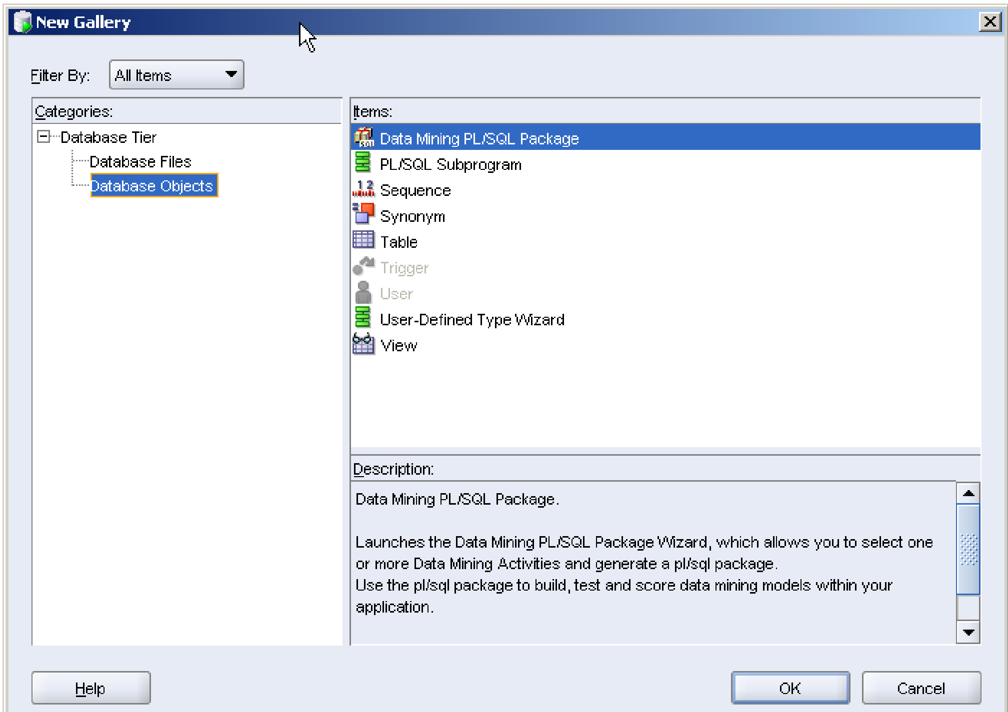


3. Create a PL/SQL Package from the Apply Activity

In order to create a PL/SQL Package, highlight the new connection name and expand it by clicking "+", then select File → New to launch a New Gallery dialog. (In JDeveloper, right-click the database name, select Connect, and highlight the connection name).

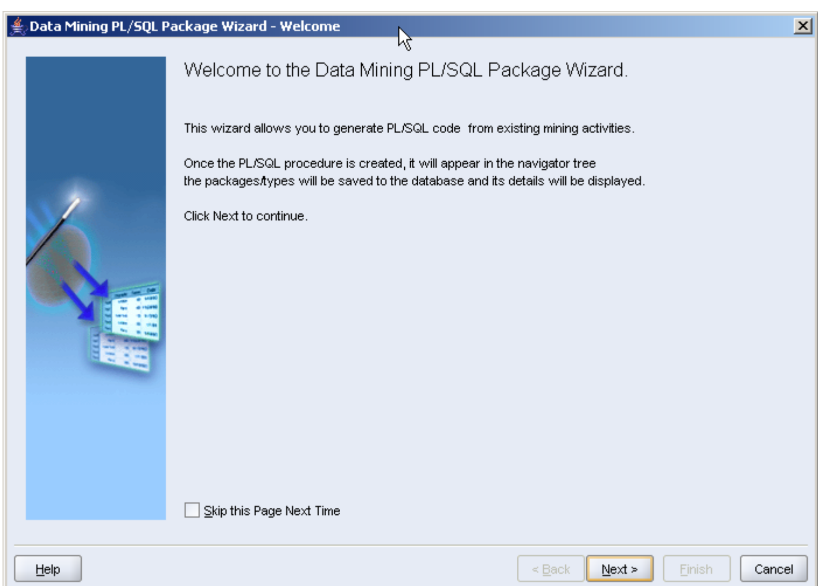


On the Filter By pull-down menu, select All Items, then (you may have to expand Database Tier) highlight Database Objects and Data Mining PL/SQL Package.

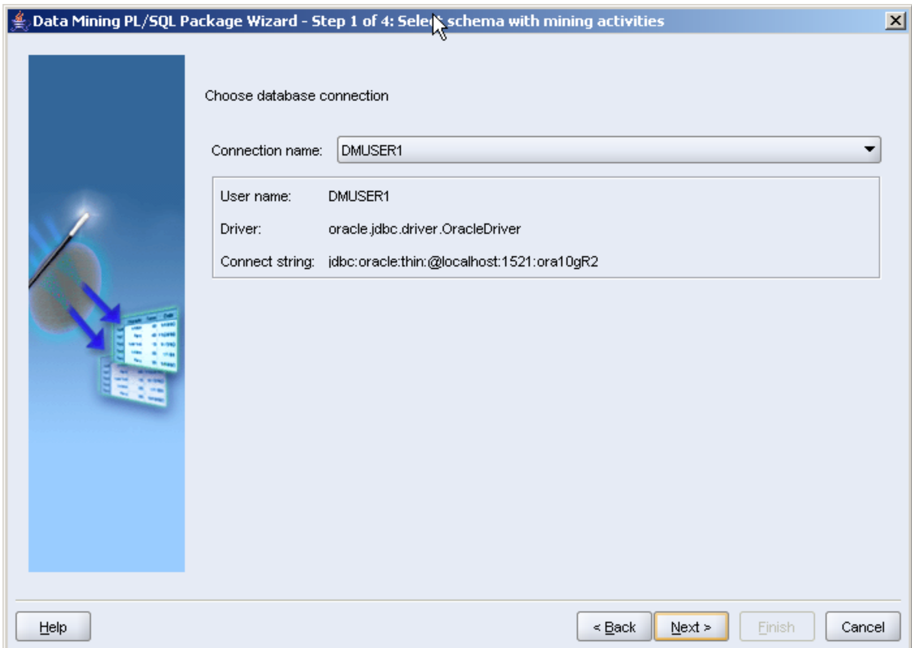


Click OK to launch the Data Mining PL/SQL Package wizard.

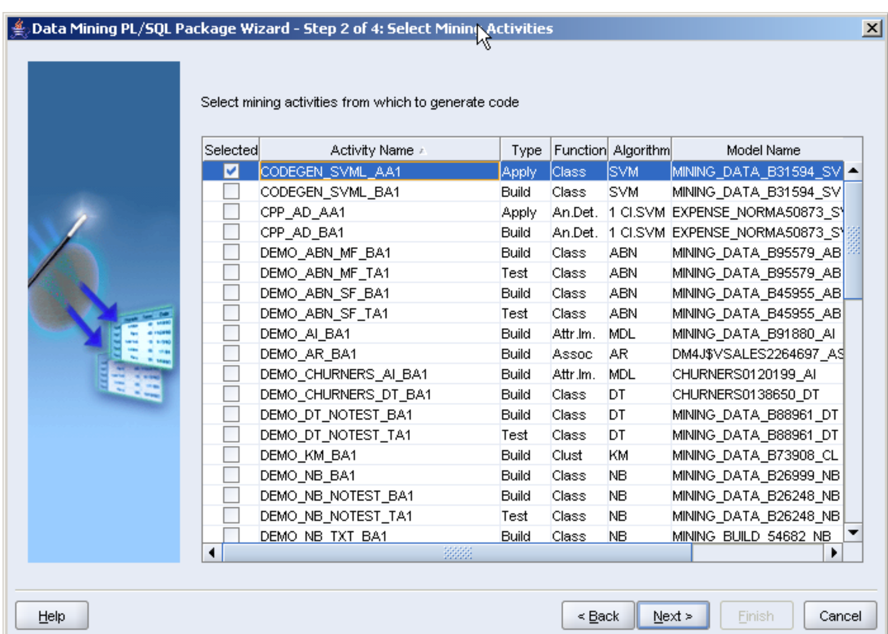
Click Next on the Welcome page.



In Step 1, choose the database connection; the default should be the connection highlighted in the Connections tree. Click Next.



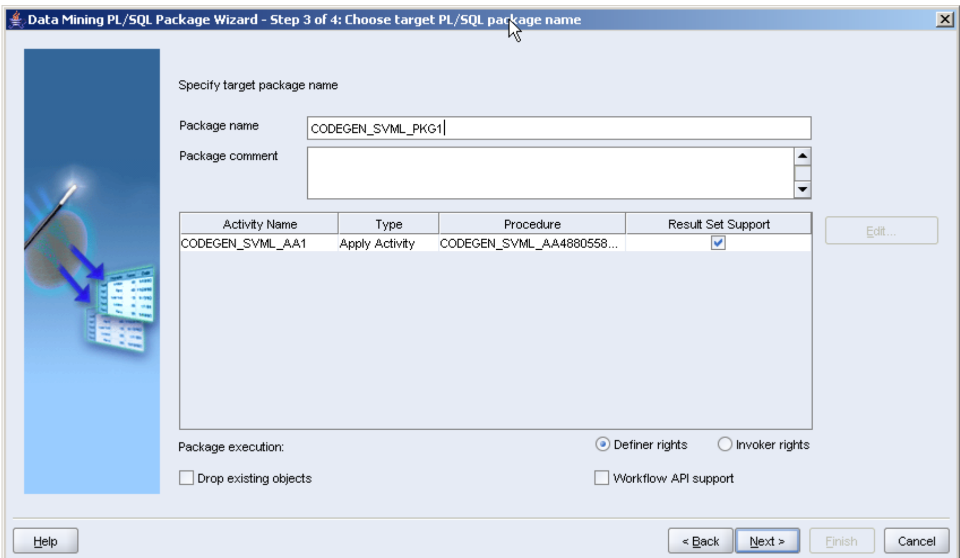
In Step 2, click the check box next to the activity to be packaged (highlighting the name is not enough), and click Next.



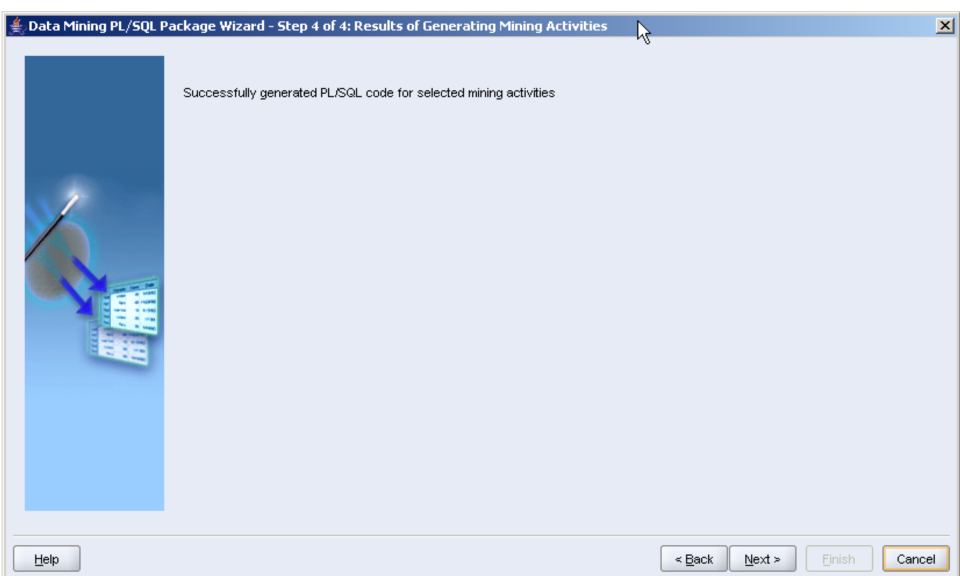
In Step 3, assign a name to the package and ensure that Result Set Support and Definer Rights are checked. Result Set Support is not required if input to the model apply code is always a table or a SQL query result. Definer Rights is necessary in order to edit the code, which will be done in most cases

Note: Do not check Workflow API **unless** the only use for the code will be in conjunction with the Oracle Workflow product. If checked, a package will be created that is designed to run only in the Workflow environment, and any execution in the absence of Workflow will fail.

Click Next.

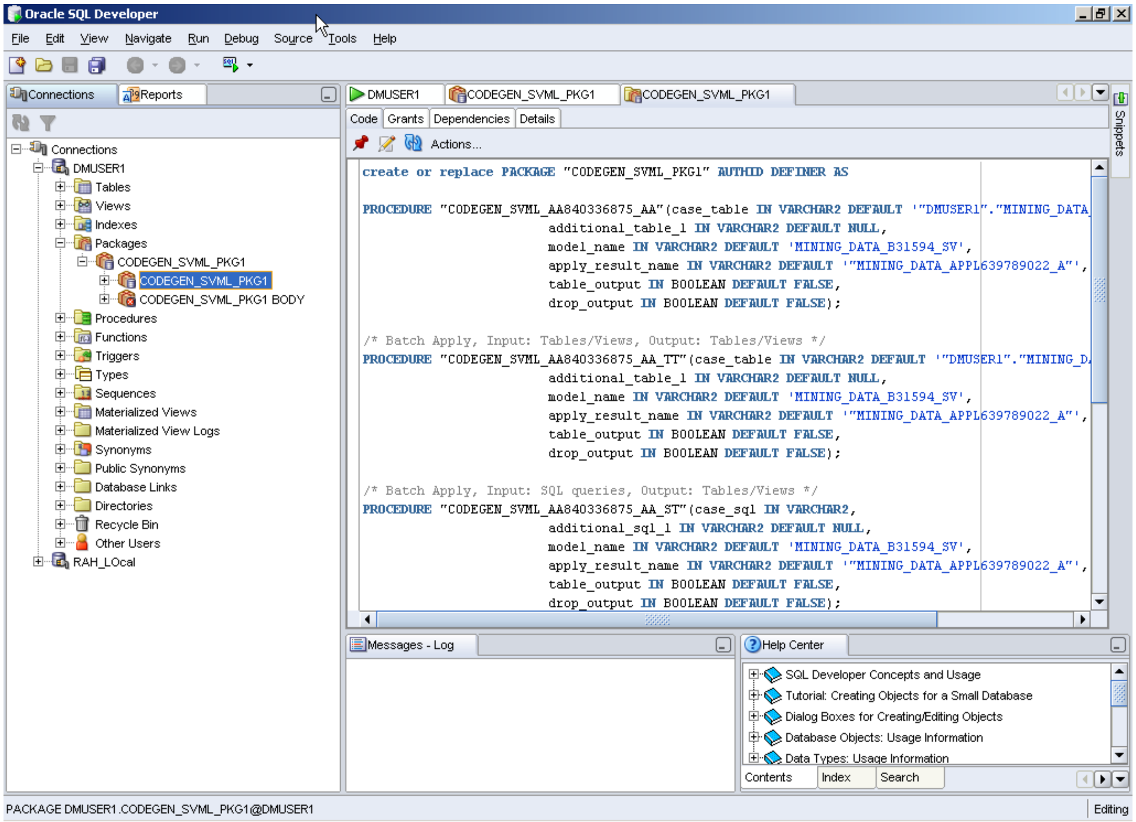


Step 4 confirms successful creation of the package. Click Next, then Finish on the final page of the wizard.

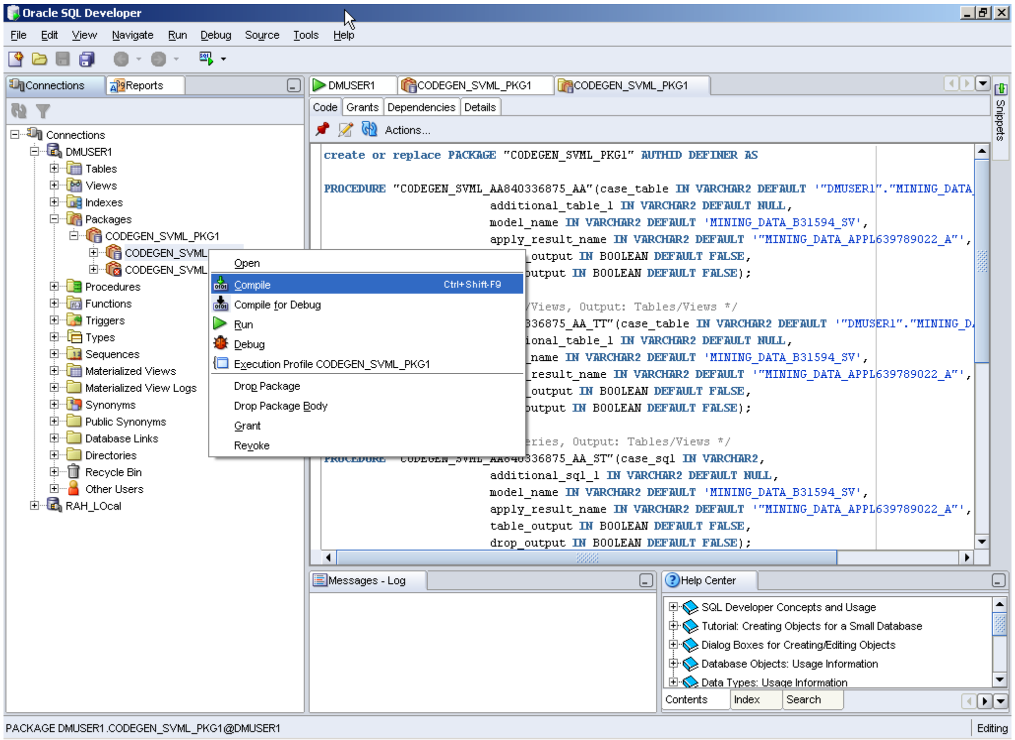


4. Test the Package Execution with Original Activity Settings

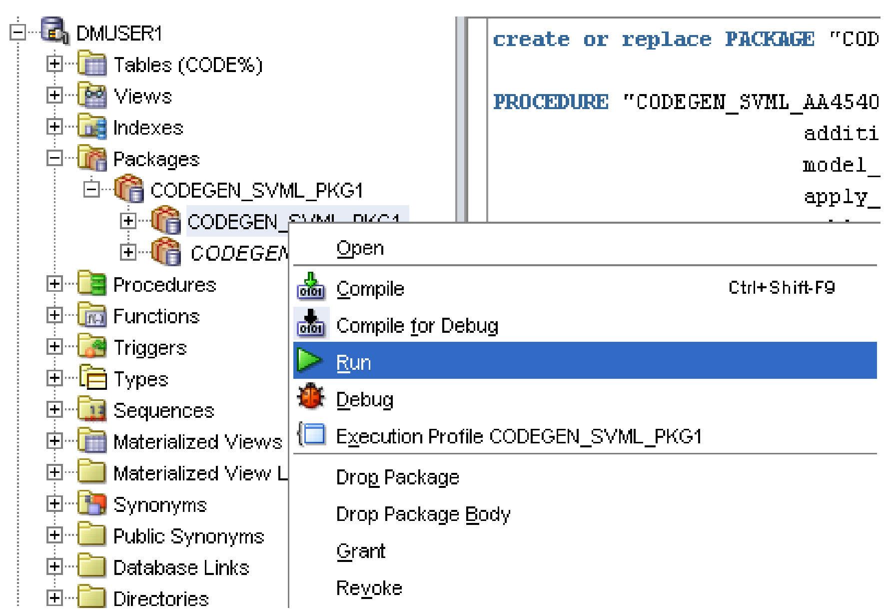
Expand Packages, then expand the package name to see the two files that make up the package: a header file with the same name as the package, and the code file with the suffix BODY added.



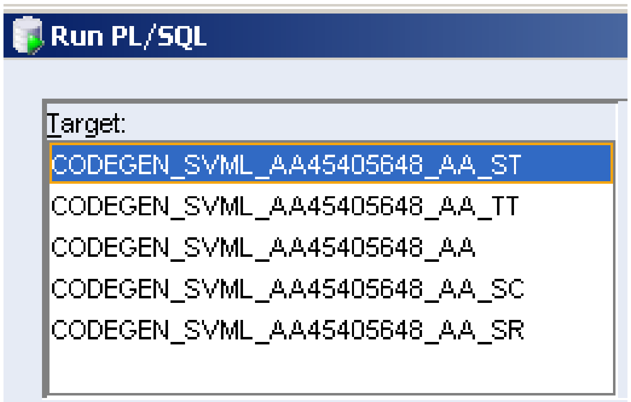
Right-click the BODY name and select Compile; repeat for the header file. The icons next to the file names may change slightly.



Right-click the header name and select Run (If you had not previously compiled the code it is automatically compiled now). The Run PL/SQL window opens and displays the components of the package.



The Target frame displays the components within the package, each with a descriptive suffix indicating the formats of the inputs and outputs:



TT: Table or View in, Table out
ST: SQL query in, table out
SC: SQL query in, cursor out
SR: SQL query in, result set out

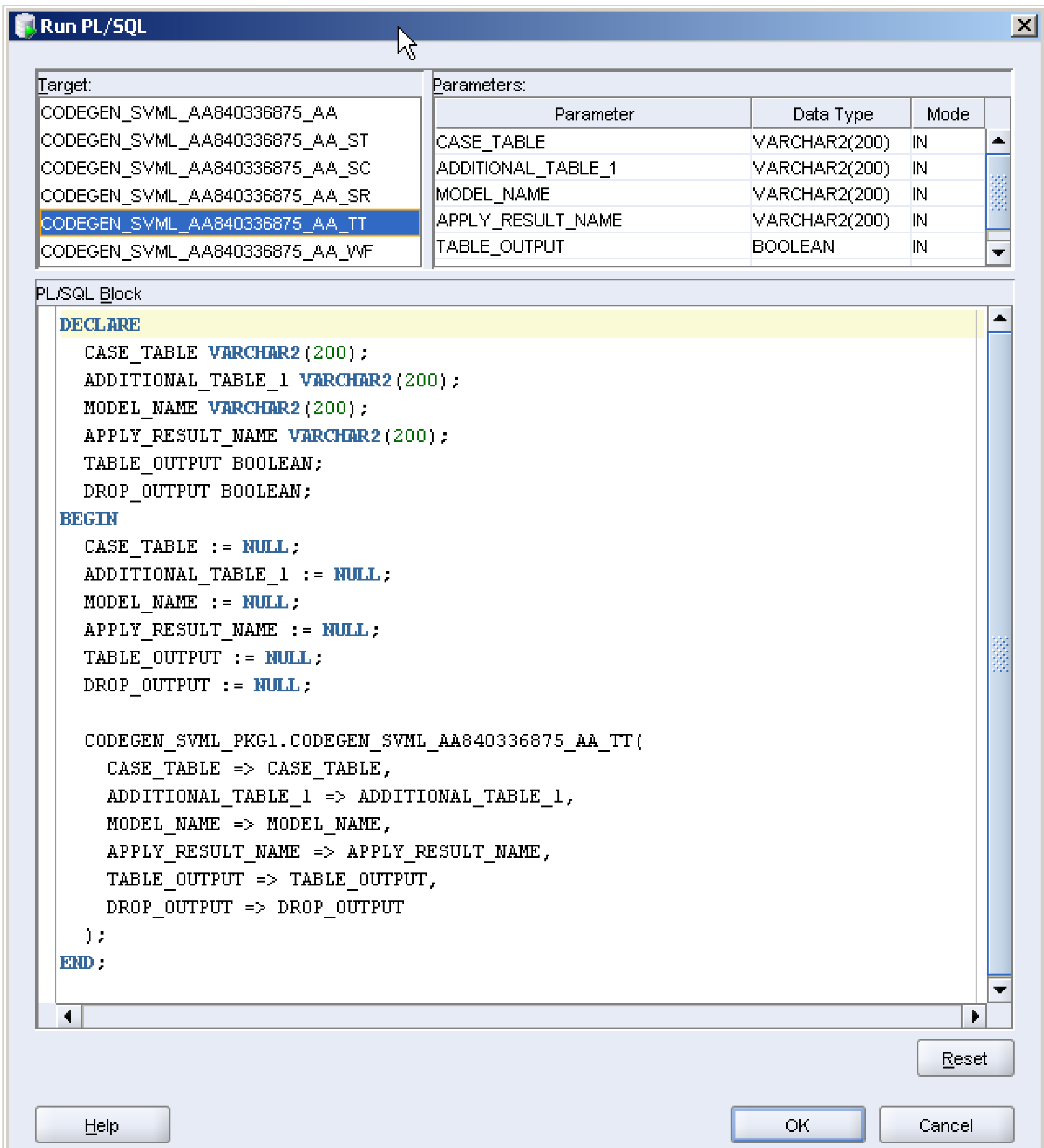
The following example will replicate the method of the Oracle Data Miner Apply activity: table or view as input for the scoring (in the activity as originally run, the view MINING_DATA_APPLY_V), and a new table containing the results of the scoring.

Select the component with the TT suffix.

By default, the arguments in the PL/SQL code take their values from the variable values defined after the BEGIN keyword, and by default these variable values are passed without change from the activity. If no values passed from the activity are to be changed before execution, leave the default NULL values.

The only change in this example will be to create a new table for the result instead of overwriting the table created when the activity was first executed from the GUI.

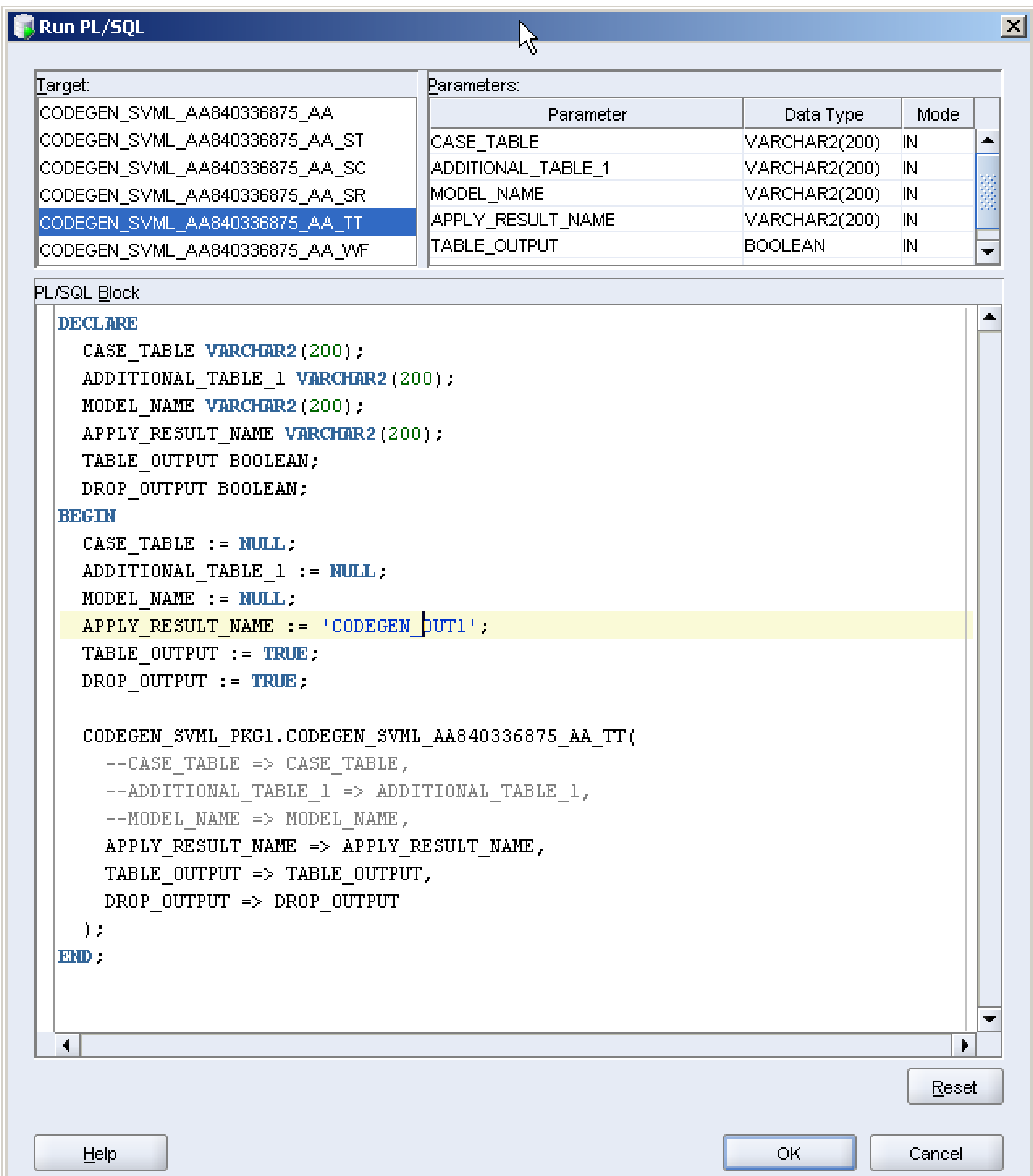
Here is the TT header information before any changes:



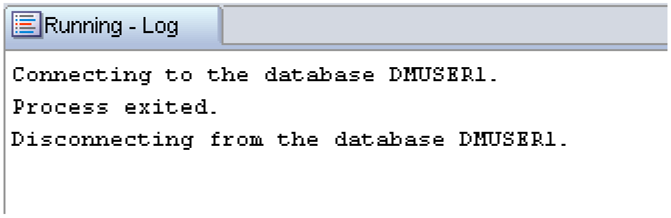
In order that the arguments CASE_TABLE, ADDITIONAL_TABLE1, and MODEL_NAME will have the values passed by the activity (rather than being passed NULL values), comment out the lines passing the NULL values by typing “ - - “ at the beginning of these three lines.

To change the output table name, enter the name, surrounded by single quotation marks, in the line assigning a value to APPLY_RESULT_NAME (in this case, ‘CODEGEN_OUT1’)

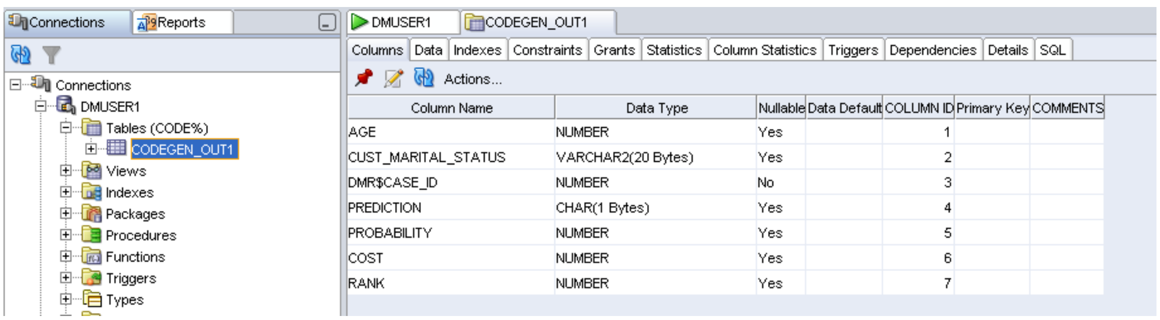
You want the example to produce a table that will overwrite any table with the same name, so replace the NULL value with TRUE for the logical variables TABLE_OUTPUT and DROP_OUTPUT.



When you click OK, the package executes, and you will see a message in the Log window:

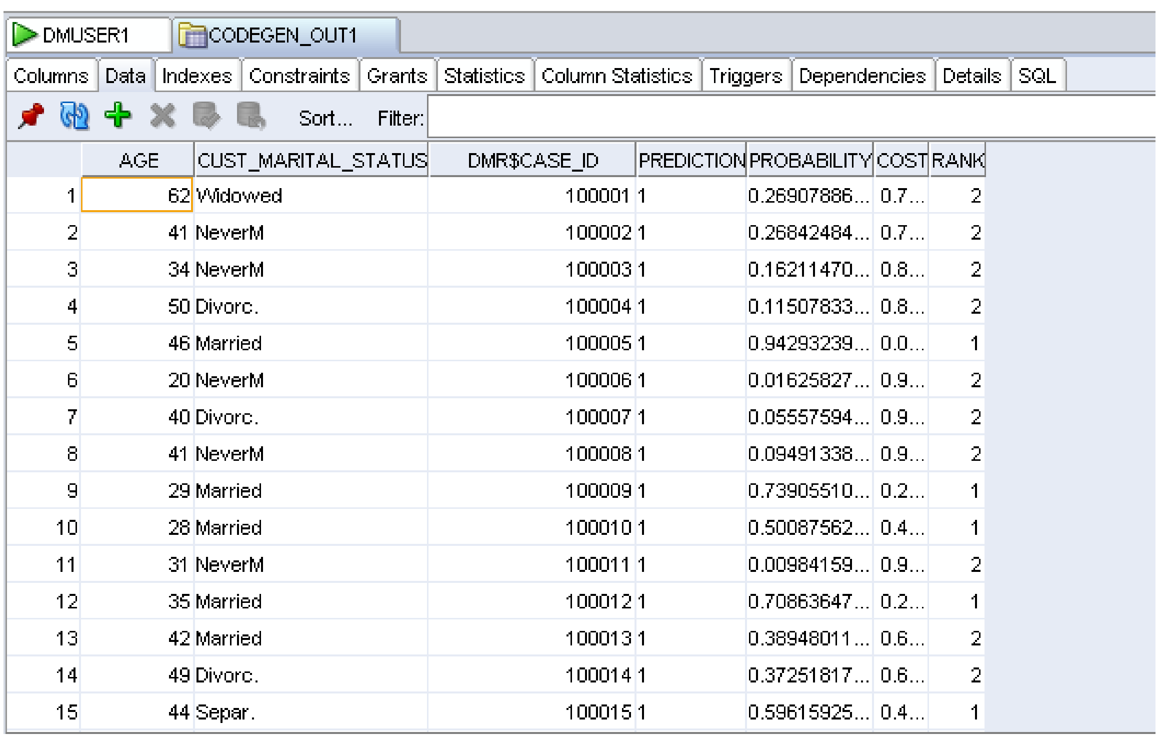


You will see the result table in the Tables tree, and you can display the table structure by clicking the table name.



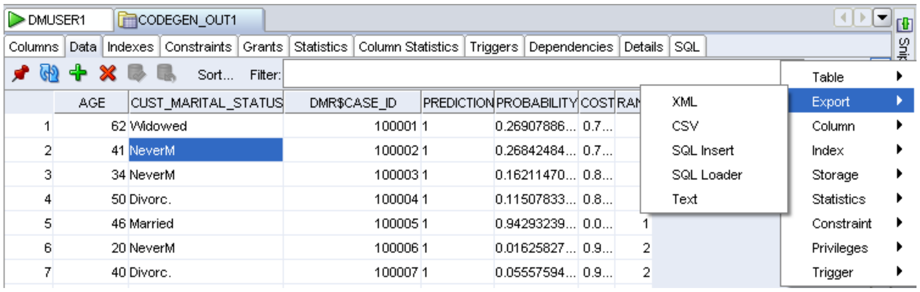
Column Name	Data Type	Nullable	Data Default	COLUMN ID	Primary Key	COMMENTS
AGE	NUMBER	Yes		1		
CUST_MARITAL_STATUS	VARCHAR2(20 Bytes)	Yes		2		
DMR$CASE_ID	NUMBER	No		3		
PREDICTION	CHAR(1 Bytes)	Yes		4		
PROBABILITY	NUMBER	Yes		5		
COST	NUMBER	Yes		6		
RANK	NUMBER	Yes		7		

If you click the Data tab, the contents are shown.



	AGE	CUST_MARITAL_STATUS	DMR$CASE_ID	PREDICTION	PROBABILITY	COST	RANK
1	62	Widowed	100001	1	0.26907886...	0.7...	2
2	41	NeverM	100002	1	0.26842484...	0.7...	2
3	34	NeverM	100003	1	0.16211470...	0.8...	2
4	50	Divorc.	100004	1	0.11507833...	0.8...	2
5	46	Married	100005	1	0.94293239...	0.0...	1
6	20	NeverM	100006	1	0.01625827...	0.9...	2
7	40	Divorc.	100007	1	0.05557594...	0.9...	2
8	41	NeverM	100008	1	0.09491338...	0.9...	2
9	29	Married	100009	1	0.73905510...	0.2...	1
10	28	Married	100010	1	0.50087562...	0.4...	1
11	31	NeverM	100011	1	0.00984159...	0.9...	2
12	35	Married	100012	1	0.70863647...	0.2...	1
13	42	Married	100013	1	0.38948011...	0.6...	2
14	49	Divorc.	100014	1	0.37251817...	0.6...	2
15	44	Separ.	100015	1	0.59615925...	0.4...	1

You can sort, filter, or modify the data in the result table, and there are many operations available by clicking the Actions button, such as Export, shown here.

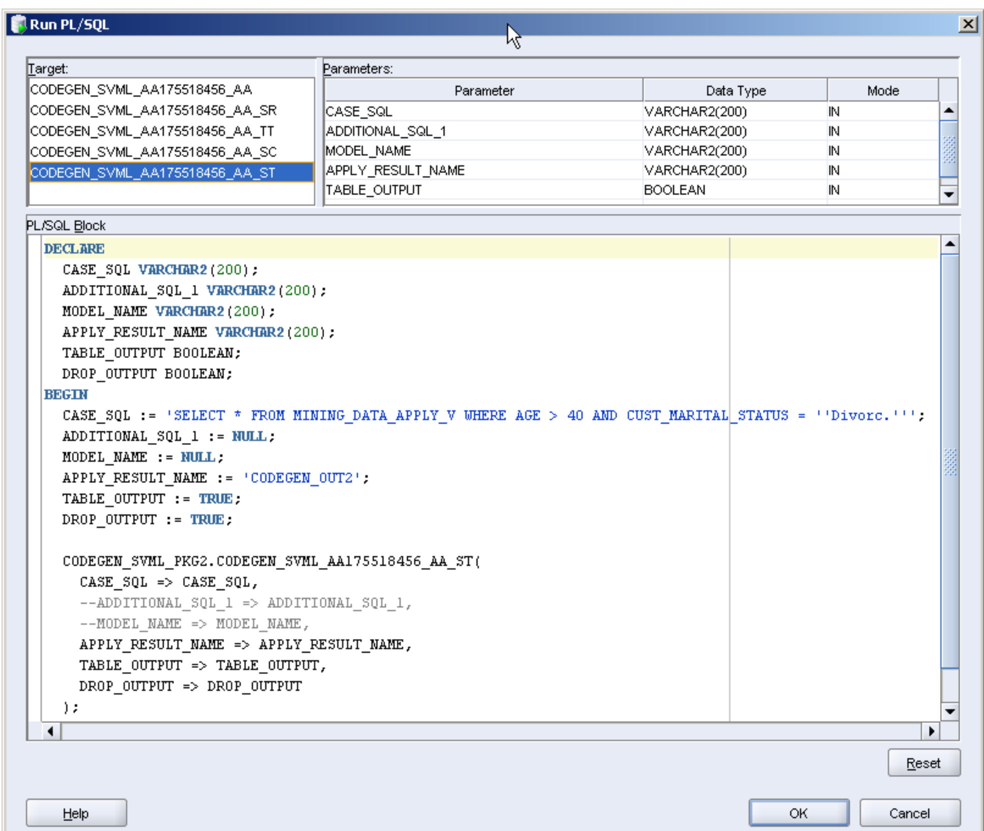


5. Use a SQL Query as Input Data Filter

You can use the result of a SQL query as input to the Apply operation, effectively filtering the data in any way you want.

Right-click the header file name and select Run, as in the previous example, but this time select the component with the suffix _ST (SQL query in, table out).

In this example, the input data is filtered to include only records of divorced individuals above the age of 40. Note that the query defined as CASE_SQL is surrounded by single quotation marks, and the categorical value Divorc. is surrounded by two single quotes.



Make other changes as shown and click OK; when execution completes, you can display the contents of the output table.

DMUSER1 | CODEGEN_OUT2

Columns | Data | Indexes | Constraints | Grants | Statistics | Column Statistics | Triggers | Dependencies | Details | SQL

Sort... Filter:

	AGE	CUST_MARITAL_STATUS	DMR$CASE_ID	PREDICTION	PROBABILITY	COST	RANK
1	48	Divorc.	100436	1	0.90843404634648...	0.0...	1
2	64	Divorc.	100759	1	0.81731242387763...	0.1...	1
3	54	Divorc.	100534	1	0.71971187292822...	0.2...	1
4	69	Divorc.	100092	1	0.70399918790796...	0.2...	1
5	42	Divorc.	100388	1	0.689629361233288	0.3...	1
6	41	Divorc.	100822	1	0.6106595550738021	0.3...	1
7	45	Divorc.	100393	1	0.59186693879379...	0.4...	1
8	58	Divorc.	100260	1	0.59054294725772...	0.4...	1
9	49	Divorc.	100295	1	0.53509245792323...	0.4...	1
10	51	Divorc.	100174	1	0.51095920247543...	0.4...	1
11	45	Divorc.	100563	1	0.49897916486154...	0.5...	2
12	65	Divorc.	100723	1	0.4786338453813711	0.5...	2
13	59	Divorc.	100731	1	0.46795974240889...	0.5...	2
14	45	Divorc.	100399	1	0.45954525633910...	0.5...	2
15	46	Divorc.	101496	1	0.44790563573967...	0.5...	2

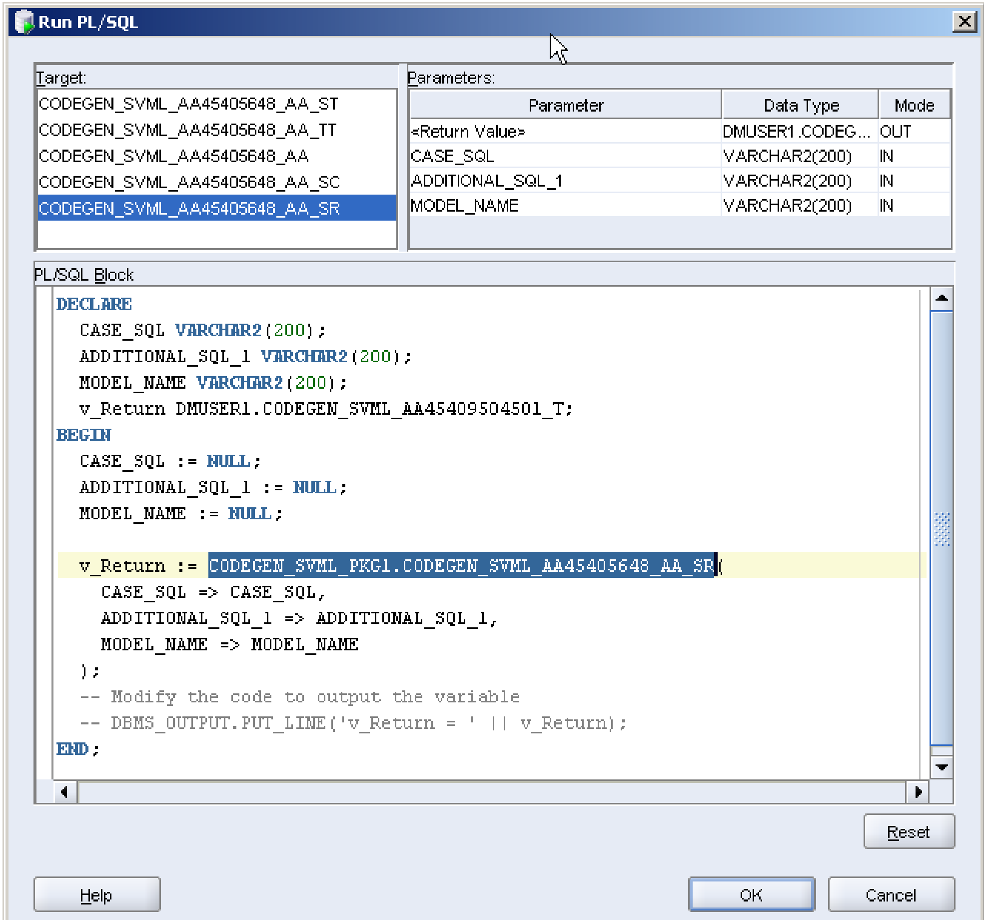
6. Deployment of Code to Score a Single Record

This example illustrates the use of the code generated by Oracle Data Miner as part of a process within an application.

Suppose that a Call Center application establishes the customer ID of a person who has called seeking information (ID number 100155 in this example), and passes that number to a PL/SQL package that executes and returns the probability that the customer in question fits the profile of someone likely to be a high lifetime value customer. The application can be written to suggest a course of action for the agent who is talking to the customer.

Right-click the header file name and select Run, as in the previous example, but this time select the component with the suffix _SR (SQL query in, record set out).

The full name of the API component is shown in the definition of v_Return.



This is the actual SQL function that will apply the model to the single record returned by a query – the arguments required by the function can be found in the code body.

Click the code body name and scan down through the code as far as the section labeled

/* Real time Apply, Input: SQL queries, Output: Result Set */

The code immediately following the section label gives the syntax for the function.

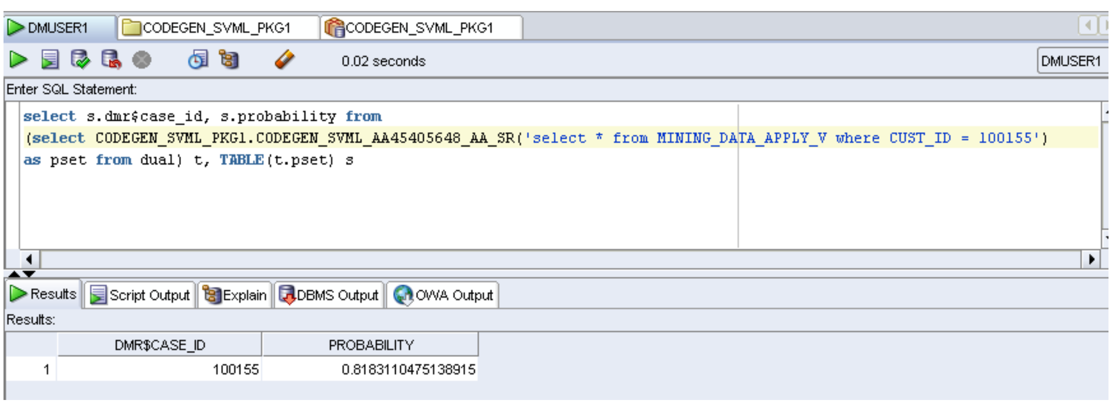
```
/* Real time Apply, Input: SQL queries, Output: Result Set */
FUNCTION "CODEGEN_SVML_AA45405648_AA_SR"(case_sql IN VARCHAR2,
                    additional_sql_l IN VARCHAR2 DEFAULT NULL,
                    model_name IN VARCHAR2 DEFAULT 'MINING_DATA_B31594_SV') RETURN  "CODEGEN_SVML_AA45409504501_T"
IS
  additional_sql  QUERY_ARRAY := QUERY_ARRAY(
    additional_sql_l
  );
  v_tempTables  TABLE_ARRAY := TABLE_ARRAY();
  v_cursor      NUMBER;
  v_feedback    INTEGER;
  v_AGE NUMBER;
  v_CUST_GENDER VARCHAR2(128);
  v_CUST_MARITAL_STATUS VARCHAR2(128);
  v_EDUCATION   VARCHAR2(128);
  v_DMR$CASE_ID NUMBER;
  v_PREDICTION  NUMBER;
  v_PROBABILITY NUMBER;
  v_COST        NUMBER;
  v_RANK        NUMBER;

  v_result      "CODEGEN_SVML_AA4540538204515_R";
  v_resultSet   "CODEGEN_SVML_AA45409504501_T" :=  "CODEGEN_SVML_AA45409504501_T"();
```

The first argument is the SQL code returning the single record to be scored (in single quotes), and the second two arguments are optional – if they are left blank, then the default values are NULL for *additional_sql_l*, and the name of the model applied in the Activity that generated this package for *model_name*. The names in the result set are listed after the function; in the example below, only the Case ID and the Probability are requested for display.

Now you can test a snippet of code as it will be embedded into the application that queries the result set of the function execution, as follows:

Click the tab containing the connection name (DMUSER1 in the example) to expose a SQL worksheet, and enter the SQL code as shown (In JDeveloper, right-click the connection name and select SQL Worksheet). Click the green diamond on the toolbar to execute. The result will appear in the Results window.



Appendix A – Installation and Configuration

Oracle Data Mining is an option of the Oracle 10*g* Release 2 Enterprise Edition database.

Installing the Database Disk

Refer to the *Installation Guide* for a particular platform to make note of platform-specific pre-installation and post-installation tasks. The example below shows the steps on a Windows system. The folder in which the database is installed is referred to as *ORACLE_HOME*.

From the Database Disk, there are two launching points for the installer: \database\setup.exe and \database\install\oui.exe. If you need to deinstall an existing database, you will use the installer oui.exe. The installer setup.exe allows a choice of Basic or Advanced installation path, whereas oui.exe uses only the Advanced path.

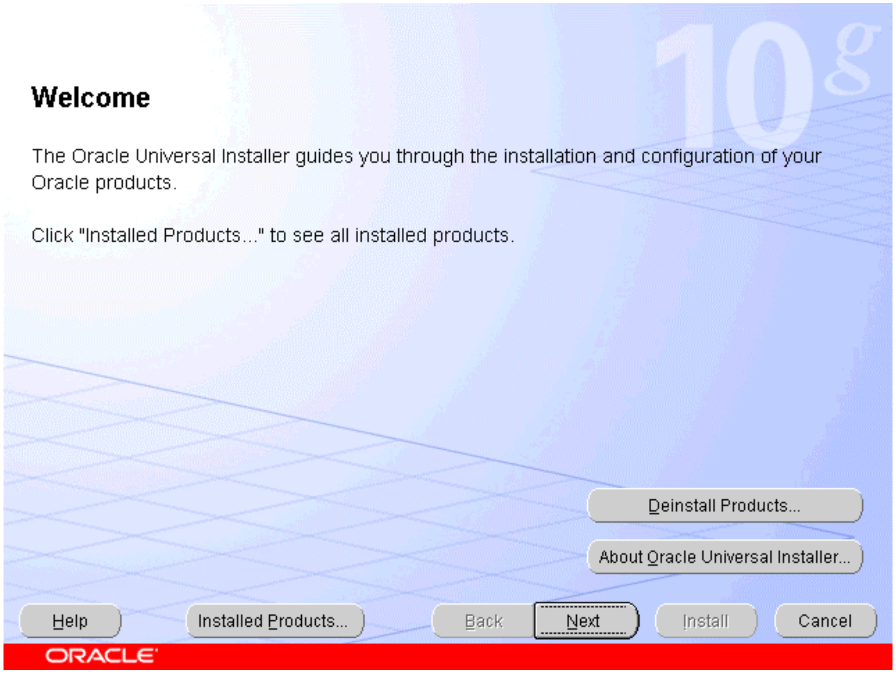
Note: Any installation of the Oracle Enterprise Edition database automatically installs Oracle Data Mining. If you perform a Custom Installation, do NOT select Data Mining Scoring Engine (DMSE is a limited version of Oracle Data Mining used in very specialized circumstances). Installing DMSE disables Oracle Data Mining.

You must disable any Oracle products before installation can begin. Go to Start → Settings → Control Panel → Administrative Tools → Services, then right-click any Oracle service that is running to Stop the service. Also from the Control Panel, choose System → Advanced → Environmental Variables to select and delete any variable with Oracle in the name.

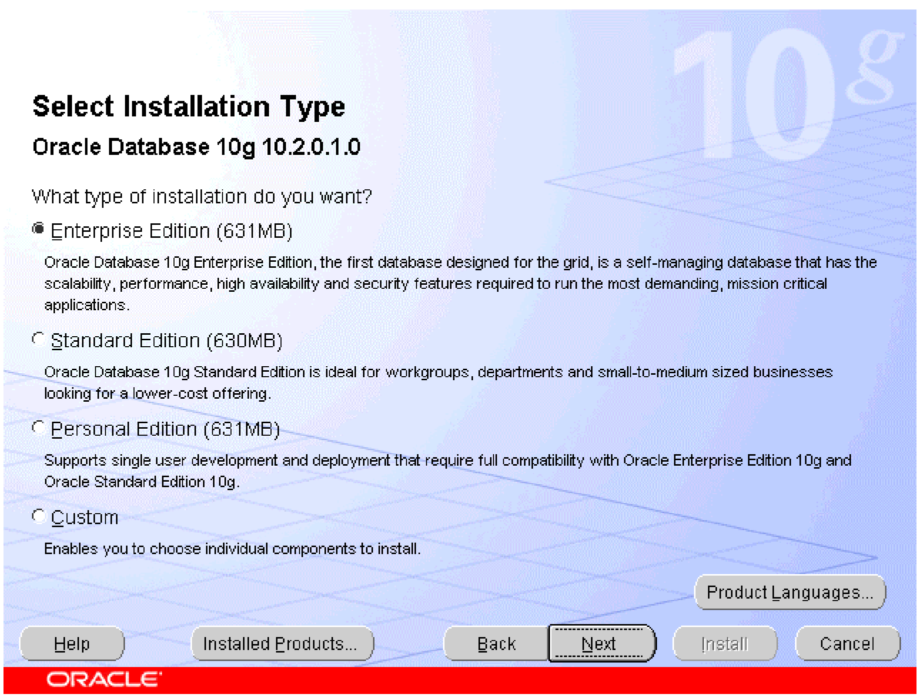
You may want to back up tables from an existing database before beginning.

The screens from oui.exe are shown in the example.

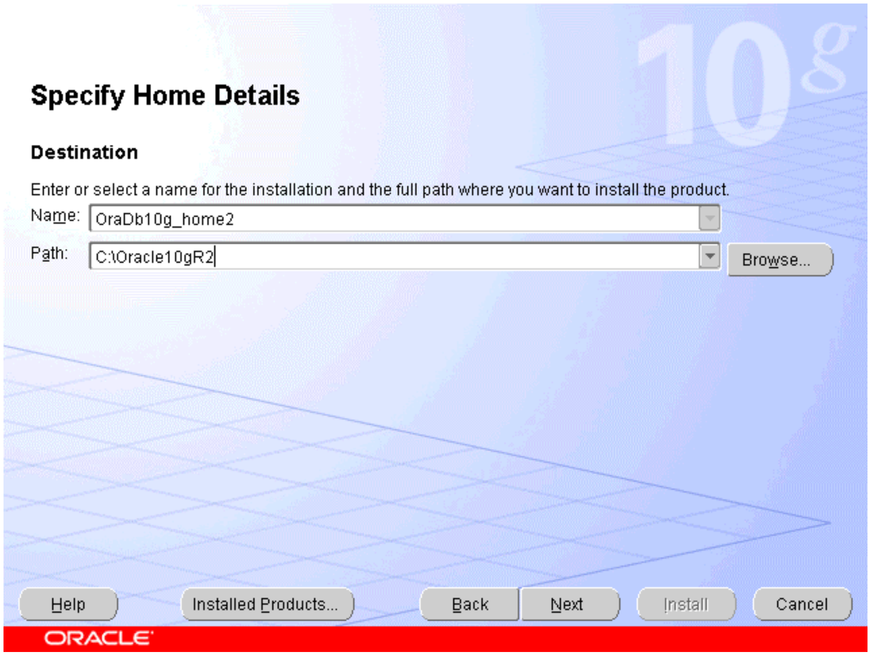
Double-click oui.exe to begin; if you need to deinstall a product, click Deinstall Products, select the product, and follow the instructions. Otherwise click Next.



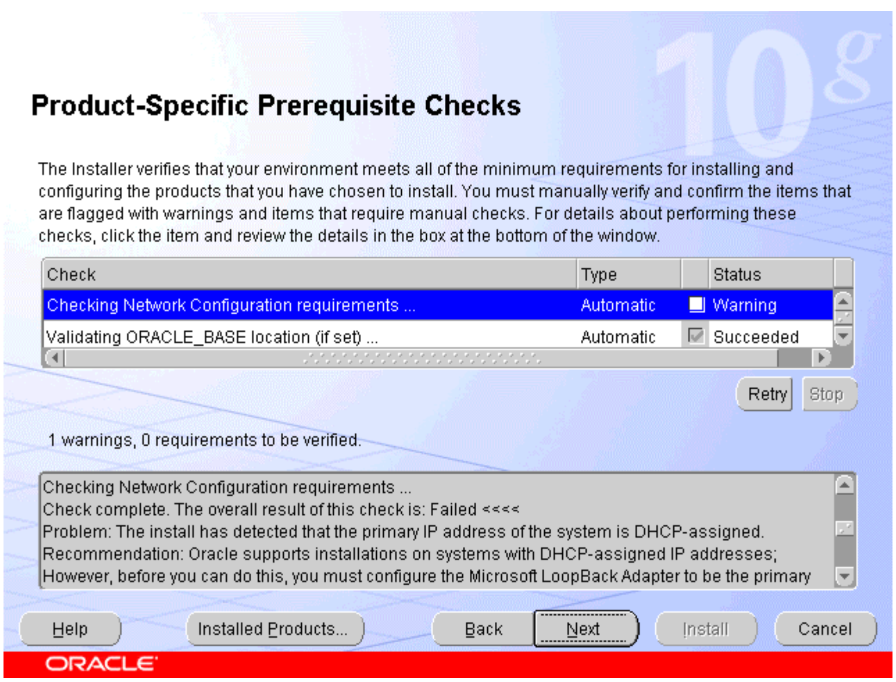
Oracle Data Mining requires the Enterprise Edition of Oracle Database 10g. Select the appropriate radio button and click Next.



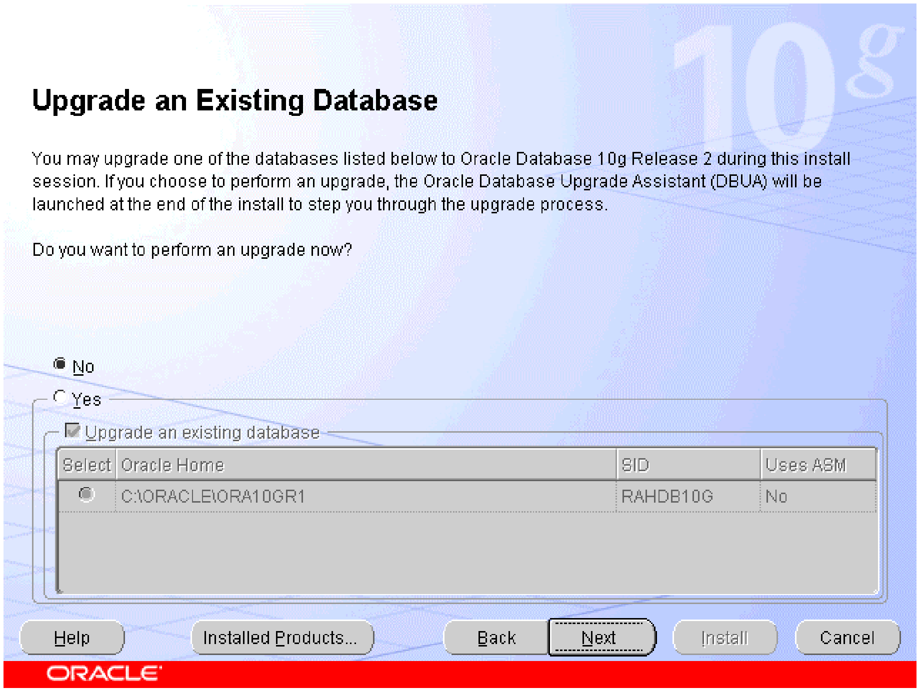
A default Name and a default Path are supplied; in this example, the Path (that is, *ORACLE_HOME*) has been simplified to identify a folder that had been created previously. Click Next



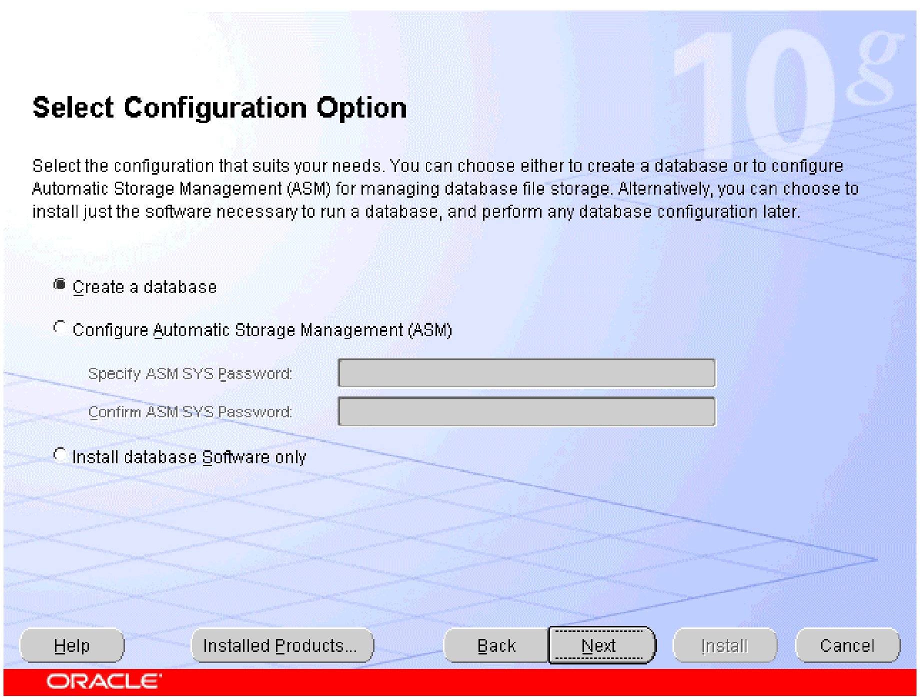
The following screen will appear only if the installation is being done on a system without a fixed IP address (for example, a laptop). The warning indicates that the database will not be accessible from a remote computer. As this is not usually an issue for a laptop (which will use “localhost” as the IP address for database access), click Next.



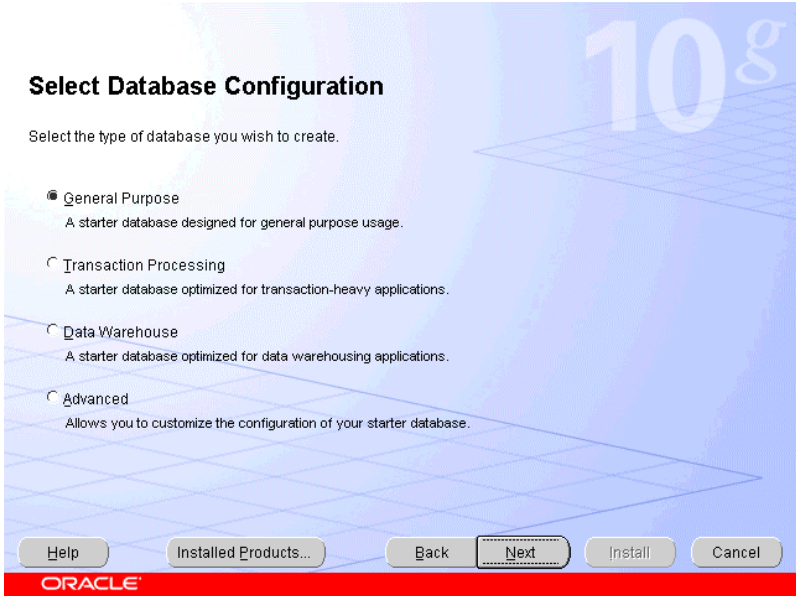
If an earlier Oracle database is detected, you have the opportunity to upgrade; in this example, select No and click Next.



Select Create a Database and click Next.



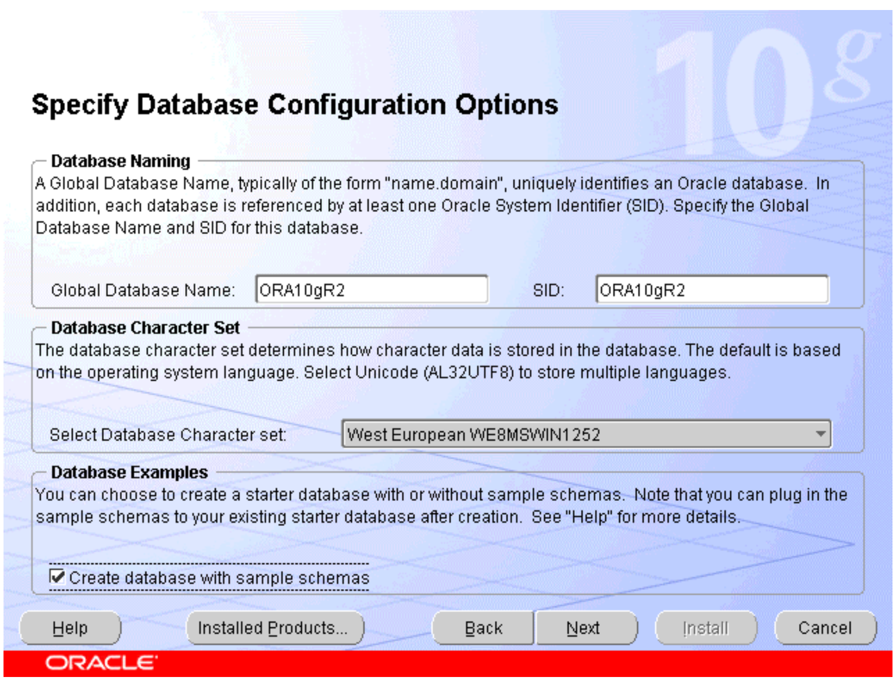
Select General Purpose and click Next.



Typically on a personal computer, the Global Database Name and the System Identifier (SID) are the same. Enter a name (and remember it!).

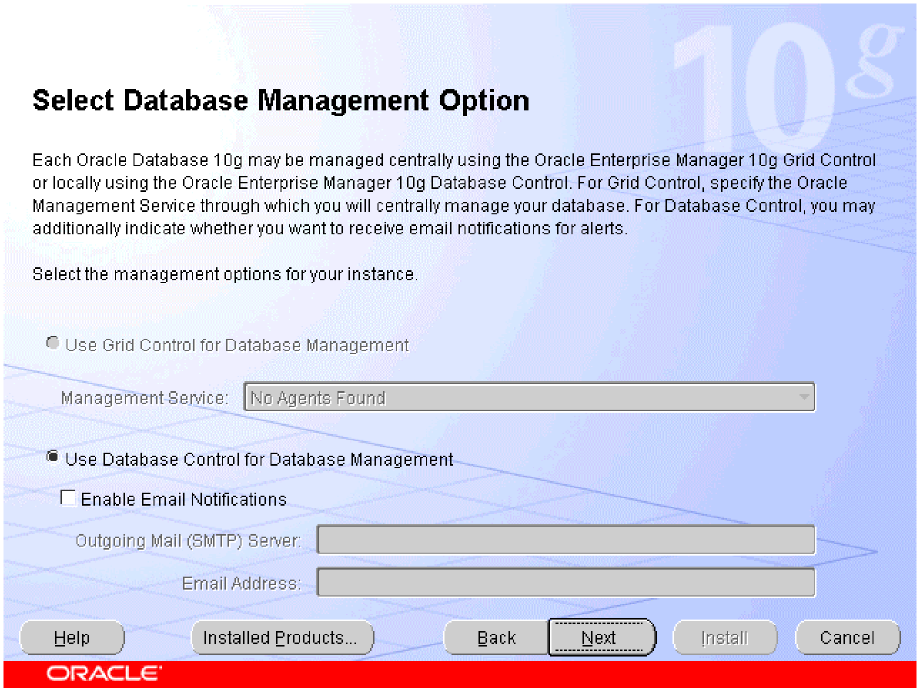
Normally, the character set is automatically selected based on the Operating System characteristics.

For the exercises in the Tutorial, you will need the sample schemas; ensure that the checkbox under Database Examples is selected, then click Next.

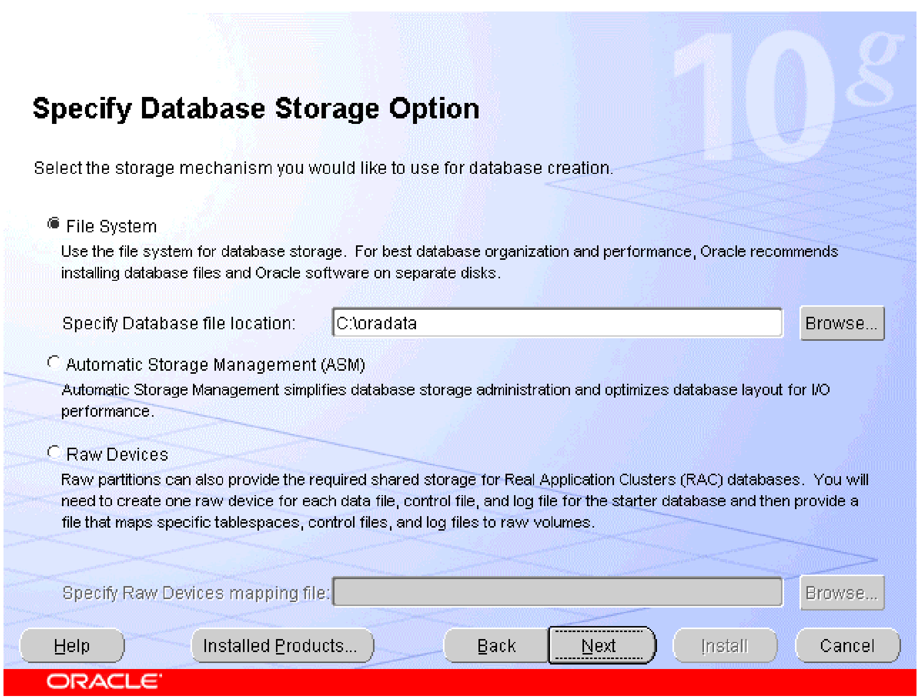


In the following screens, the simplest options are chosen.

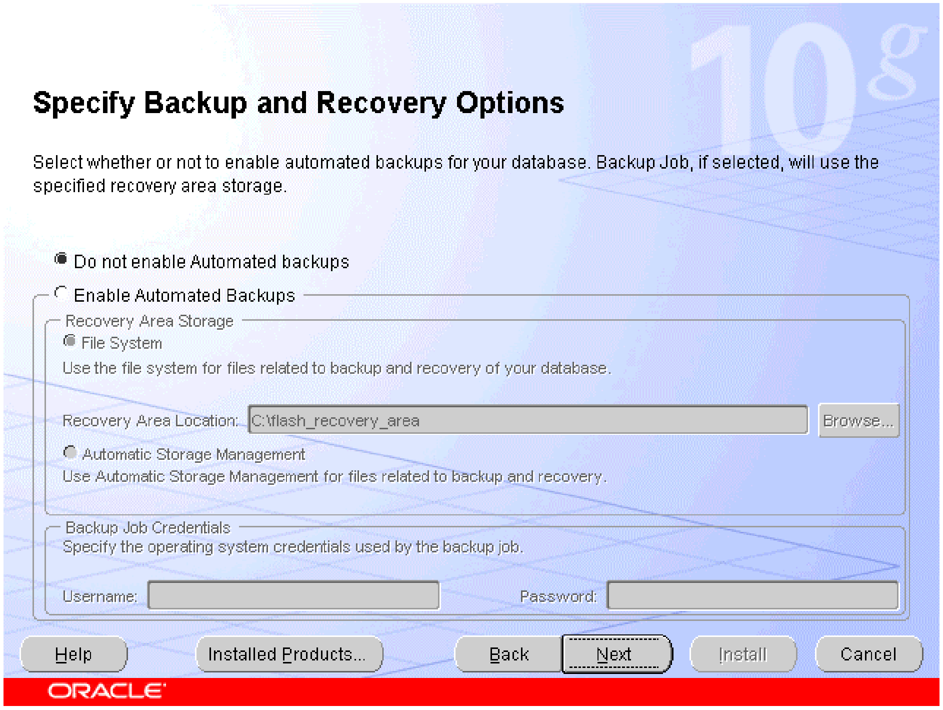
Select Use Database Control for Database Management and click Next.



Select File System for Storage Option and click Next.

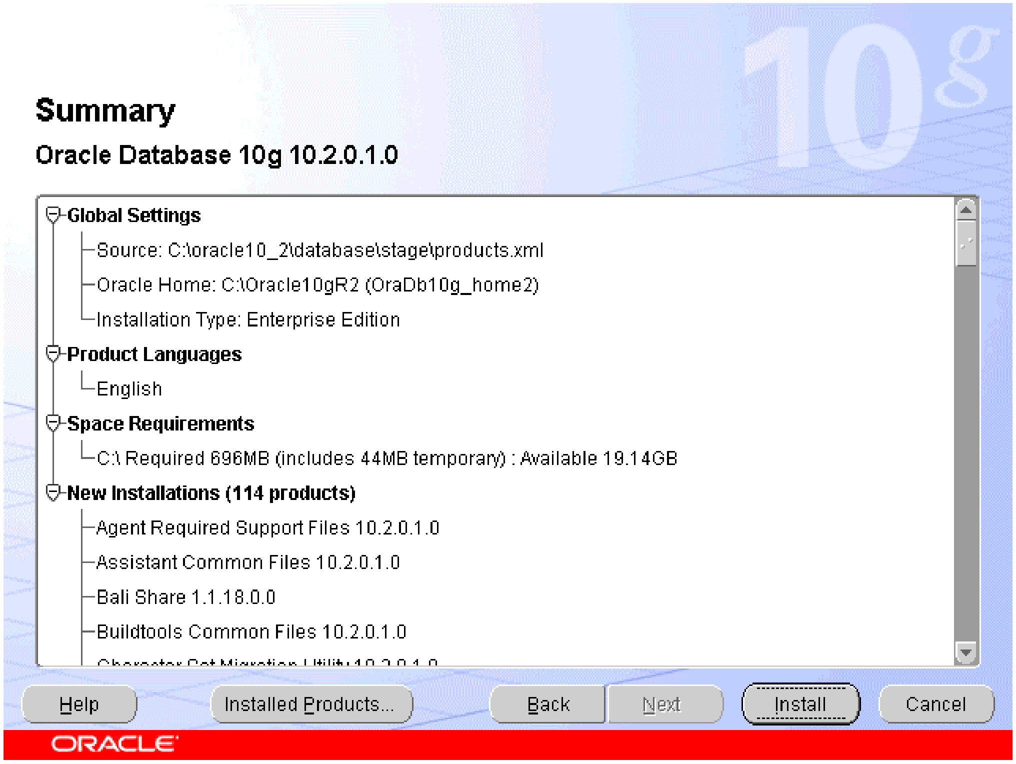


Select Do not enable Automated backups and click Next.

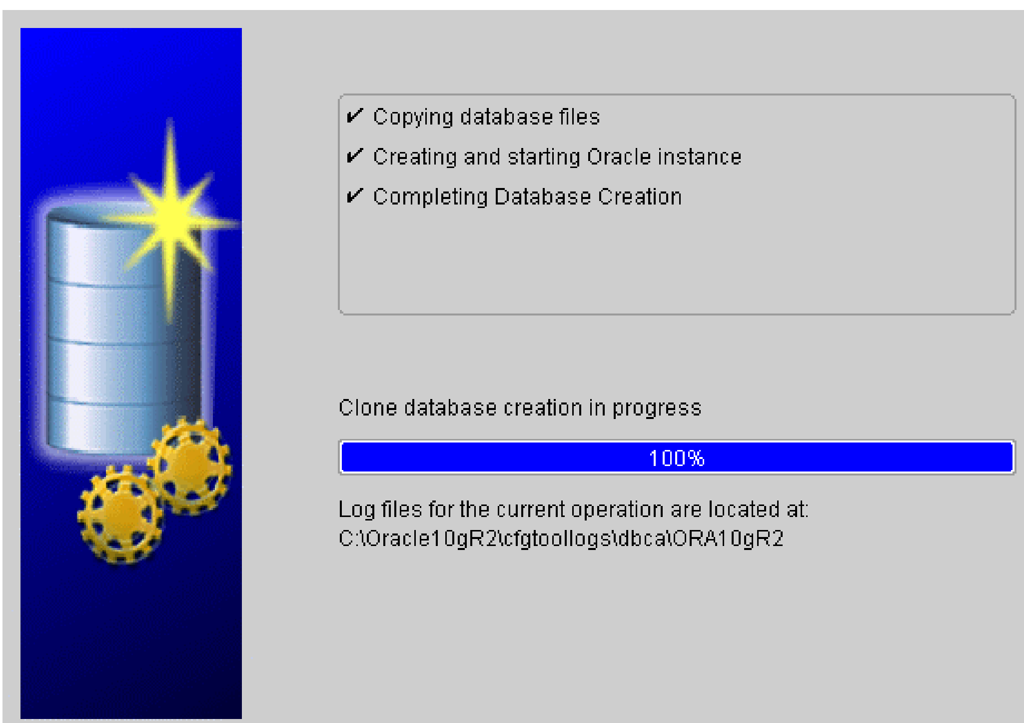


It’s convenient for all administrative user names to have the same password (especially on a personal computer). Select the appropriate radio button and enter the passwords, then click Next.

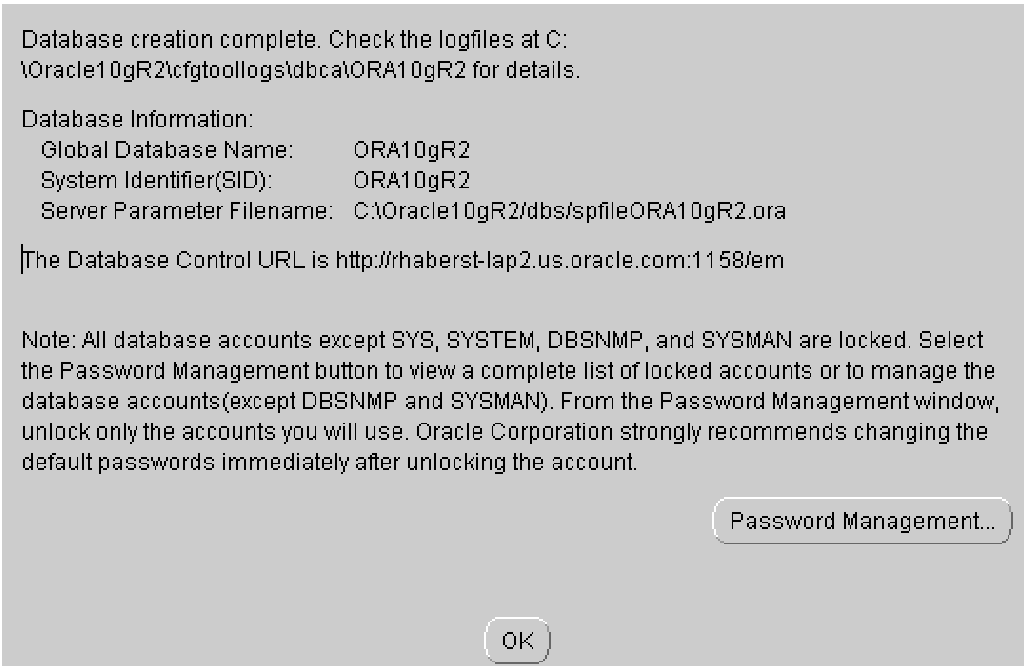
Review the Summary and click Install.



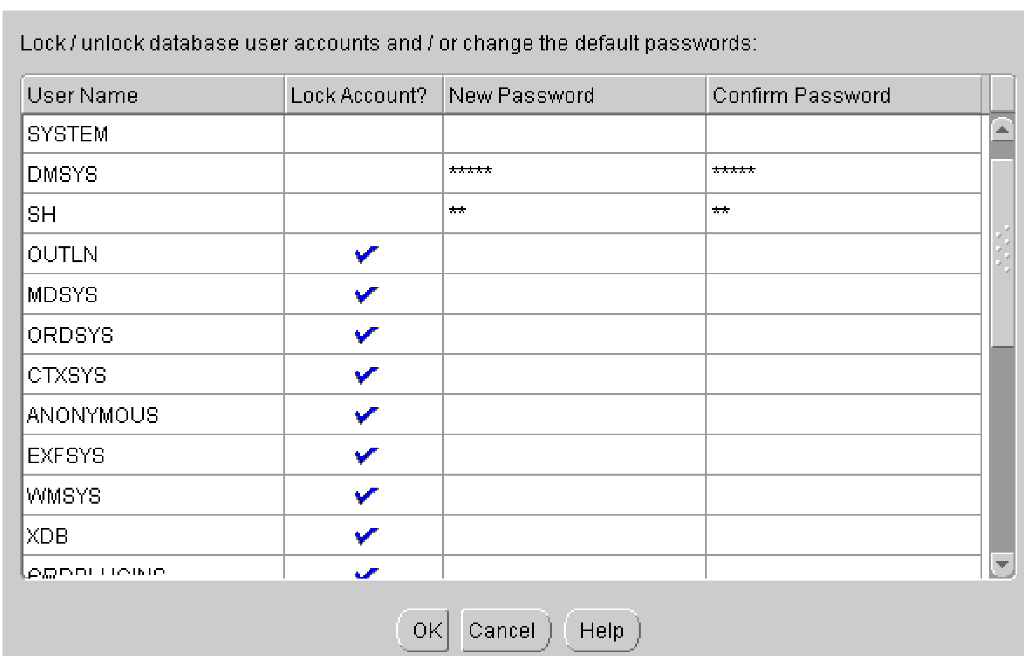
As the install proceeds, the progress is displayed.



When the database has been installed, a summary is displayed. Click Password Management to enable the Oracle Data Mining administrative function and to unlock schemas required for the data mining examples.



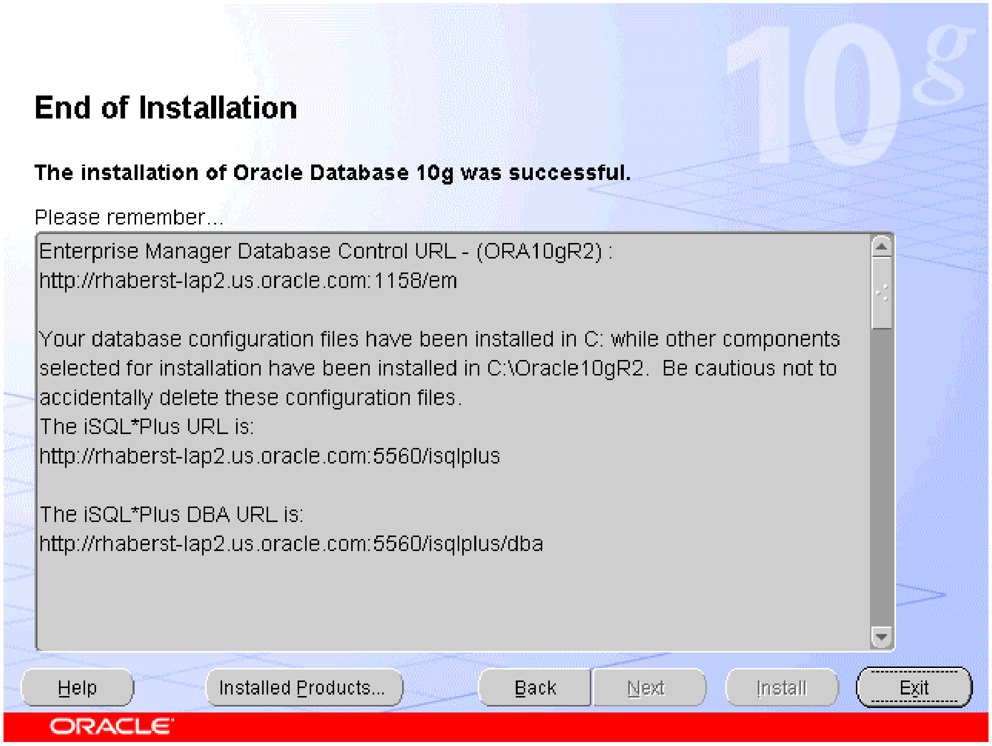
You must click on the checkmark and supply a password for the DMSYS account. To access the sample tables used in the Tutorial, do the same for SH. You may optionally choose to enable other accounts, such as SCOTT. When done, click OK.


Lock / unlock database user accounts and / or change the default passwords:

User Name	Lock Account?	New Password	Confirm Password
SYSTEM			
DMSYS		*****	*****
SH		**	**
OUTLN	✔		
MDSYS	✔		
ORDSYS	✔		
CTXSYS	✔		
ANONYMOUS	✔		
EXFSYS	✔		
WMSYS	✔		
XDB	✔		

OK Cancel Help

On the End of Installation screen, click Exit.



Adding the Data Mining Option to an existing 10.2.0.1.0 Database

If a custom installation was used to install and create a 10g Release 2 database and Oracle Data Mining was specifically excluded, you can add the option by using the Database Configuration Assistant (DBCA).

> **Test:** Log in to one of the unlocked accounts (for example SH) of the database using SQLPLUS and see if the connection information includes Oracle Data Mining:

```
Oracle Database 10g Enterprise Edition Release
10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options
```

To add the Oracle Data Mining option in a Windows environment, click ORACLE_HOME\BIN\dbca.bat to launch the wizard.

Select Configure Database Options, then check Oracle Data Mining on the Database Components screen, then Finish to enable the ODM option.

Verifying the Data Mining Option after Installation

If during installation the user SH was not unlocked, you can log in as sysdba and unlock and assign a password (for example SH) to the SH user.

```
SQL> ALTER USER SH IDENTIFIED BY “SH” ACCOUNT UNLOCK;
```

The following command will verify the correct installation of ODM:

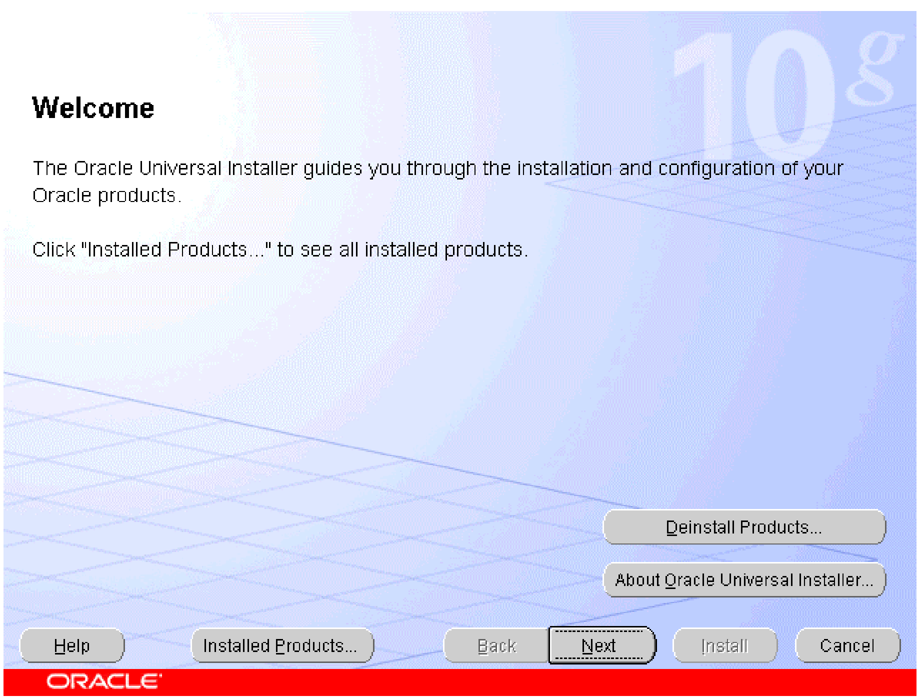
```
SQL> select comp_name, status from dba_registry;

COMP_NAME      STATUS
--------------------          ------------
               .
               .
Oracle Data Mining       VALID
               .
               .
```

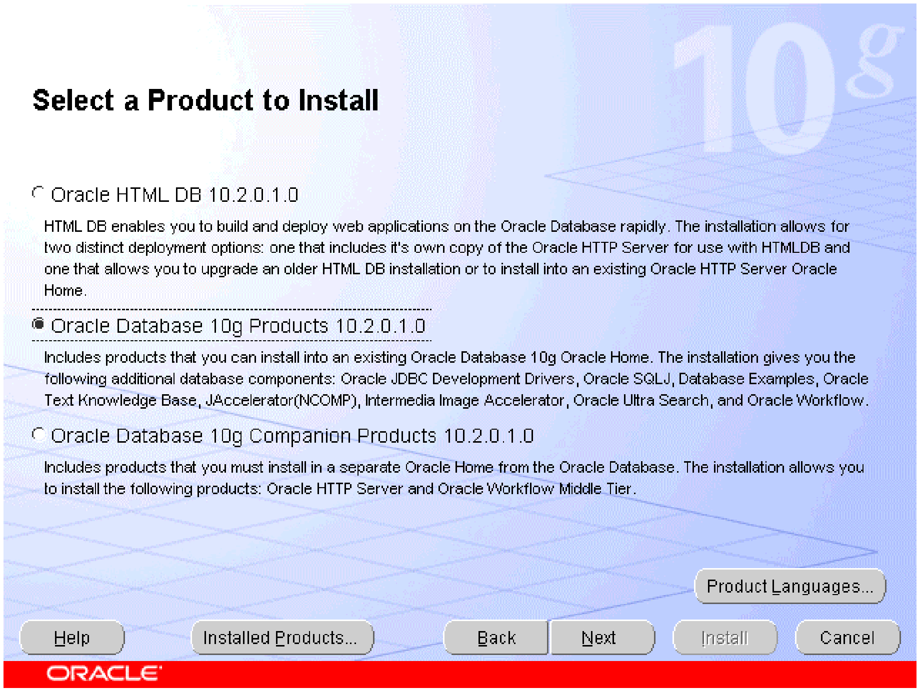
Installing the Companion Disk

To copy the sample PL/SQL and Java data mining programs into the folder ORACLE_HOME\RDBMS\Demo, you must install the Companion CD.

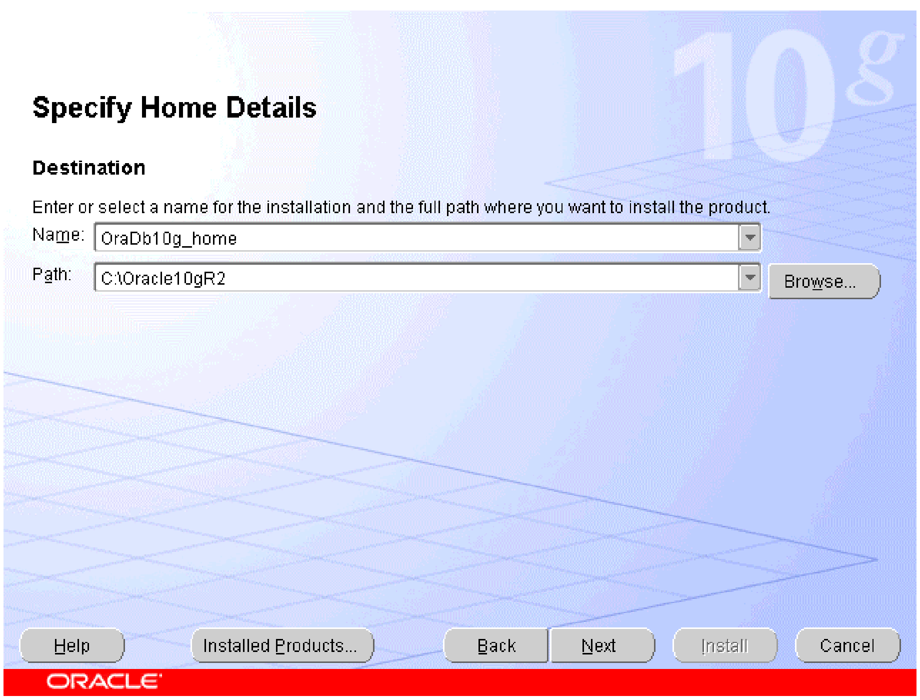
From the Companion Disk, double-click setup.exe to launch the installer. Click Next to begin.



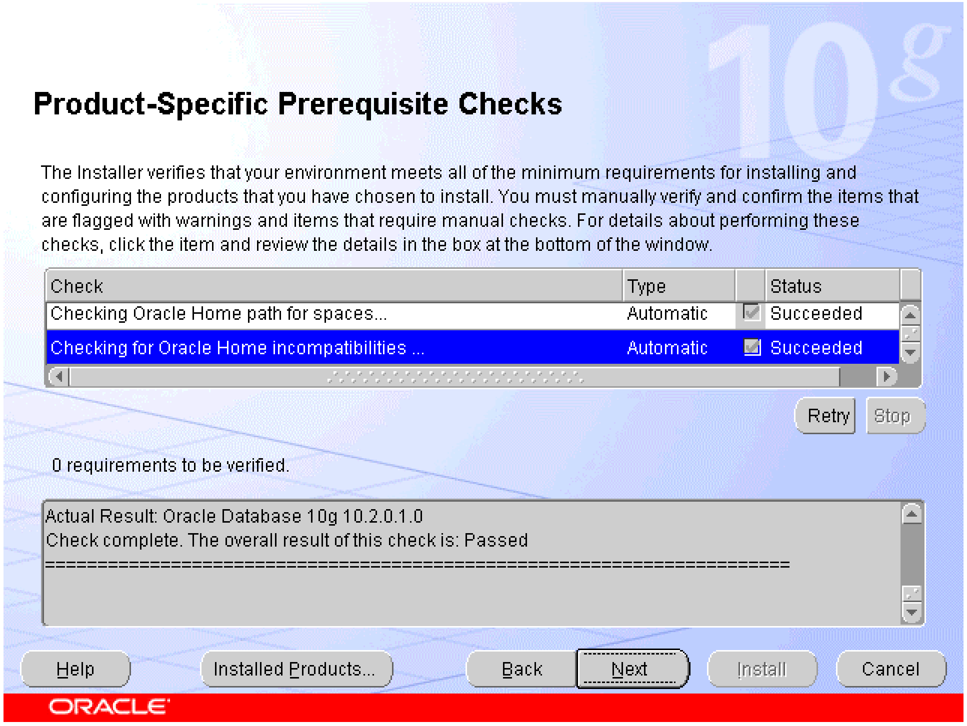
In order to install the ODM sample programs, you must select Oracle Database 10g Products 10.2.0.1.0 and click Next.



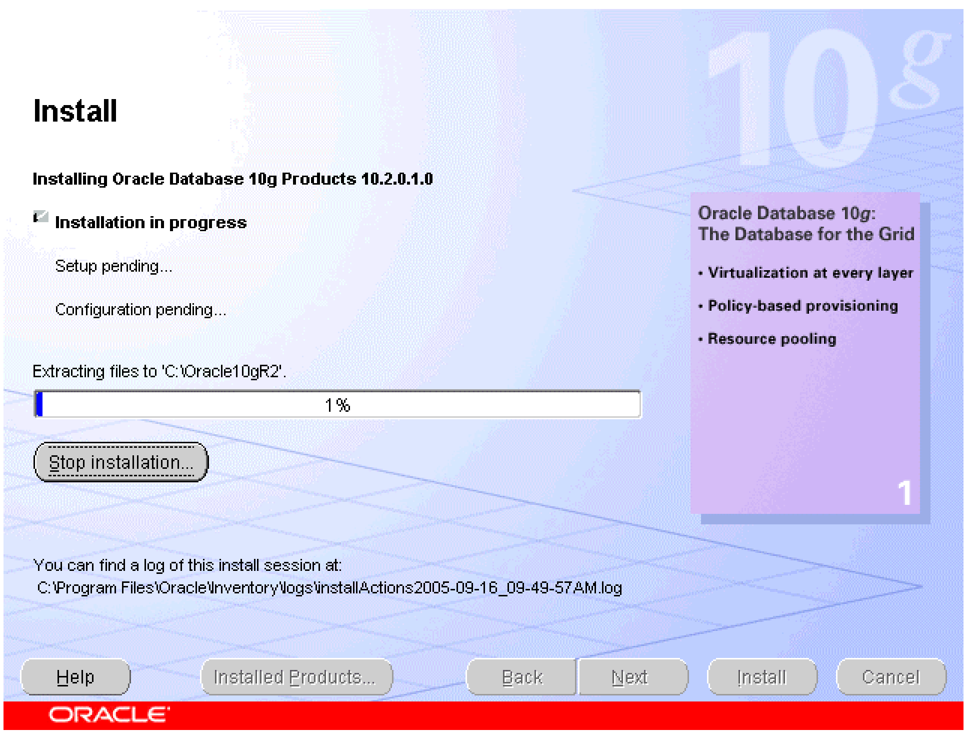
The Name and Path for the Home Details must match the Name and Path entered when the database was installed. Enter the names and click Next.



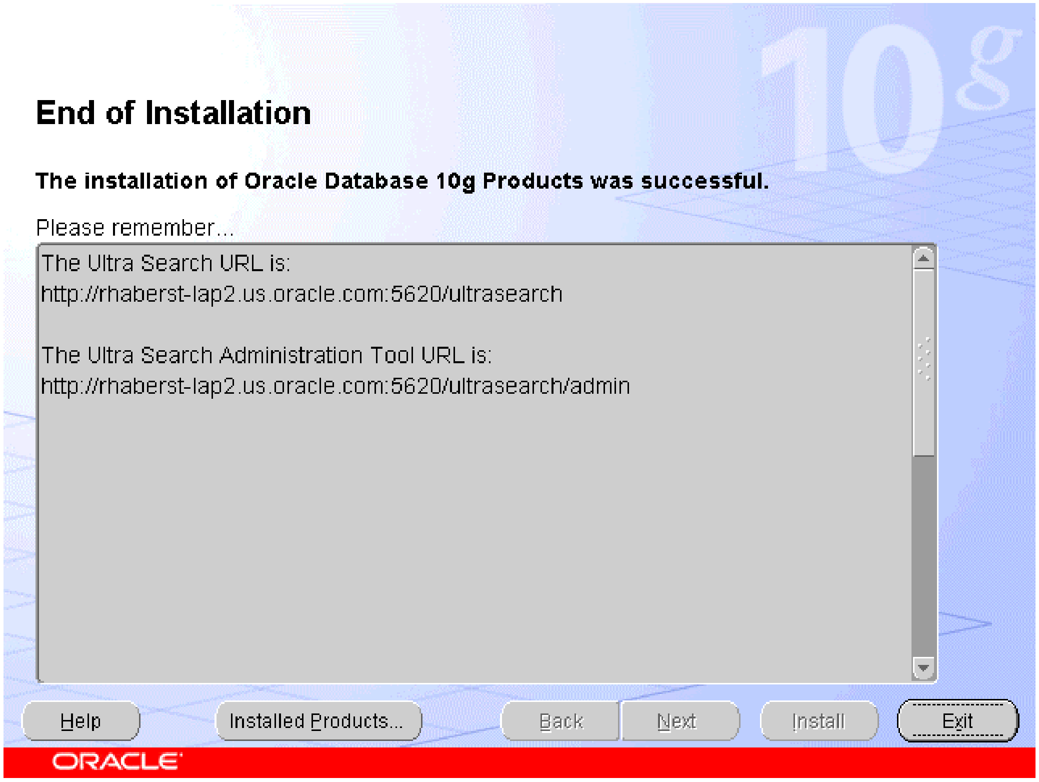
Verify that the Checks succeeded and click Next



Confirm the Summary page and click Install. The progress during installation is displayed. When done, click Next.



At the end of installation click Exit.



Creating Oracle Data Mining Users

Each database user who will execute ODM operations must have:

- default and temporary tablespaces specified
- permission to access the mining data

A user on a personal computer (where only one user is active), or users in a training environment (where only small sample tables will be used) can be assigned an existing tablespace (for example USERS). Under any circumstances, users can share an existing temporary tablespace (for example TEMP).

In a production setting with several users, it is better to create separate tablespaces for each user.

First, on the command line, change directory to the folder containing the administrative scripts, *ORACLE_HOME*\RDBMS\demo.

```
C:\>cd \Oracle10gR2\RDBMS\demo
```

Then, log into the database as sysdba and find the existing tablespaces and their locations (if you want to create new ones):

```
C:\Oracle10gR2\RDBMS\demo>sqlplus sys/oracle@ora10gr2
as sysdba

SQL*Plus: Release 10.2.0.1.0 - Production on Tue Sep
27 14:32:40 2005

Copyright (c) 1982, 2005, Oracle.  All rights
reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release
10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> select tablespace_name, file_name from
dba_data_files;

TABLESPACE_NAME     FILE_NAME
---------------     -----------------
USERS          C:\ORADATA\ORA10GR2\USERS01.DBF

SYSAUX         C:\ORADATA\ORA10GR2\SYSAUX01.DBF

UNDOTBS1       C:\ORADATA\ORA10GR2\UNDOTBS01.DBF

                    .
                    .
                    .
```

Now you know the full path name for the location of the tablespaces, so if you want to create new tablespaces for the data mining user(s) rather than use the existing ones, you can enter commands to create the default and temporary tablespaces as follows:

```
SQL> CREATE TABLESPACE dmuser1 DATAFILE
'C:\oradata\ORA10gR2\dmuser1.dbf' SIZE 20M REUSE
AUTOEXTEND ON NEXT 20M;

Tablespace created.
```

Now you can create the data mining user(s), making reference to the tablespace created with the previous command.

```
SQL> CREATE USER dmuser1 IDENTIFIED BY dmuser1 DEFAULT
TABLESPACE dmuser1 TEMPORARY TABLESPACE temp QUOTA
UNLIMITED ON dmuser1;

User created
```

In a training environment where several users will access small tables on the same database server, you can create DMUSER1, DMUSER2, etc. and use the existing tablespaces without having to create new tablespaces. For example,

```
SQL> CREATE USER dmuser2 IDENTIFIED BY dmuser2 DEFAULT
TABLESPACE users TEMPORARY TABLESPACE temp QUOTA
UNLIMITED ON users;

User created
```

Next, the user(s) must be granted permissions to carry out data mining tasks, to access the Oracle Text package, and also to access the sample data in the SH schema. The script `dmshgrants.sql` accomplishes these tasks, using as inputs the password for SH and the password for the user being granted permissions.

```
SQL> @dmshgrants sh dmuser1
old   1: GRANT create procedure to &DMUSER
new   1: GRANT create procedure to dmuser1

Grant succeeded.

old   1: grant create session to &DMUSER
new   1: grant create session to dmuser1

Grant succeeded.

old   1: grant create table to &DMUSER
new   1: grant create table to dmuser1

Grant succeeded.

     .
     .
     .

SQL>
```

Finally, the schema for each new user can be populated with tables and views constructed from the data in the SH schema; each new user must do this individually, using the `dmsh.sql` script.

```
SQL> connect dmuser1
Enter password:
Connected.
SQL> @dmsh


View created.


View created.


View created.


View created.

     .
     .
     .
```

Installing the Oracle Data Miner User Interface in a Windows environment

Create a folder (*not* in *ORACLE_HOME*) that will serve as home for the Graphical User Interface Oracle Data Miner (ODMr), for example C:\ODMr10_2, then browse to

http://www.oracle.com/technology/products/bi/odm/odminer.html

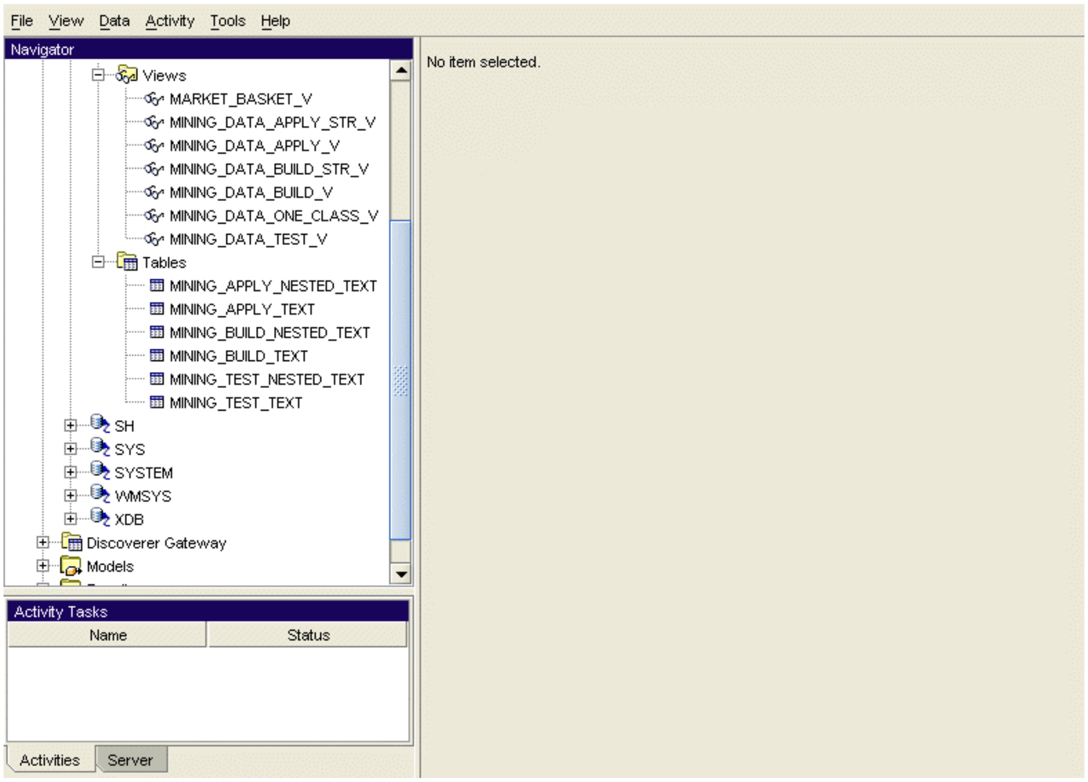
right-click and choose Save Link Target as … to download the zipfile odminer.zip into the folder created above.

Unzip the contents into C:\ODMr10_2 and double-click C:\ODMr\bin\odminerw.exe to launch the GUI (you may want to create a shortcut to odminerw.exe on your desktop).

There will be no database connection when ODMr is first launched, so click New to create a connection to a specific database user/schema.

Enter a meaningful name for the connection, the user/password assigned to you, the name or IP address of the database server (if the GUI and server are on the same system, you can use the host name localhost), and the listener port and the SID that were established during installation.

When the Oracle Data Miner opens, you can click the “+” next to Data Sources, then your user name, then Views and Tables to confirm that the sample tables and views are in your user schema. You should see the names as displayed below.



Installing the Oracle Data Miner User Interface in a MAC OS environment

The program requires Java JDK 1.4.2, included in Mac OS X 10.4.5. To check the version of Java, open a terminal session (using the Terminal application) and use the command:

```
java –version
```

Browse to

http://www.oracle.com/technology/products/bi/odm/odminer.html

right-click and choose Save Link Target as … to download the zipfile odminer.zip.

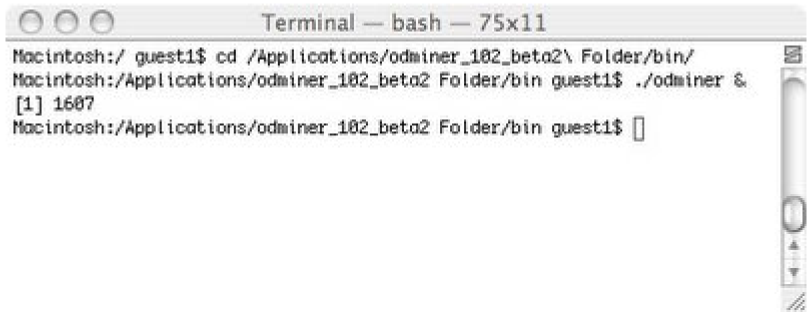
Unzip `odminer.zip` by double clicking on odminer.zip. This creates the folder `odminer` (in the current working folder) and inflates the archive into it. (For the Oracle Data Miner beta version, it creates the `odminer_102_beta2 Folder`.)

Move the created folder to the desired location, for example to the `Applications` folder.

To start Oracle Data Miner, open a `Terminal` shell and change directory to `MINER_HOME/bin`, where `MINER_HOME` is the directory where Oracle Data Miner is installed. In this example, `MINER_HOME` is `/Applications/odminer_102_beta2 Folder.` Reset the permissions to add execute permission to the script `odminer:`

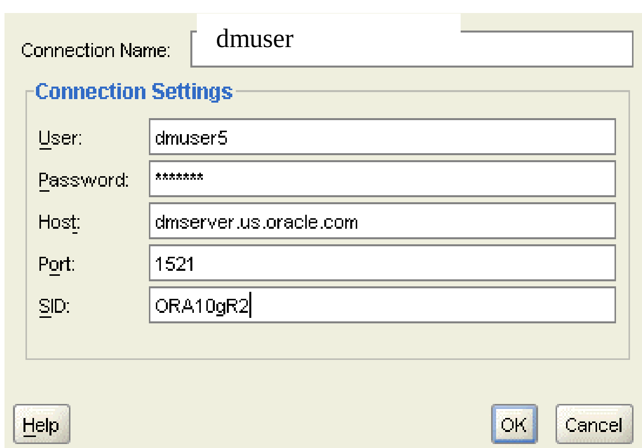
```
chmod +x odminer
```

Execute the script odminer to launch the Oracle Data Miner; you may use "&" after the command to run in the background.

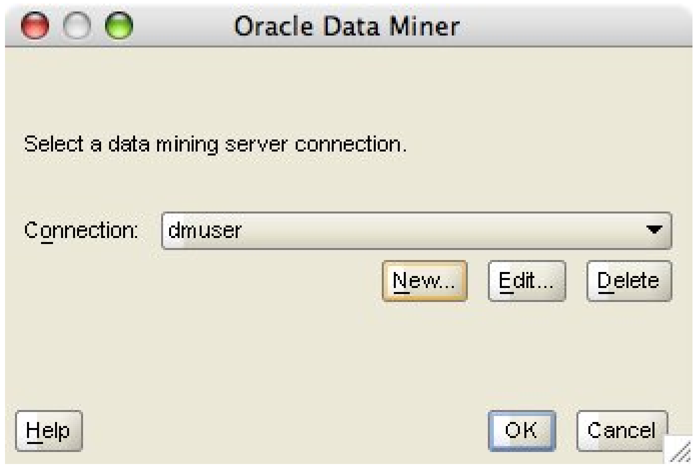


There will be no database connection when ODMr is first launched, so click New to create a connection to a specific database user/schema.

Enter a meaningful name for the connection, the user/password assigned to you, the name or IP address of the database server (if the GUI and server are on the same system, you can use the host name localhost), and the listener port and the SID that were established during installation.

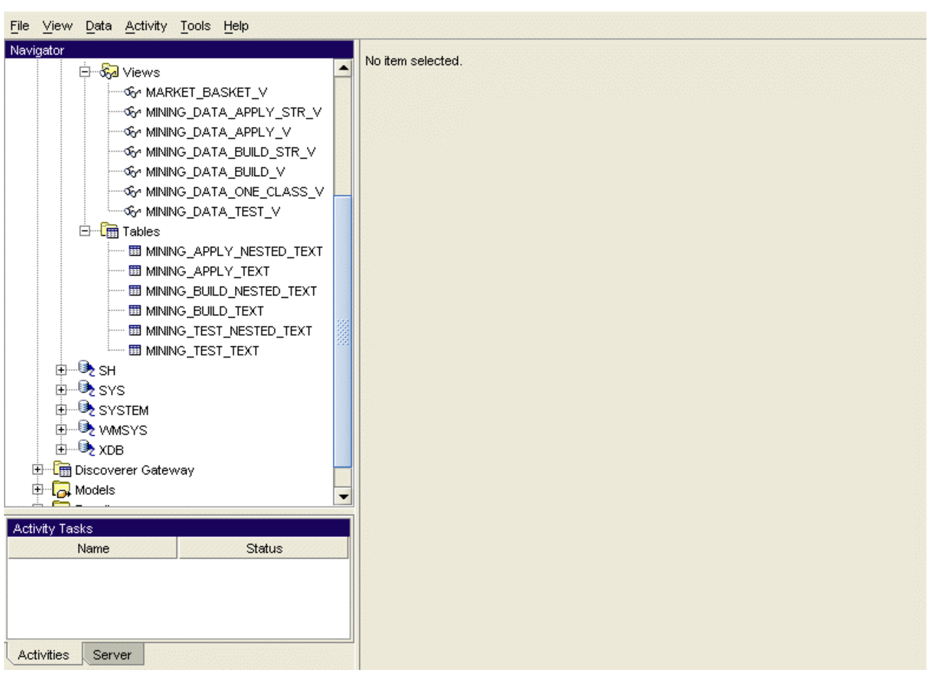


Click OK when you finish the definition. You are returned to the Choose Connection dialog. You can now select the connection that you just defined from the dropdown box.



Click OK to bring up the Oracle Data Miner main screen.

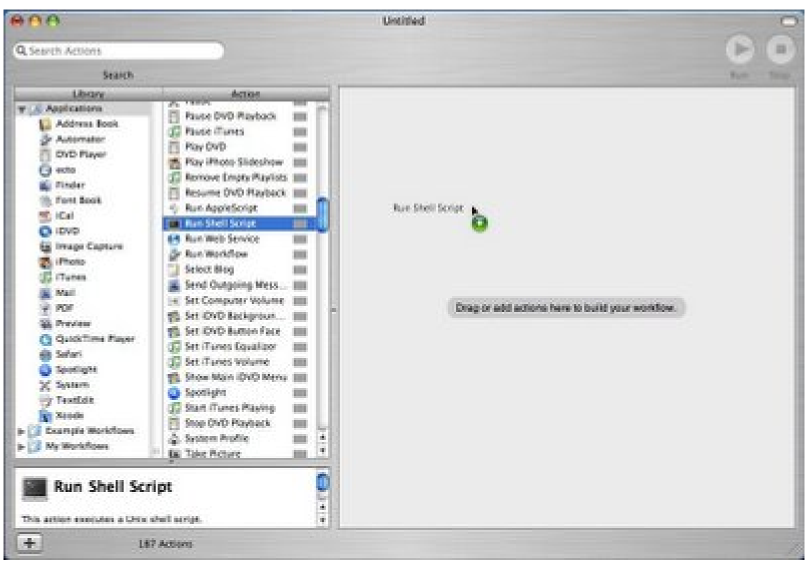
When the Oracle Data Miner opens, you can click the "+" next to Data Sources, then your user name, then Views and Tables to confirm that the sample tables and views are in your user schema. You should see the names as displayed below.



You can run Oracle Data Miner as an application from the Mac OS graphical interface by writing a shell script using Automator In Mac OS 10.4.

Select Automator:

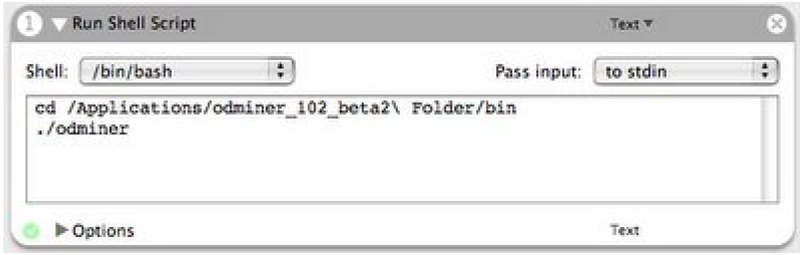
then select Run Shell Script from the Action list and drag it to the left pane.



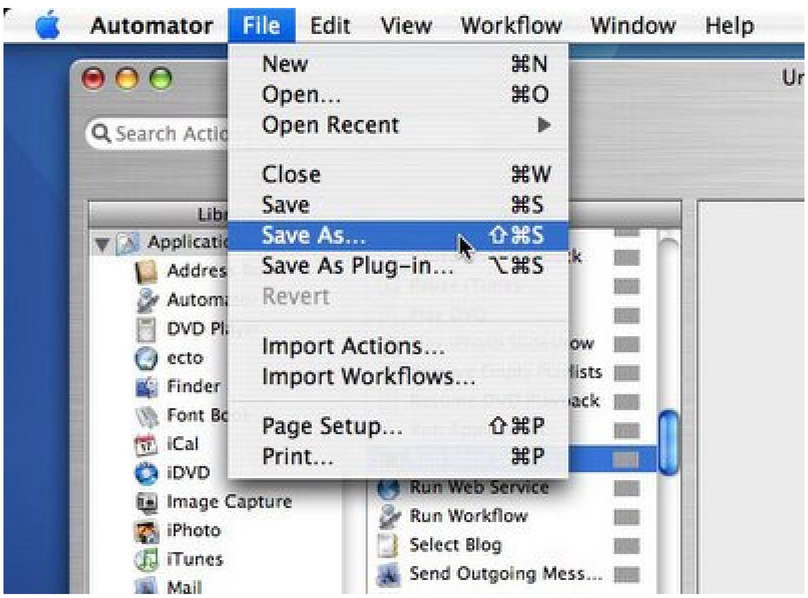
Next enter the following lines in the Run Shell Script text box:

```
cd MINER_HOME/bin
./odminer
```

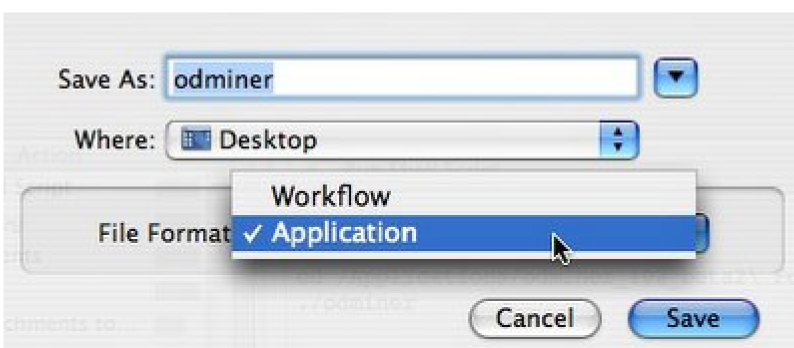
where, in the example, `MINER_HOME` is
`/Applications/odminer_102_beta2 Folder`.



Save the script using the File - Save As... menu option



Select a name for the application and `Application` for File Format and you can now launch Oracle Data Miner by double clicking the icon for the new application.

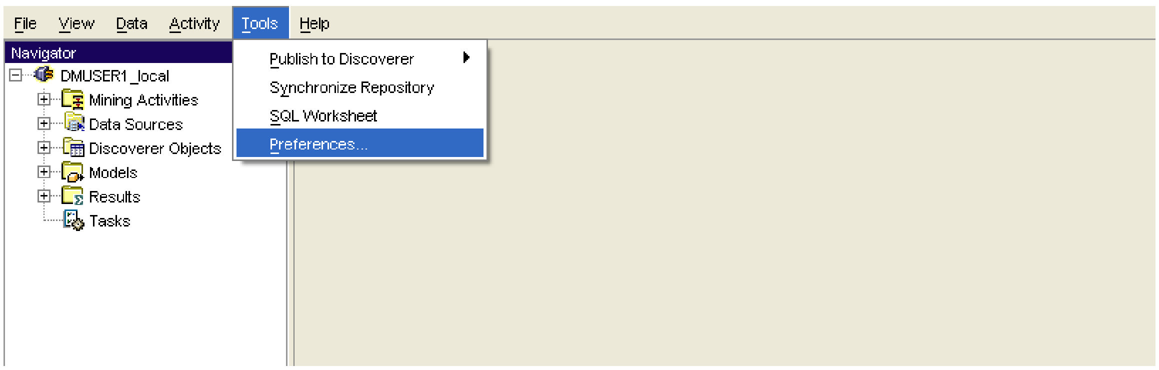


Appendix B – Setting Preferences

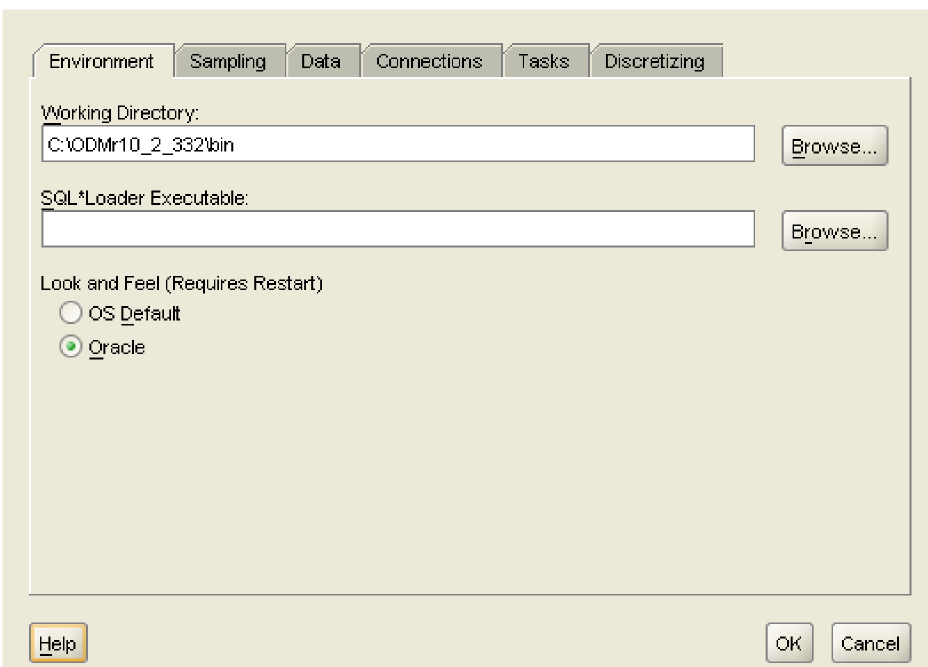
You can set preferred values for several types of actions that take place in Oracle Data Miner.

Note: There are occasions when an activity will override the user-defined preference settings in order to conform to algorithm-specific requirements.

To begin, select Preferences on the Tools pull-down menu.

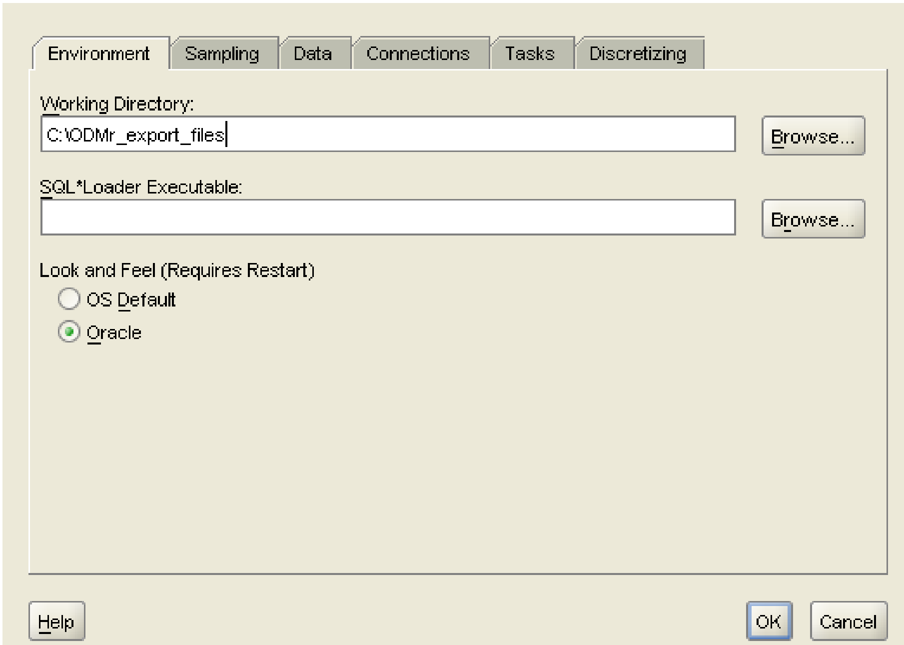


Each tab represents a type of setting.

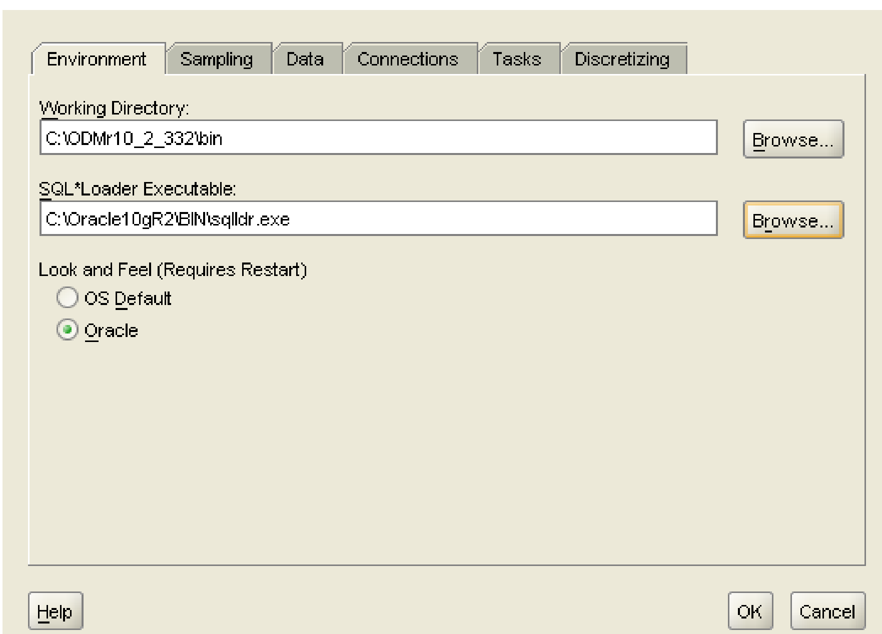


The Working Directory is the destination directory when data is exported from Oracle Data Miner, such as when the  icon is used to export data to a textfile. The default value is the bin directory under the directory in which Oracle

Data Miner was installed. It is suggested that you use a directory that will not be affected if a new version of Oracle Data Miner is installed, as shown below:

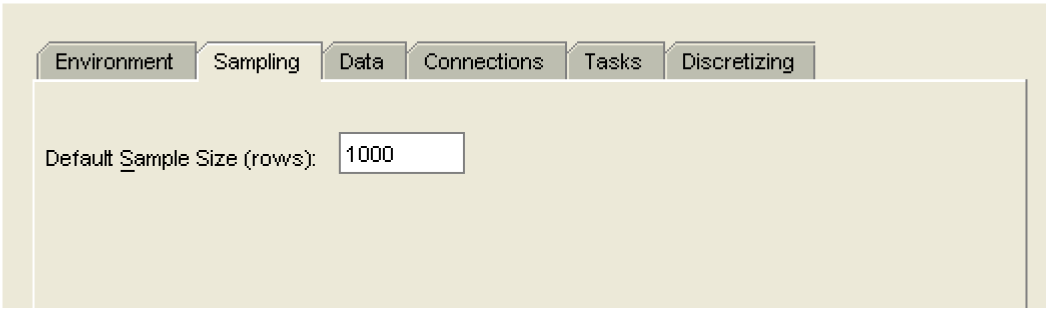


The SQL*Loader Executable is required only if you intend to use the Data→Import wizard. If you have either the Database Server or the Database Client (with the Administrator option) installed on the same system as Oracle Data Miner, then SQL*Loader is found in the BIN directory under Oracle_Home.



The Look and Feel radio button selects a certain appearance for the GUI.

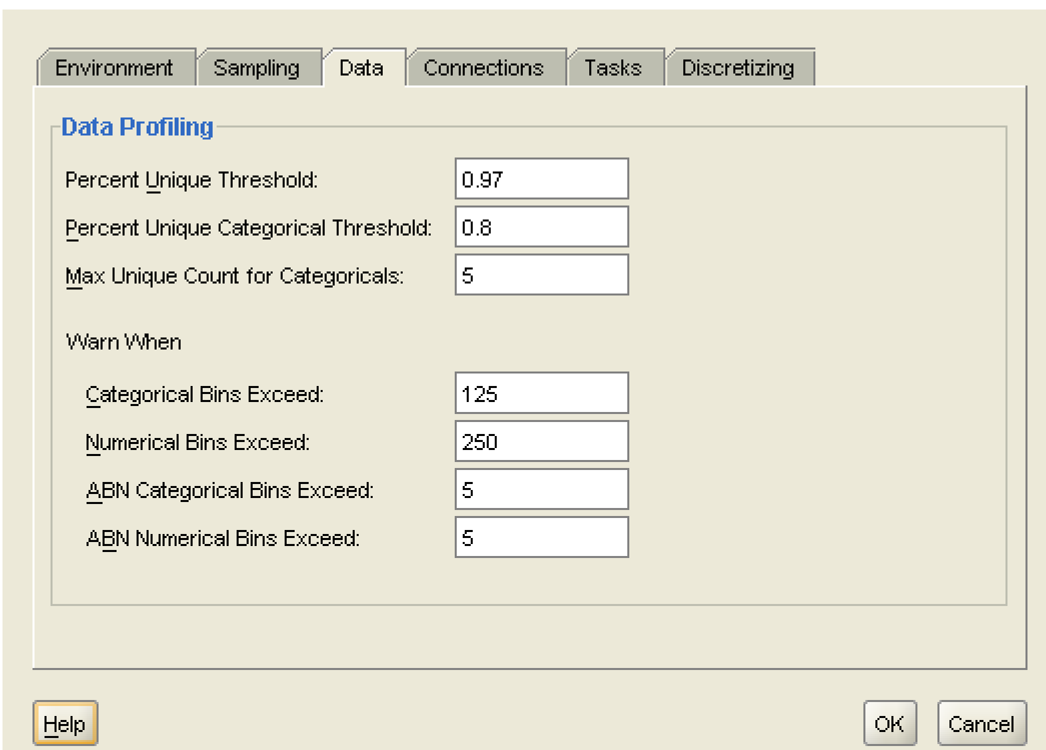
The Statistical Summary, as well as some other displays, is based on a random sample of rows in the table or view. The default sample size is 1000; click the Sample tab and enter a value to change the default.



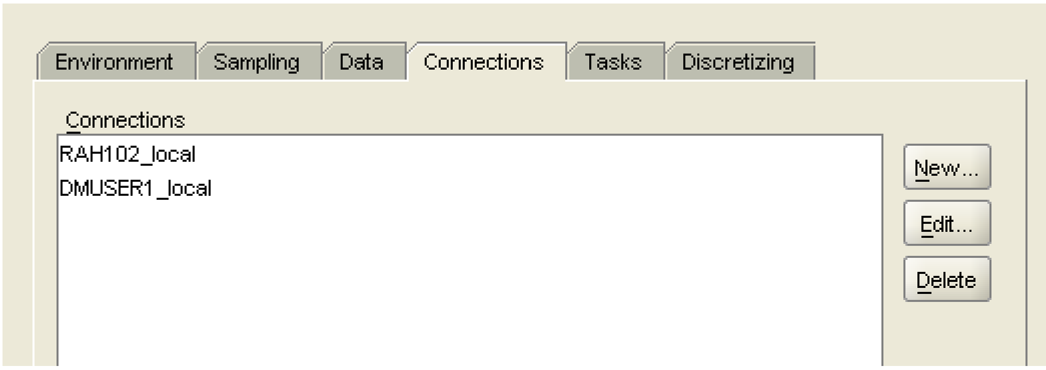
Internally, certain criteria are used to establish numerical attributes as either NUMERIC or CATEGORICAL, by percentage of distinct values or by number of distinct values. These decisions determine, for example, what method is used for automatic binning.

You will receive a warning when the number of bins exceeds the default values shown (for reasons of performance); the ABN algorithm, being very sensitive to a high number of bins, is a special case.

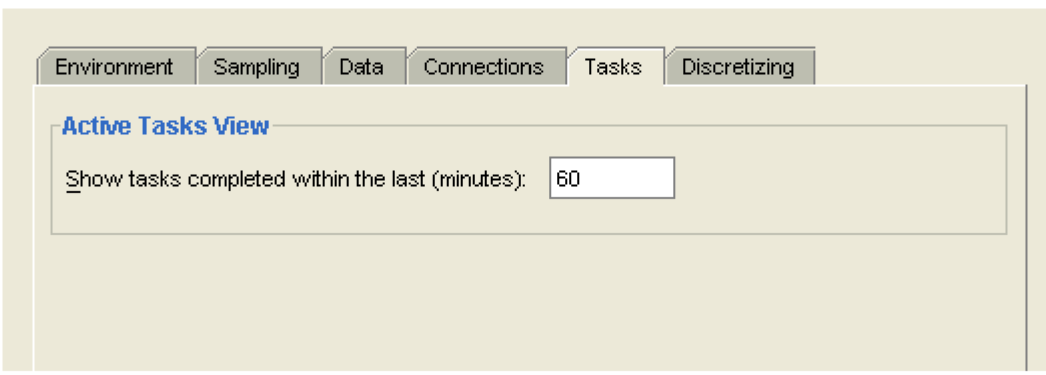
You can click the Data tab to enter new default values.



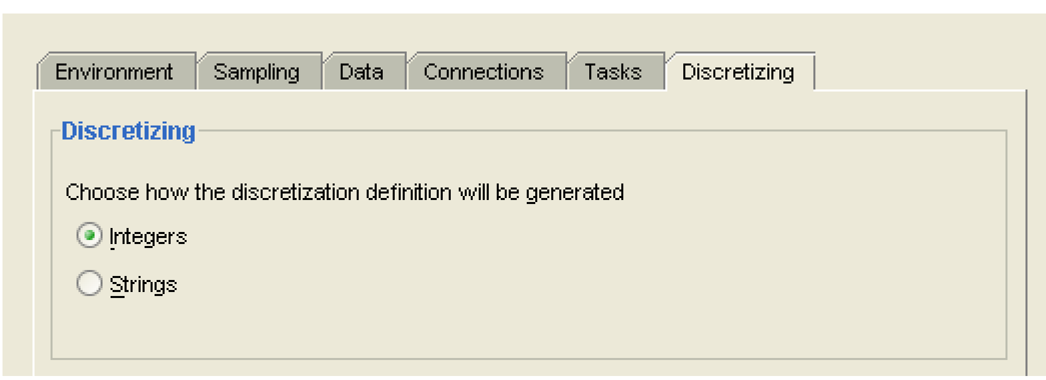
Your database connections are displayed by clicking the Connections tab. You can modify, delete, or create a connection on this page.



The window in the lower left of the GUI displays tasks that are running or are recently completed. Tasks are shown for 60 minutes after completion by default. You can change the default by entering a value on the Tasks page.



When bins are defined for an attribute, the default (internal) method of naming the bins is with integer values. However, when bins are displayed by name, the default value is overridden if necessary for clarity. For example, if AGE is binned and the bins are displayed, you will see ranges such as 16 – 21, 21 – 25 rather than Bin 2, Bin 3.



Appendix C – Predictive Analytics

Oracle Data Mining provides fully automated versions of Attribute Importance (Explain) and Classification/Regression (Predict).

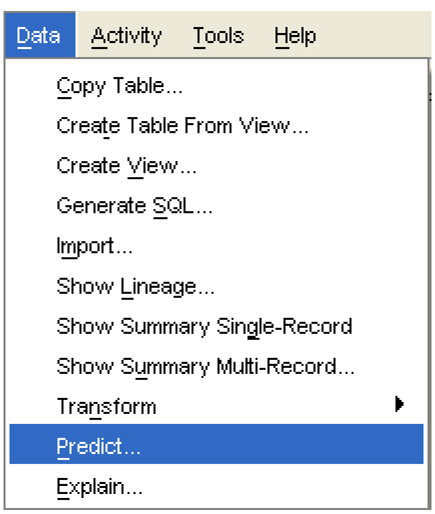
The prerequisites for using Predict and Explain are that the data has been gathered into one table or view, and any desired transformation (for example Recode) that is not required by a particular algorithm has been performed beforehand.

Default or optimized values are used for all parameter settings, required transformations (such as Normalization) are executed automatically, and an appropriate algorithm is applied. The results are in a table named by the user.

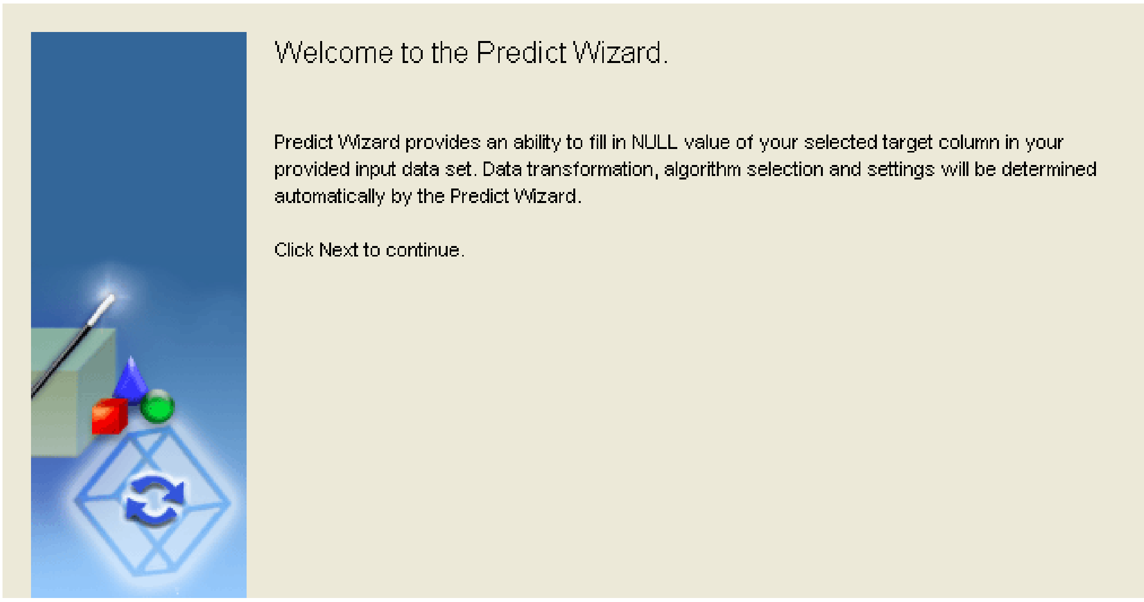
Predict

The user selects the source data and the target attribute. All rows having non-null target values are used as input to the model build process, and the model is applied to all rows. The wizard applies heuristics to determine whether the problem type is Classification or Regression.

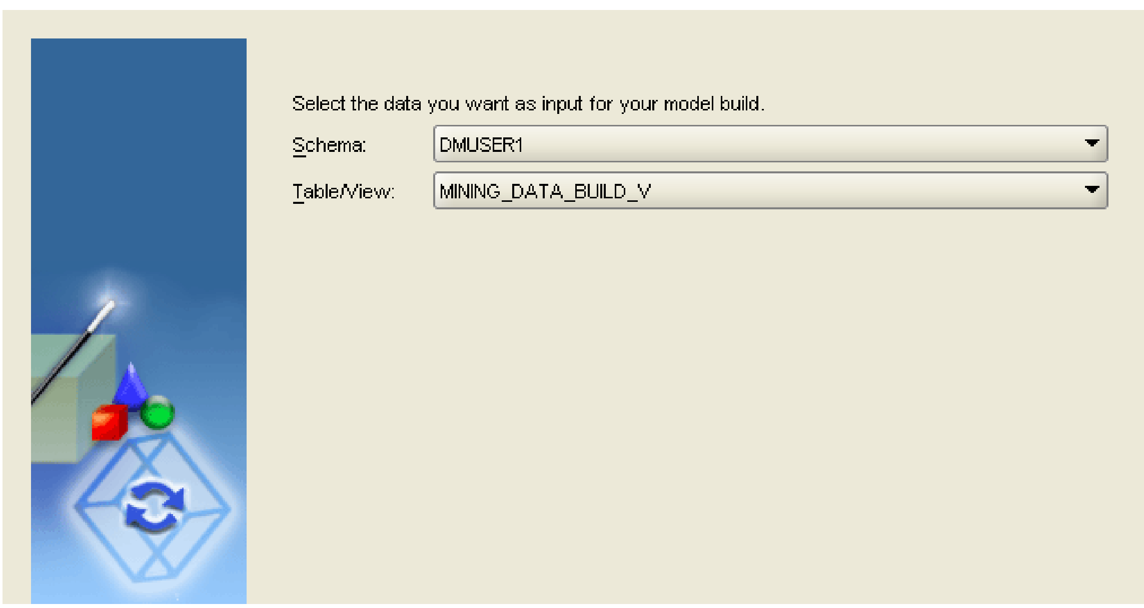
Launch the Predict wizard by selecting Predict on the Data pull-down menu.



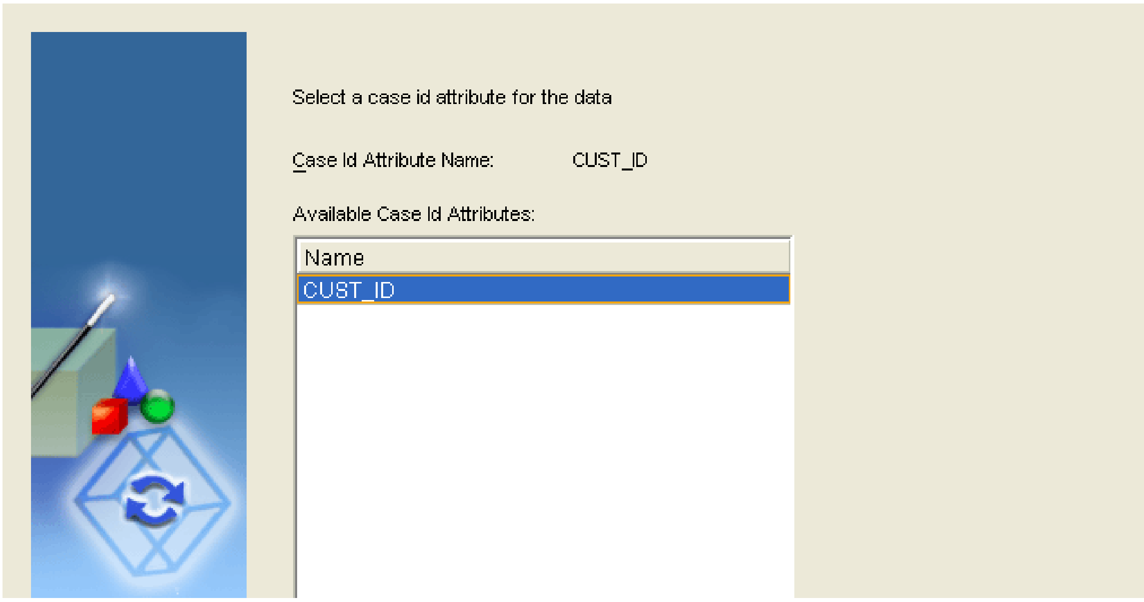
Click Next on the Welcome Screen to continue.



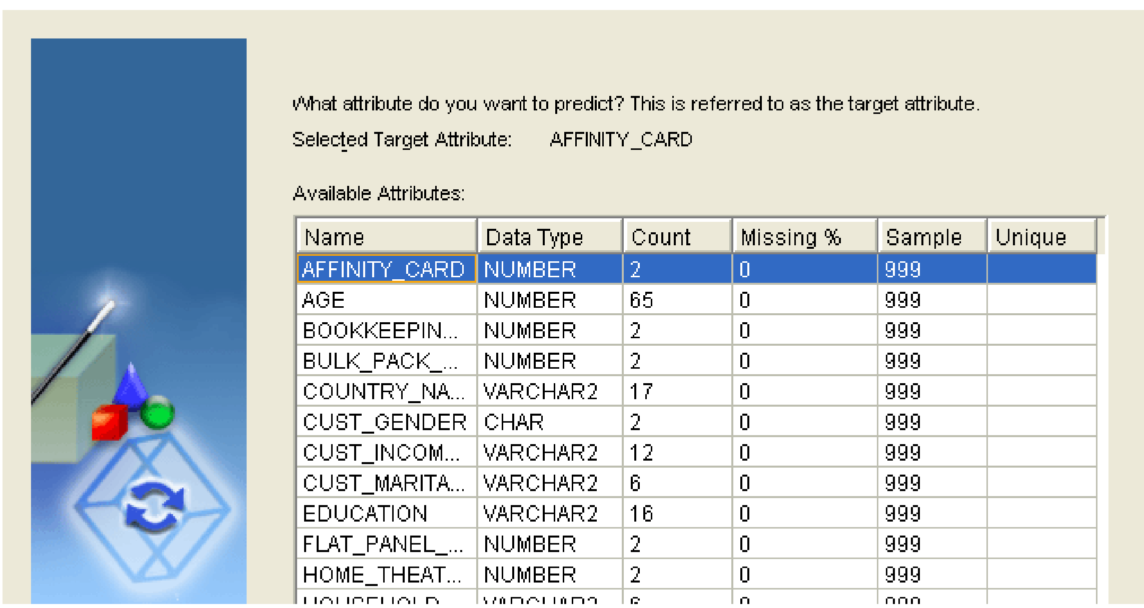
Select the schema and the table/view to analyze. Normally, the table has a target column that is only partially populated; to illustrate the method, select the view MINING_DATA_BUILD_V (which has no NULLS in the target column). Click Next.



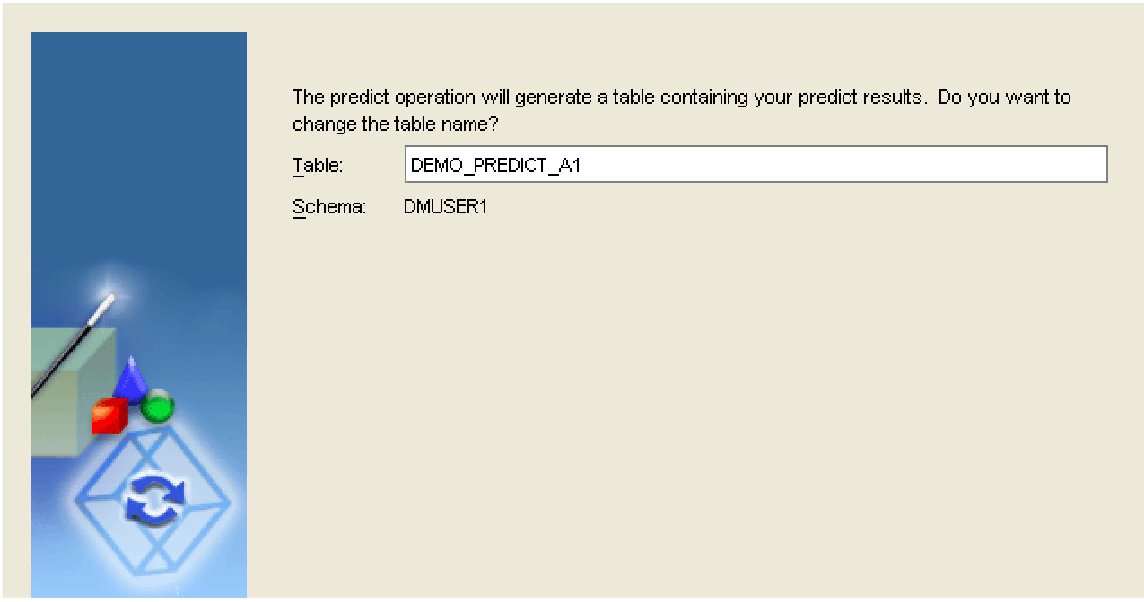
The wizard creates a list of likely row identifiers. Select the Case ID from the list and click Next.



In this data, the column AFFINITY_CARD represents the customer value – 0 for low revenue and 1 for high revenue – you want to predict the value to replace any NULL in this column. Highlight AFFINITY_CARD to specify the target attribute and click Next.

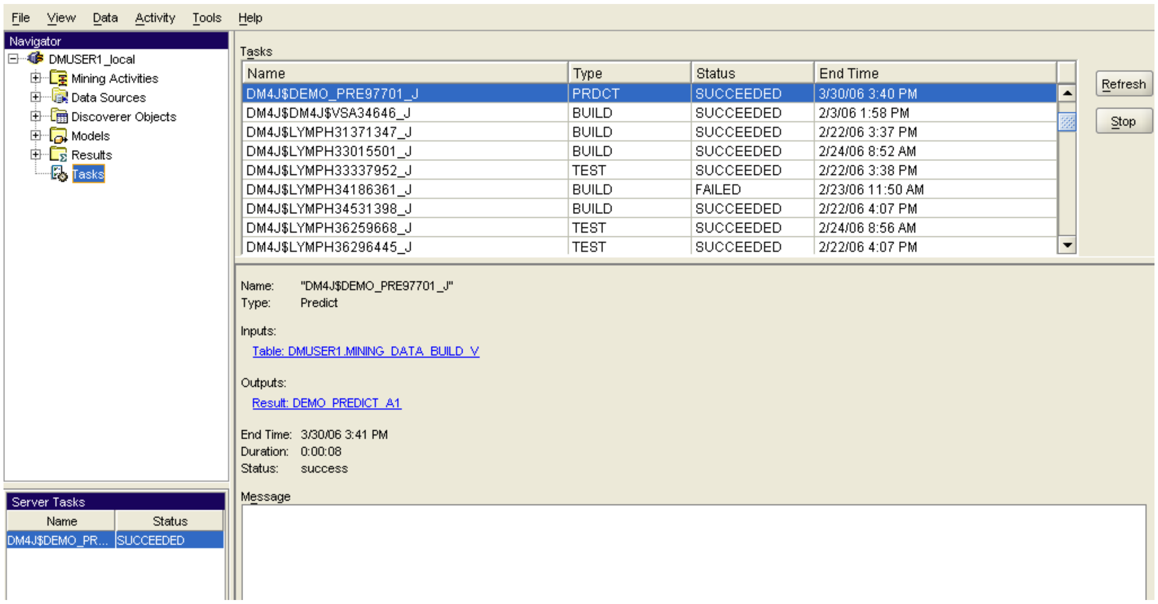


Enter a name for the table that will contain the predictions and click Next.

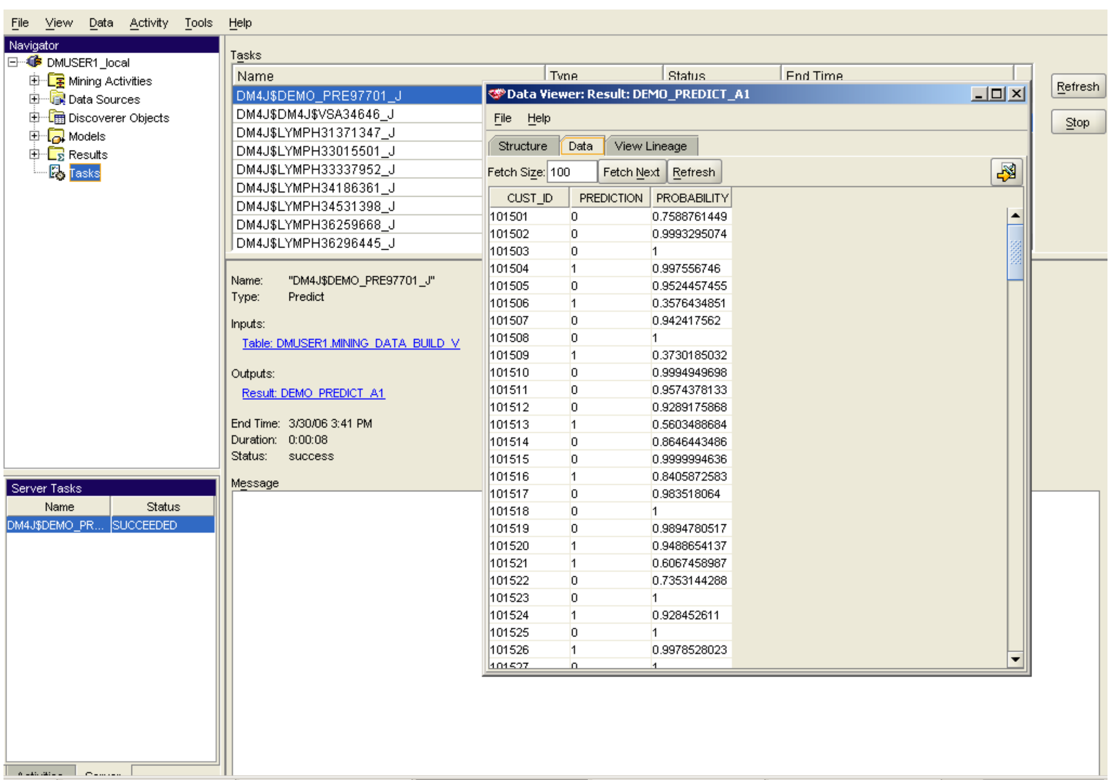


Click Finish on the final page of the wizard.

When the execution completes, click on the task name in the Server Tasks frame to display details of the task.



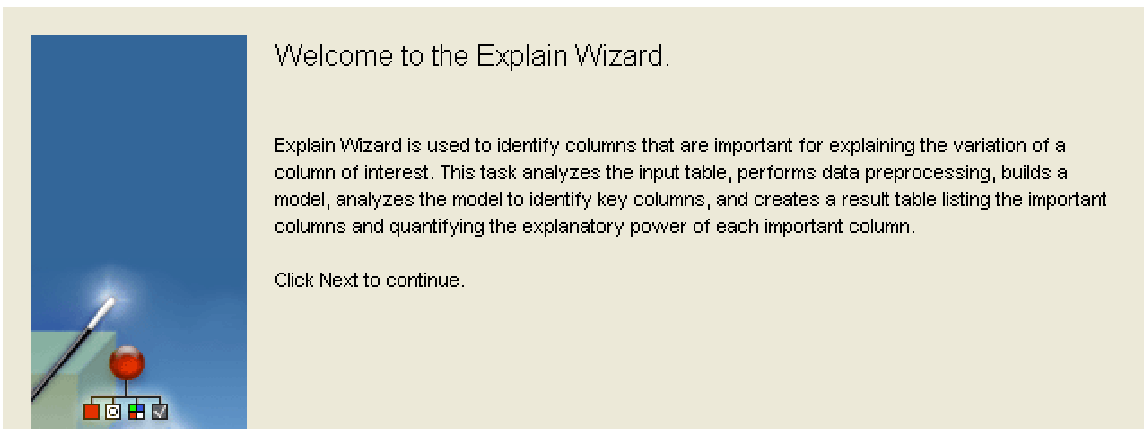
Click the link to the Output Result to display a sample of the contents from the result table.



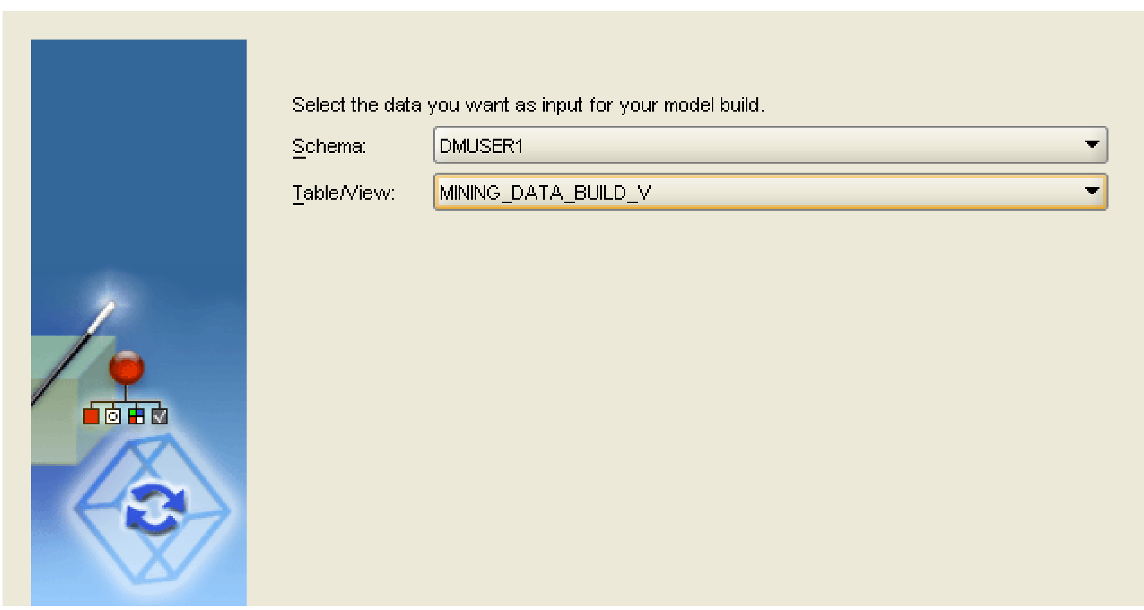
Explain

Explain performs an Attribute Importance as discussed in Chapter 4, except that all user input other than data and target identification is hidden and automated. The end result is a list of attributes ranked by importance in predicting the target value, with attributes having no importance or negative importance assigned the value 0.

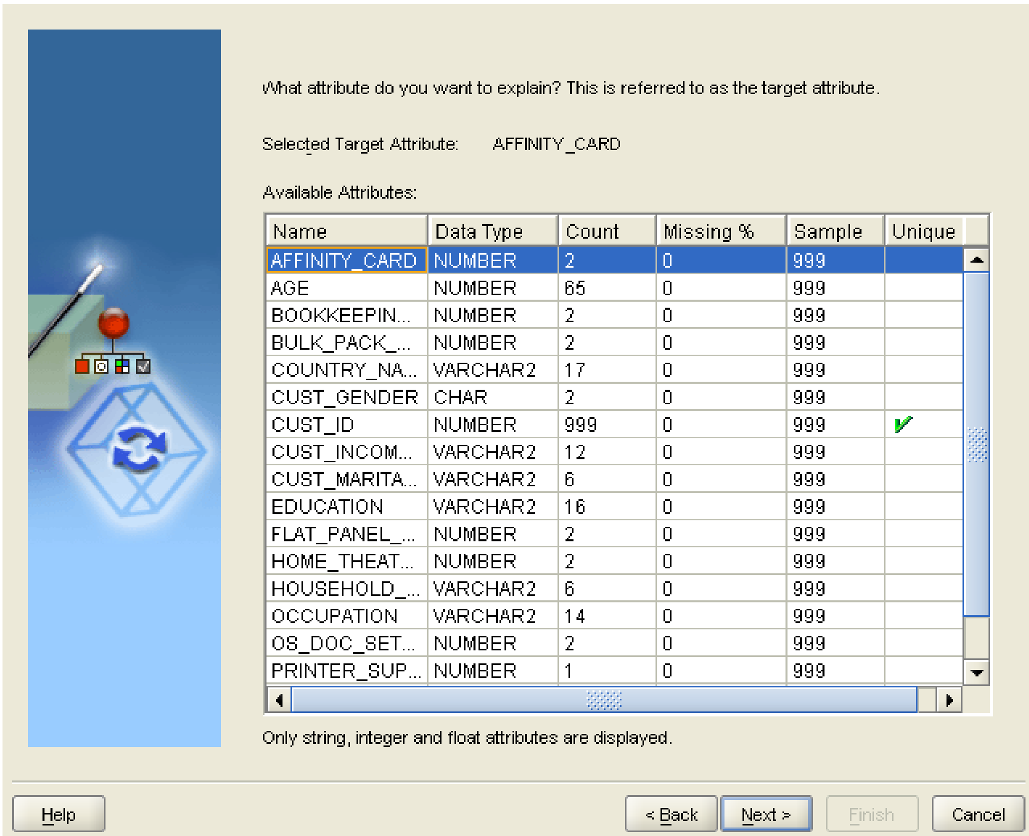
Select Explain from the Data pull-down menu and click Next on the Welcome page to continue.



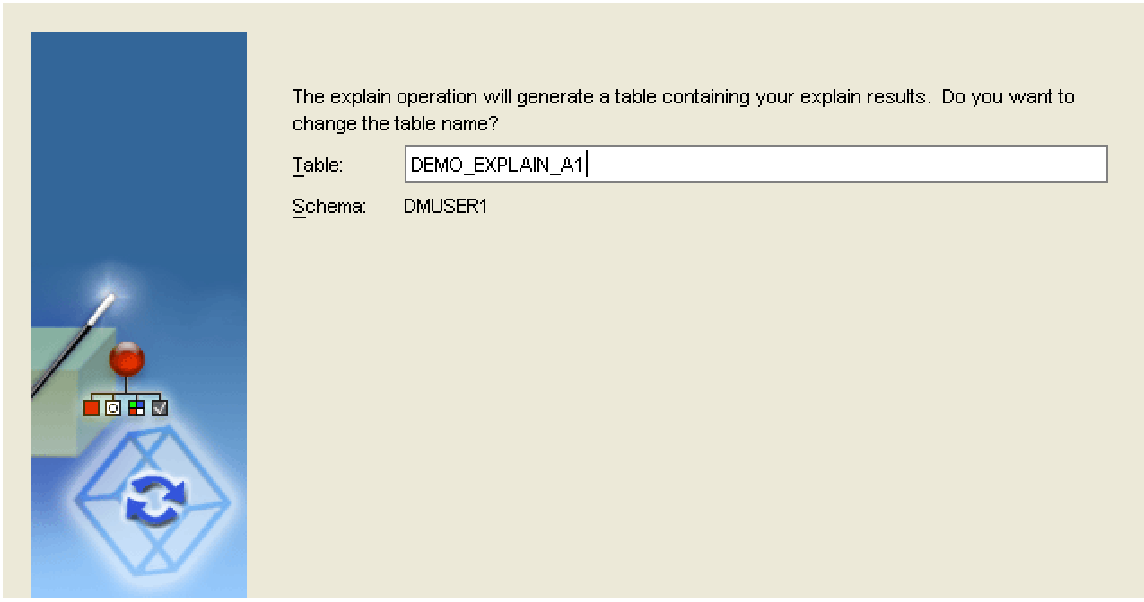
Select the schema and table or view, and click Next.



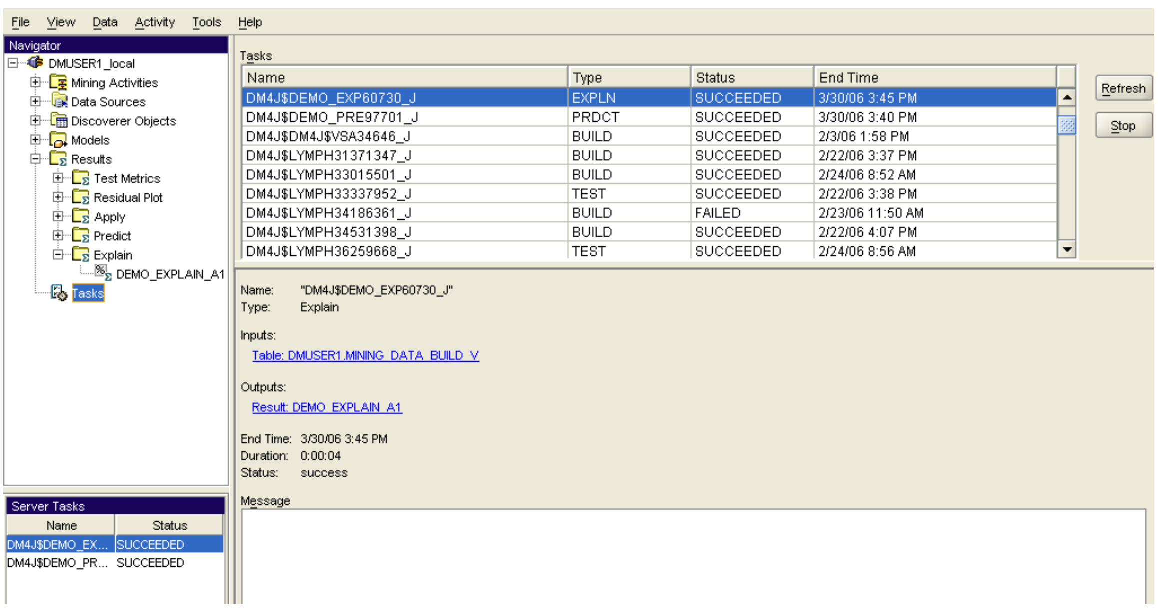
The target to be predicted in MINING_DATA_BUILD_V is the attribute indicating high or low value customers, AFFINITY_CARD. Highlight AFFINITY_CARD and click Next.



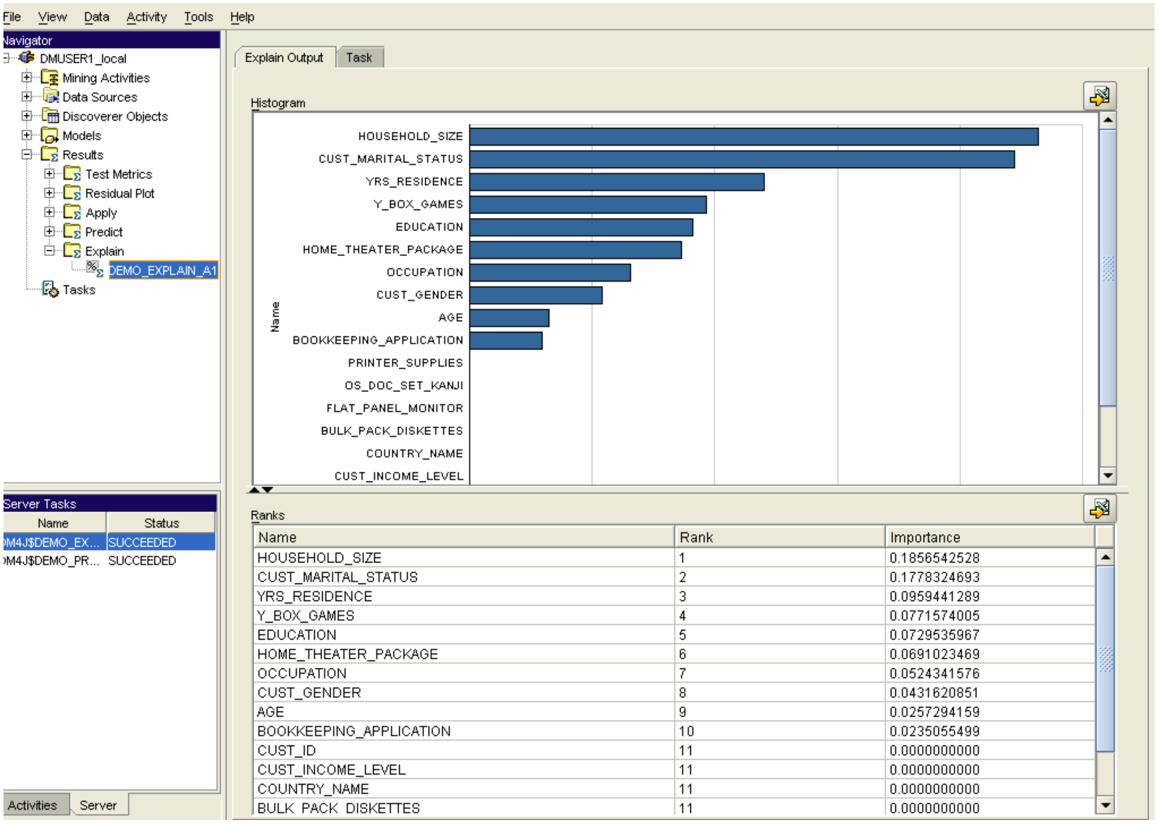
Enter a name for the table that will contain the results, and click Next.



Click Finish on the final wizard page, and when the task completes, click the task name to show the details.



Click the link to the Output Result to display the results, in both graphical and tabular form.


File View Data Activity Tools Help
Navigator
DMUSER1_local
Mining Activities
Data Sources
Discoverer Objects
Models
Results
Test Metrics
Residual Plot
Apply
Predict
Explain
DEMO_EXPLAIN_A1
Tasks
Server Tasks
Name Status
DM4J$DEMO_EX... SUCCEEDED
DM4J$DEMO_PR... SUCCEEDED
Activities Server
Explain Output Task
Histogram
Name
HOUSEHOLD_SIZE
CUST_MARITAL_STATUS
YRS_RESIDENCE
Y_BOX_GAMES
EDUCATION
HOME_THEATER_PACKAGE
OCCUPATION
CUST_GENDER
AGE
BOOKKEEPING_APPLICATION
PRINTER_SUPPLIES
OS_DOC_SET_KANJI
FLAT_PANEL_MONITOR
BULK_PACK_DISKETTES
COUNTRY_NAME
CUST_INCOME_LEVEL
Ranks
Name Rank Importance
HOUSEHOLD_SIZE 1 0.1856542528
CUST_MARITAL_STATUS 2 0.1778324693
YRS_RESIDENCE 3 0.0959441289
Y_BOX_GAMES 4 0.0771574005
EDUCATION 5 0.0729535967
HOME_THEATER_PACKAGE 6 0.0691023469
OCCUPATION 7 0.0524341576
CUST_GENDER 8 0.0431620851
AGE 9 0.0257294159
BOOKKEEPING_APPLICATION 10 0.0235055499
CUST_ID 11 0.0000000000
CUST_INCOME_LEVEL 11 0.0000000000
COUNTRY_NAME 11 0.0000000000
BULK_PACK_DISKETTES 11 0.0000000000

Appendix D — Oracle Data Miner 11.1

Oracle Data Miner 11.1 is the graphical user interface for Oracle Data Mining 11*g*, Release 1 (11.1). Oracle Data Miner 11.1 requires a connection to an Oracle 11*g* Release 1 database; it does not work with any other Oracle databases.

Oracle Data Mining 11.1 New and Changed Features

Oracle Data Mining 11.1 includes Generalized Linear Models (GLM), a new algorithm for classification and regression. Oracle Data Miner 11.1 supports using GLM for classification (logistic regression) and for regression (linear regression).

Note: This tutorial does not describe GLM.

Oracle Data Mining 11.1 deprecates the Adaptive Bayes Network (ABN) algorithm. Oracle Data Miner 11.1 does not support using ABN for classification models. Use the Decision Tree algorithm to create a classification model that includes rules.

For information about all new and changed features in Oracle Data Mining 11.1, see Oracle Data Mining Concepts.

Oracle Data Mining 11.1 Installation

See Oracle Data Mining Administrator's Guide for information about how to install Oracle Data Mining, how to create a user, and how to connect to a database for data mining.

Installation of Oracle Data Miner is described in readme.html, which is part of the Oracle Data Miner download.

Information About Oracle Data Mining 11.1

The Oracle Data Mining Documentation is in the Oracle Database Documentation Library for Oracle 11*g* Release 1. To find data mining documentation, view or download the library, and then click the Data Warehousing and Business Intelligence link.

For information about Oracle Data Mining, go to Oracle Data Mining on Oracle Technology Network. From this page you can find information about Oracle Data Mining, documentation, downloads, forums, blogs, and other useful information.