| | |
|---|---|
| Course Code: | COMP 309 |
| Course Name: | Data Warehousing and Mining |
| Section Number: | 005 |
| Assignment 4: | Developing data mining models |
| Written by: | Kevin Ma |
| Student Number: | 300867968 |
| Date: | Thursday, December 13, 2018 |

## OBJECTIVES:

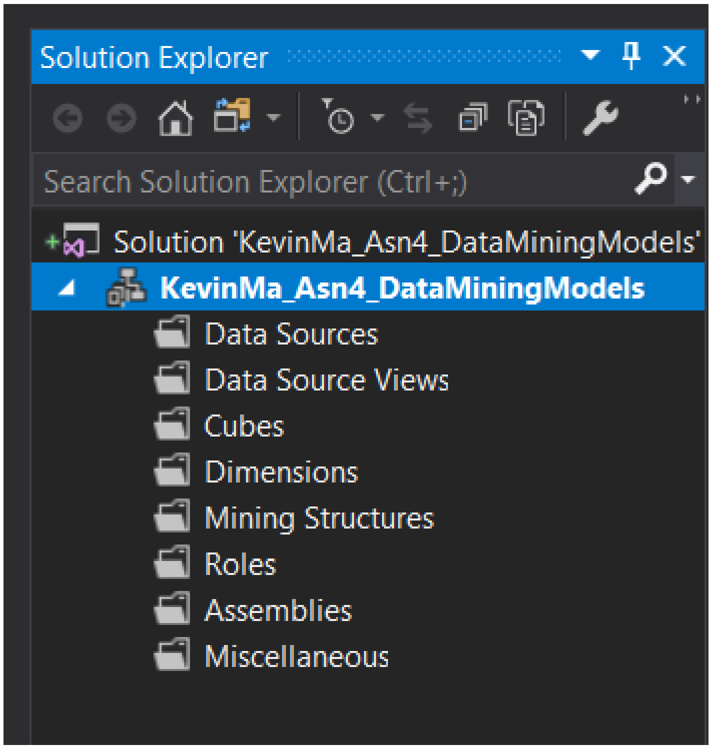
So far in this course, all of the previous projects we have worked on have been dealing with working with past and current data. We have used Data Warehouses, SSAS, and SSIS to take retrospective actions on data and events which have already occurred. This has yielded benefits in making data more easily accessible in homogenous formats in central repositories. In addition, these retrospective actions have allowed businesses to identify patterns and analyze them to identify trends to know how to deal with such situations if they should arise again. However, we are still missing out on potential opportunities if we simply act retrospectively. If businesses were to be more proactive in working with data, executives could make better business decisions through the insight they can gain through predictions made by data mining.

In this assignment, we will go through the process of selecting attributes from the data in the AdventureWorks database to develop 3 data mining models. The three models we will be exploring will be the Microsoft Decision Tree, Microsoft Naïve Bayes and the Microsoft Clustering data mining models. Using the models we created, we will explore the properties of the different data mining models, and learn how to measure the accuracy of their predictions and identify the most efficient data mining model to apply to our current data set.

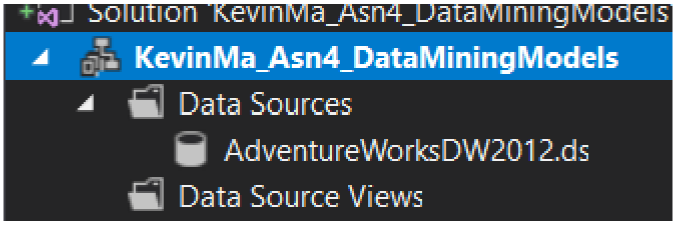
## TIME FOR ACTION – CREATING A DATA MINING SOLUTION WITH THE MICROSOFT DECISION TREE ALGORITHM

In this example, we will create our first data mining solution with the Microsoft decision tree algorithm. The **`AdventureWorksDW`** database contains information about bike buyers. We want to use that information as input variables to the decision tree algorithm and find out what drives someone to buy a bike.

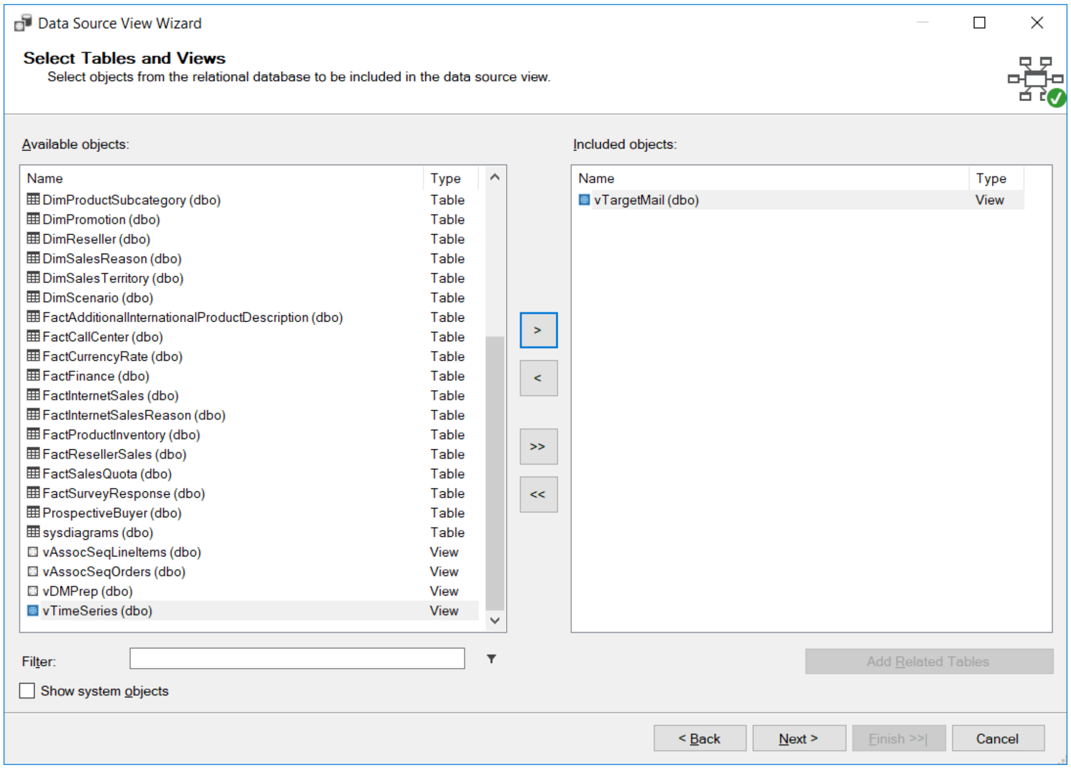
1. Open SSDT and create an **Analysis Service Multidimensional and Data Mining** project. Name the project **`Chapter 07 Data Mining Descriptive Models`**.



2. Right-click on the **`data source`** folder and create a new data source for **`AdventureWorksDW2012`**. Choose the impersonation information with the account that has enough privilege, then accept the default name as **Adventure Works DW2012** and complete the wizard.


Solution 'KevinMa_Asn4_DataMiningModels
KevinMa_Asn4_DataMiningModels
Data Sources
AdventureWorksDW2012.ds
Data Source Views

**3.** Create a data source view with the Adventure Works DW2012 data source. In the **Select Table and Views** menu, select only **vTargetMail** and add it to the included object's list in the box to the right. Click on **Continue** in the wizard and accept the default name for the data source view.



Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

| Name | Type |
|---|---|
| DimProductSubcategory (dbo) | Table |
| DimPromotion (dbo) | Table |
| DimReseller (dbo) | Table |
| DimSalesReason (dbo) | Table |
| DimSalesTerritory (dbo) | Table |
| DimScenario (dbo) | Table |
| FactAdditionalInternationalProductDescription (dbo) | Table |
| FactCallCenter (dbo) | Table |
| FactCurrencyRate (dbo) | Table |
| FactFinance (dbo) | Table |
| FactInternetSales (dbo) | Table |
| FactInternetSalesReason (dbo) | Table |
| FactProductInventory (dbo) | Table |
| FactResellerSales (dbo) | Table |
| FactSalesQuota (dbo) | Table |
| FactSurveyResponse (dbo) | Table |
| ProspectiveBuyer (dbo) | Table |
| sysdiagrams (dbo) | Table |
| vAssocSeqLineItems (dbo) | View |
| vAssocSeqOrders (dbo) | View |
| vDMPrep (dbo) | View |
| vTimeSeries (dbo) | View |

Included objects:

| Name | Type |
|---|---|
| vTargetMail (dbo) | View |

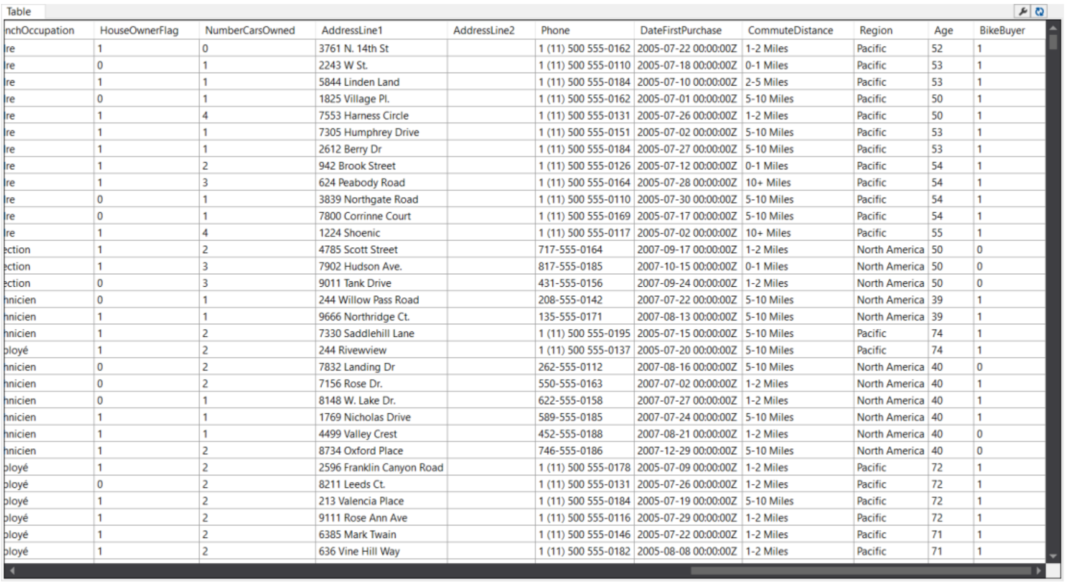
Filter:

□ Show system objects

**4.** In DSV designer, right-click on the `vTargetMail` table and click on **Explore Data**.

You will see the customers' information plus an additional field for the bike buyer.

The `BikeBuyer` field is populated with the existing sales record in the `sales` table.

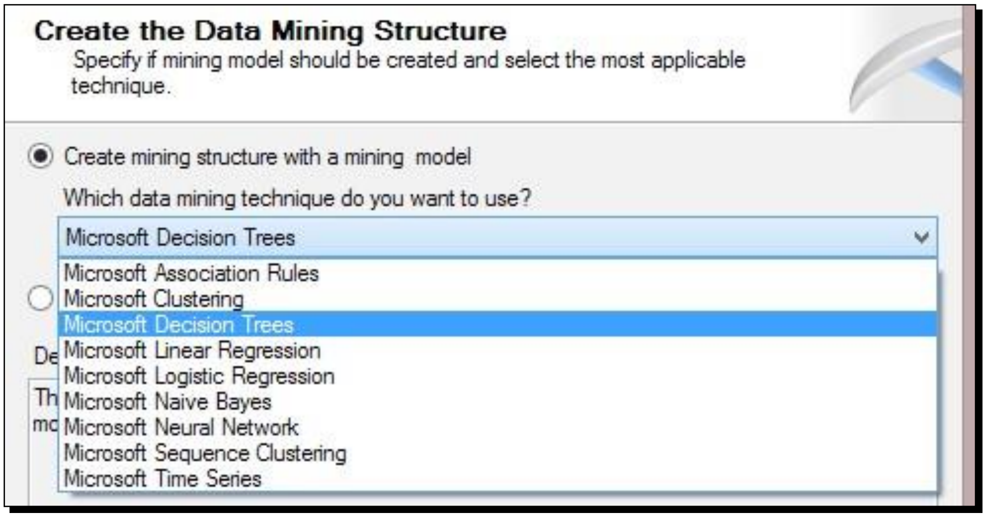


| nchOccupation | HouseOwnerFlag | NumberCarsOwned | AddressLine1 | AddressLine2 | Phone | DateFirstPurchase | CommuteDistance | Region | Age | BikeBuyer |
|---|---|---|---|---|---|---|---|---|---|---|
| re | 1 | 0 | 3761 N. 14th St |  | 1 (11) 500 555-0162 | 2005-07-22 00:00:00Z | 1-2 Miles | Pacific | 52 | 1 |
| re | 0 | 1 | 2243 W St. |  | 1 (11) 500 555-0110 | 2005-07-18 00:00:00Z | 0-1 Miles | Pacific | 53 | 1 |
| re | 1 | 1 | 5844 Linden Land |  | 1 (11) 500 555-0184 | 2005-07-10 00:00:00Z | 2-5 Miles | Pacific | 53 | 1 |
| re | 0 | 1 | 1825 Village Pl. |  | 1 (11) 500 555-0162 | 2005-07-01 00:00:00Z | 5-10 Miles | Pacific | 50 | 1 |
| re | 1 | 4 | 7553 Harness Circle |  | 1 (11) 500 555-0131 | 2005-07-26 00:00:00Z | 1-2 Miles | Pacific | 50 | 1 |
| re | 1 | 1 | 7305 Humphrey Drive |  | 1 (11) 500 555-0151 | 2005-07-02 00:00:00Z | 5-10 Miles | Pacific | 53 | 1 |
| re | 1 | 1 | 2612 Berry Dr |  | 1 (11) 500 555-0184 | 2005-07-27 00:00:00Z | 5-10 Miles | Pacific | 53 | 1 |
| re | 1 | 2 | 942 Brook Street |  | 1 (11) 500 555-0126 | 2005-07-12 00:00:00Z | 0-1 Miles | Pacific | 54 | 1 |
| re | 1 | 3 | 624 Peabody Road |  | 1 (11) 500 555-0164 | 2005-07-28 00:00:00Z | 10+ Miles | Pacific | 54 | 1 |
| re | 0 | 1 | 3839 Northgate Road |  | 1 (11) 500 555-0110 | 2005-07-30 00:00:00Z | 5-10 Miles | Pacific | 54 | 1 |
| re | 0 | 1 | 7800 Corrinne Court |  | 1 (11) 500 555-0169 | 2005-07-17 00:00:00Z | 5-10 Miles | Pacific | 54 | 1 |
| re | 1 | 4 | 1224 Shoenic |  | 1 (11) 500 555-0117 | 2005-07-02 00:00:00Z | 10+ Miles | Pacific | 55 | 1 |
| ection | 1 | 2 | 4785 Scott Street |  | 717-555-0164 | 2007-09-17 00:00:00Z | 1-2 Miles | North America | 50 | 0 |
| ection | 1 | 3 | 7902 Hudson Ave. |  | 817-555-0185 | 2007-10-15 00:00:00Z | 0-1 Miles | North America | 50 | 0 |
| ection | 0 | 3 | 9011 Tank Drive |  | 431-555-0156 | 2007-09-24 00:00:00Z | 1-2 Miles | North America | 50 | 0 |
| hnicien | 0 | 1 | 244 Willow Pass Road |  | 208-555-0142 | 2007-07-22 00:00:00Z | 5-10 Miles | North America | 39 | 1 |
| hnicien | 1 | 1 | 9666 Northridge Ct. |  | 135-555-0171 | 2007-08-13 00:00:00Z | 5-10 Miles | North America | 39 | 1 |
| hnicien | 1 | 2 | 7330 Saddlehill Lane |  | 1 (11) 500 555-0195 | 2005-07-15 00:00:00Z | 5-10 Miles | Pacific | 74 | 1 |
| ployé | 1 | 2 | 244 Rivewview |  | 1 (11) 500 555-0137 | 2005-07-20 00:00:00Z | 5-10 Miles | Pacific | 74 | 1 |
| hnicien | 0 | 2 | 7832 Landing Dr |  | 262-555-0112 | 2007-08-16 00:00:00Z | 5-10 Miles | North America | 40 | 0 |
| hnicien | 0 | 2 | 7156 Rose Dr. |  | 550-555-0163 | 2007-07-02 00:00:00Z | 1-2 Miles | North America | 40 | 1 |
| hnicien | 0 | 1 | 8148 W. Lake Dr. |  | 622-555-0158 | 2007-07-27 00:00:00Z | 1-2 Miles | North America | 40 | 1 |
| hnicien | 1 | 1 | 1769 Nicholas Drive |  | 589-555-0185 | 2007-07-24 00:00:00Z | 5-10 Miles | North America | 40 | 1 |
| hnicien | 1 | 1 | 4499 Valley Crest |  | 452-555-0188 | 2007-08-21 00:00:00Z | 1-2 Miles | North America | 40 | 0 |
| hnicien | 1 | 2 | 8734 Oxford Place |  | 746-555-0186 | 2007-12-29 00:00:00Z | 5-10 Miles | North America | 40 | 0 |
| ployé | 1 | 2 | 2596 Franklin Canyon Road |  | 1 (11) 500 555-0178 | 2005-07-09 00:00:00Z | 1-2 Miles | Pacific | 72 | 1 |
| ployé | 0 | 2 | 8211 Leeds Ct. |  | 1 (11) 500 555-0131 | 2005-07-26 00:00:00Z | 1-2 Miles | Pacific | 72 | 1 |
| ployé | 1 | 2 | 213 Valencia Place |  | 1 (11) 500 555-0184 | 2005-07-19 00:00:00Z | 5-10 Miles | Pacific | 72 | 1 |
| ployé | 1 | 2 | 9111 Rose Ann Ave |  | 1 (11) 500 555-0116 | 2005-07-29 00:00:00Z | 1-2 Miles | Pacific | 72 | 1 |
| ployé | 1 | 2 | 6385 Mark Twain |  | 1 (11) 500 555-0146 | 2005-07-22 00:00:00Z | 1-2 Miles | Pacific | 71 | 1 |
| ployé | 1 | 2 | 636 Vine Hill Way |  | 1 (11) 500 555-0182 | 2005-08-08 00:00:00Z | 1-2 Miles | Pacific | 71 | 1 |

5. Right-click on the **Mining Structures** folder and create a new mining structure.

6. In the **Data Mining** wizard, the first step is to select the definition method.  Choose from the existing relational databases or data warehouse.

7. In the **Create the Data Mining Structure** window, click on the **Create mining structure with a mining model** radio button and choose **Microsoft** from the list of algorithms in the **Which data mining technique do you want to use?** dropdown and continue, as shown in the following screenshot:



Create the Data Mining Structure

Specify if mining model should be created and select the most applicable technique.

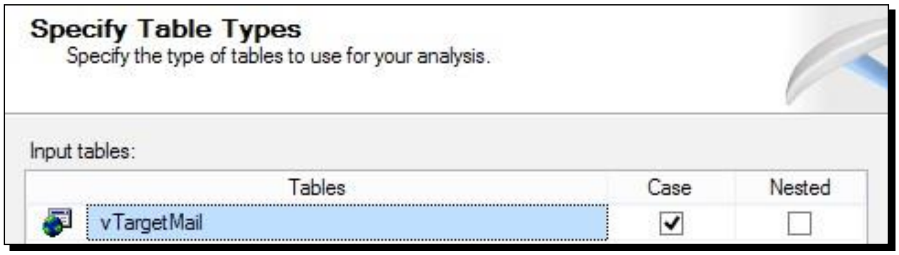
● Create mining structure with a mining model

Which data mining technique do you want to use?

Microsoft Decision Trees

8. In the **Select Data Source View** menu, choose the **Adventure Works DW2012** data source view and go to the next step.

9. In the **Specify Table Types** window, verify that the `vTargetMail` table is selected as the `Case` table, as shown in the following screenshot:



Specify Table Types

Specify the type of tables to use for your analysis.

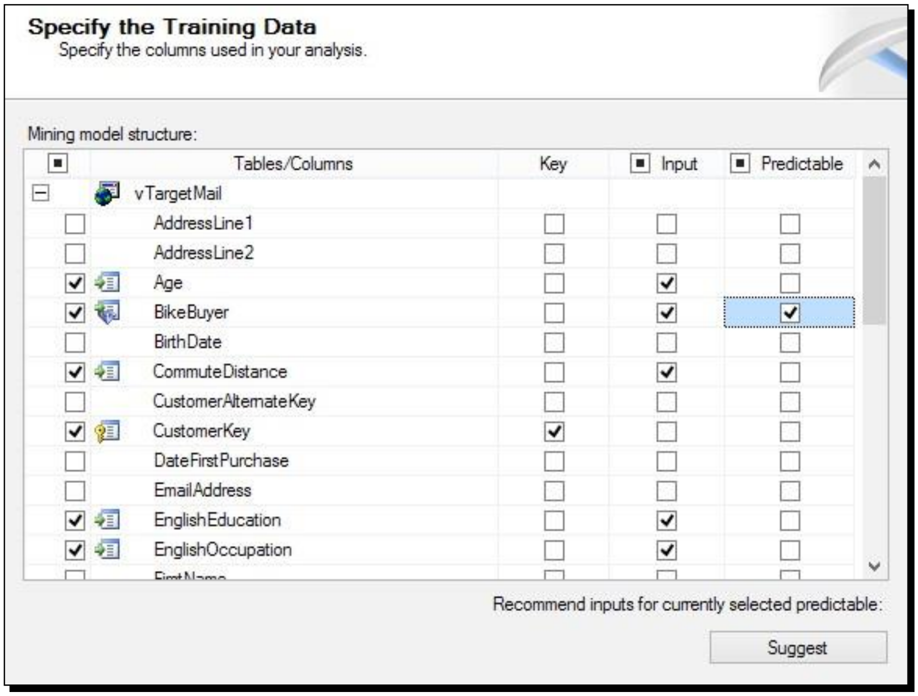
Input tables:

| Tables | Case | Nested |
|---|---|---|
| vTargetMail | ☑ | □ |

10. In the **Specify the Training Data** window, choose `CustomerKey` as the `Key` column. Select these columns as input: **Age**, **BikeBuyer**, **CommuteDistance**,

**EnglishEducation**, **EnglishOccupation**, **Gender**, **HouseOwnerFlag**, **MaritalStatus**,

**NumberCarsOwned**, **NumberChildrenAtHome**, **Region**, **TotalChildren**, and **YearlyIncome**. Finally, set **BikeBuyer** as **Predictable**. The following screenshot shows this configuration (note that not all columns are shown in this screenshot):



**Specify the Training Data**

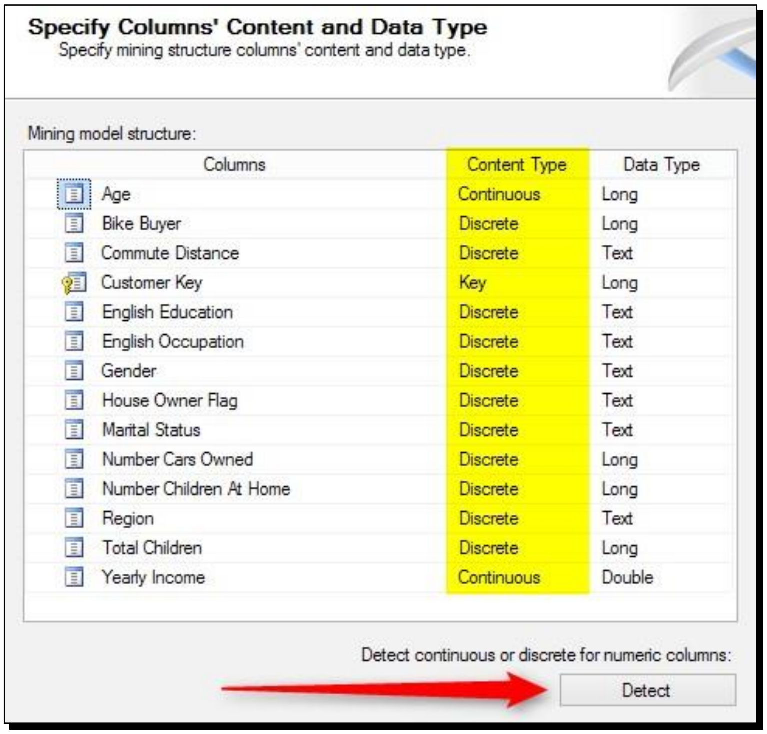
Specify the columns used in your analysis.

Mining model structure:

| ■ | Tables/Columns | Key | ■ Input | ■ Predictable |
|---|---|---|---|---|
| ⊟ | vTargetMail | | | |
| □ | AddressLine1 | □ | □ | □ |
| □ | AddressLine2 | □ | □ | □ |
| ☑ | Age | □ | ☑ | □ |
| ☑ | BikeBuyer | □ | ☑ | ☑ |
| □ | BirthDate | □ | □ | □ |
| ☑ | CommuteDistance | □ | ☑ | □ |
| □ | CustomerAlternateKey | □ | □ | □ |
| ☑ | CustomerKey | ☑ | □ | □ |
| □ | DateFirstPurchase | □ | □ | □ |
| □ | EmailAddress | □ | □ | □ |
| ☑ | EnglishEducation | □ | ☑ | □ |
| ☑ | EnglishOccupation | □ | ☑ | □ |
| □ | FirstName | □ | □ | □ |

Recommend inputs for currently selected predictable:

11. In the **Specify Column's Content and Data Type** window, you will see the **Content Type** and **Data Type** of the column fetched based on the metadata of the underlying data source view. Click on **Detect** to adjust the content type based on existing values for each column. As a result, you will see that some columns, such as `BikeBuyer` and `CommuteDistance`, have the content type `Discrete`. This is shown in the following screenshot:



**Specify Columns' Content and Data Type**

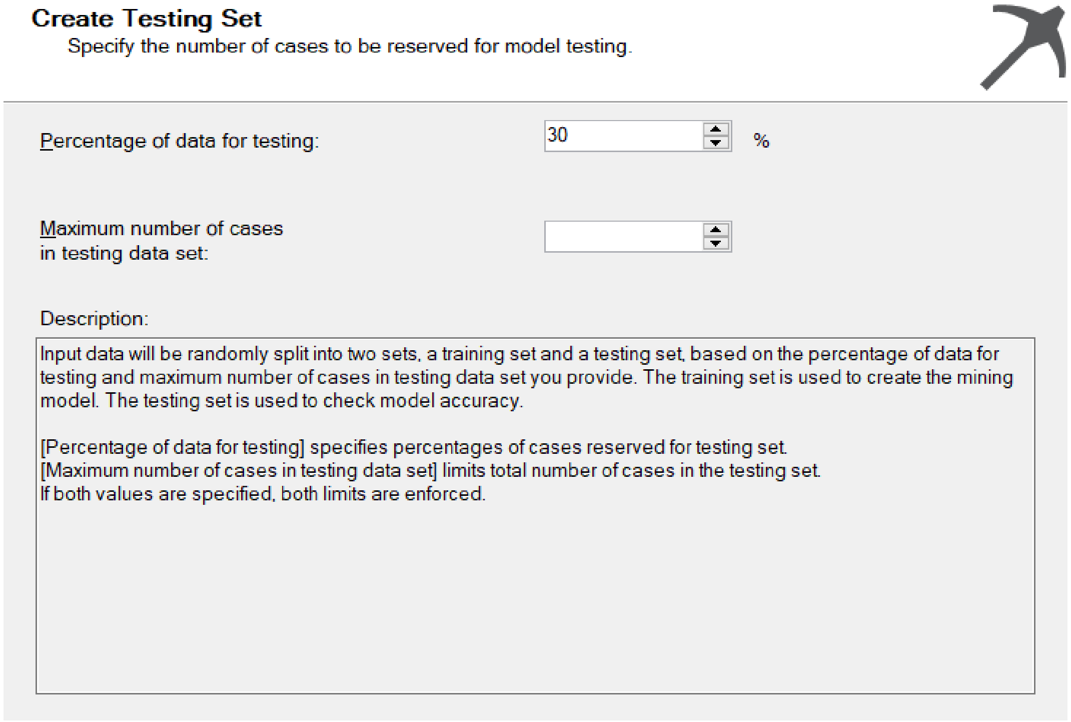
Specify mining structure columns' content and data type.

Mining model structure:

| Columns | Content Type | Data Type |
|---|---|---|
| Age | Continuous | Long |
| Bike Buyer | Discrete | Long |
| Commute Distance | Discrete | Text |
| Customer Key | Key | Long |
| English Education | Discrete | Text |
| English Occupation | Discrete | Text |
| Gender | Discrete | Text |
| House Owner Flag | Discrete | Text |
| Marital Status | Discrete | Text |
| Number Cars Owned | Discrete | Long |
| Number Children At Home | Discrete | Long |
| Region | Discrete | Text |
| Total Children | Discrete | Long |
| Yearly Income | Continuous | Double |

Detect continuous or discrete for numeric columns:

***12.*** In the **Create Testing Set** window, enter `30%` for the percentage of data for testing. Leave the other option as it is and go to the next step.



Create Testing Set

Specify the number of cases to be reserved for model testing.

Percentage of data for testing: 30 %

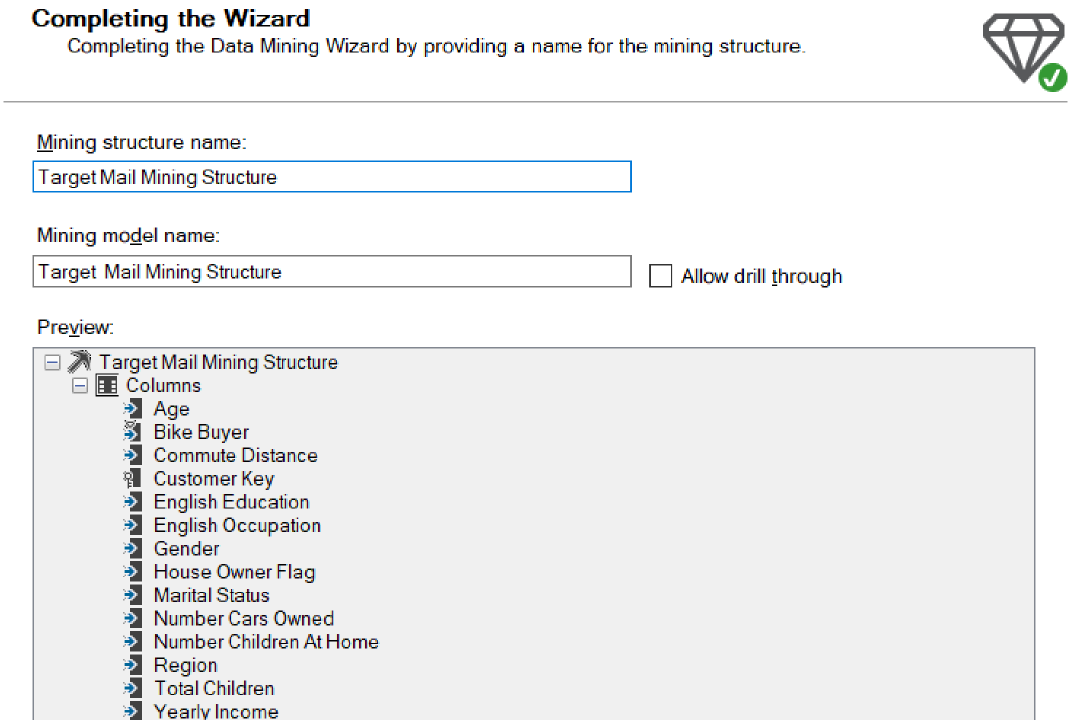
Maximum number of cases in testing data set:

Description:

Input data will be randomly split into two sets, a training set and a testing set, based on the percentage of data for testing and maximum number of cases in testing data set you provide. The training set is used to create the mining model. The testing set is used to check model accuracy.

[Percentage of data for testing] specifies percentages of cases reserved for testing set.
[Maximum number of cases in testing data set] limits total number of cases in the testing set.
If both values are specified, both limits are enforced.

***13.*** In the **Create Testing Set** window, enter the name of **Mining Structure** as `Target Mail Mining Structure` and rename **Mining Model** to `Target  Mail Mining Structure`.



**Completing the Wizard**

Completing the Data Mining Wizard by providing a name for the mining structure.

Mining structure name: Target Mail Mining Structure

Mining model name: Target Mail Mining Structure

□ Allow drill through

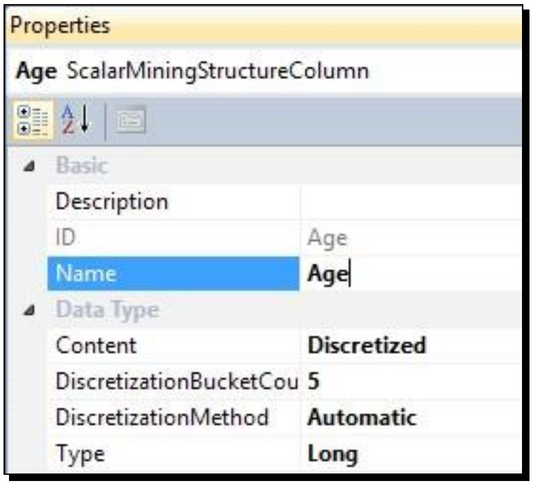
Preview:

- Target Mail Mining Structure
  - Columns
    - Age
    - Bike Buyer
    - Commute Distance
    - Customer Key
    - English Education
    - English Occupation
    - Gender
    - House Owner Flag
    - Marital Status
    - Number Cars Owned
    - Number Children At Home
    - Region
    - Total Children
    - Yearly Income

14. After completing the wizard, you will see the **Mining Structure** designer window, and you will see `vTargetMail` as the only case table with a yellow header in the designer.

> Before analyzing the result of the decision tree algorithm, we need to change the input variable with a continuous content type to discrete. The reason for doing this is explained in the *What just happened?* section with full details.

15. Select the **Age** column in the **Mining Structure** tab and go to the **Properties** window. Change the **Content** type from **Continuous** to **Discretized**, set **DiscretizationBucketCount** as **5**, and leave the other properties as they are, as shown in the following screenshot:

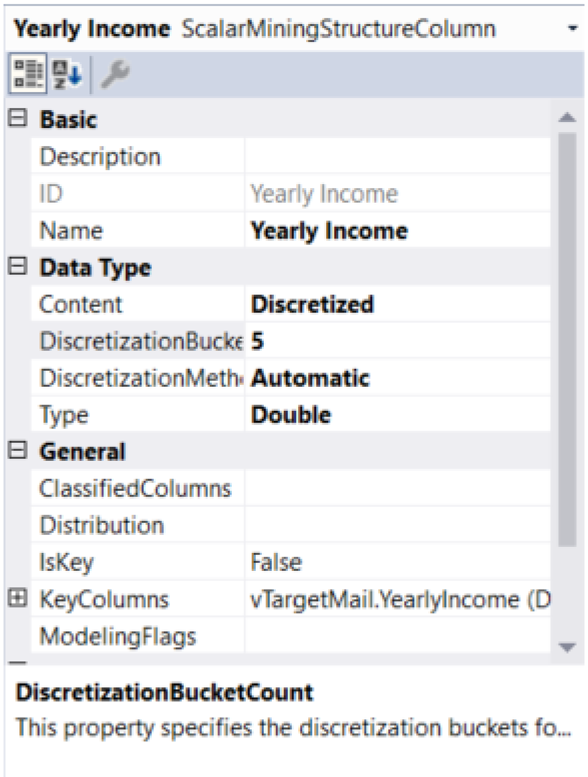


Properties

**Age** ScalarMiningStructureColumn

| | |
|---|---|
| Basic | |
| Description | |
| ID | Age |
| Name | Age |
| Data Type | |
| Content | **Discretized** |
| DiscretizationBucketCou | **5** |
| DiscretizationMethod | **Automatic** |
| Type | **Long** |

**16.** Perform the same discretization configuration for the `YearlyIncome` column.



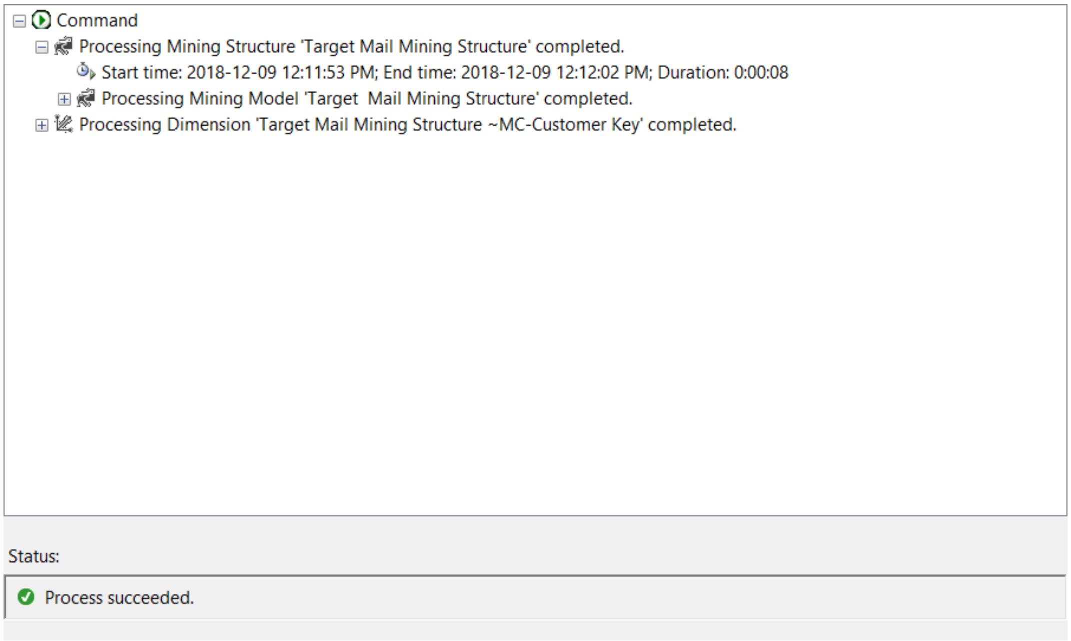
**Yearly Income** ScalarMiningStructureColumn

| | |
|---|---|
| **Basic** | |
| Description | |
| ID | Yearly Income |
| Name | **Yearly Income** |
| **Data Type** | |
| Content | **Discretized** |
| DiscretizationBucke | **5** |
| DiscretizationMeth | **Automatic** |
| Type | **Double** |
| **General** | |
| ClassifiedColumns | |
| Distribution | |
| IsKey | False |
| KeyColumns | vTargetMail.YearlyIncome (D |
| ModelingFlags | |

**DiscretizationBucketCount**

This property specifies the discretization buckets fo...

**17.** In the solution explorer, right-click on **Target Mail Mining Structure** and process it.



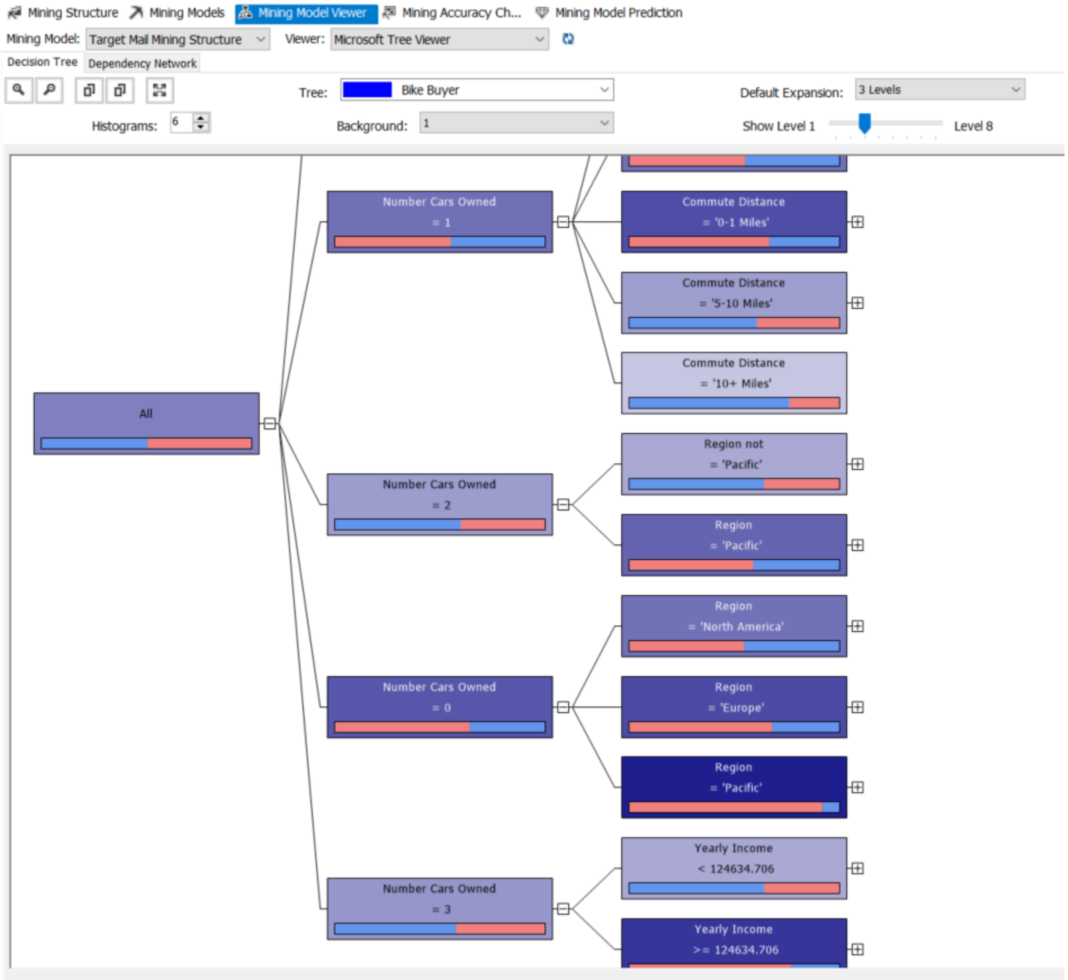
```
⊟ Command
  ⊟ Processing Mining Structure 'Target Mail Mining Structure' completed.
      Start time: 2018-12-09 12:11:53 PM; End time: 2018-12-09 12:12:02 PM; Duration: 0:00:08
    ⊞ Processing Mining Model 'Target  Mail Mining Structure' completed.
  ⊞ Processing Dimension 'Target Mail Mining Structure ~MC-Customer Key' completed.
```

Status:

✓ Process succeeded.

18. Go to the **Mining Model Viewer** tab. You will see the **Microsoft Tree Viewer** as the default viewer, which shows that the decision tree structure is fetched based on the mining model configuration and the underlying database.

19. In the **Decision Tree** tab, set the background color as `1`. This means that the tree node's background color will be set such that the **`BikeBuyer`** variable will be 1. As a result, the node with more possibility of bike buyers in its subset will be shaded with a background color of dark blue. This is shown in the following screenshot:



Mining Model: Target Mail Mining Structure

Viewer: Microsoft Tree Viewer

Tree: Bike Buyer

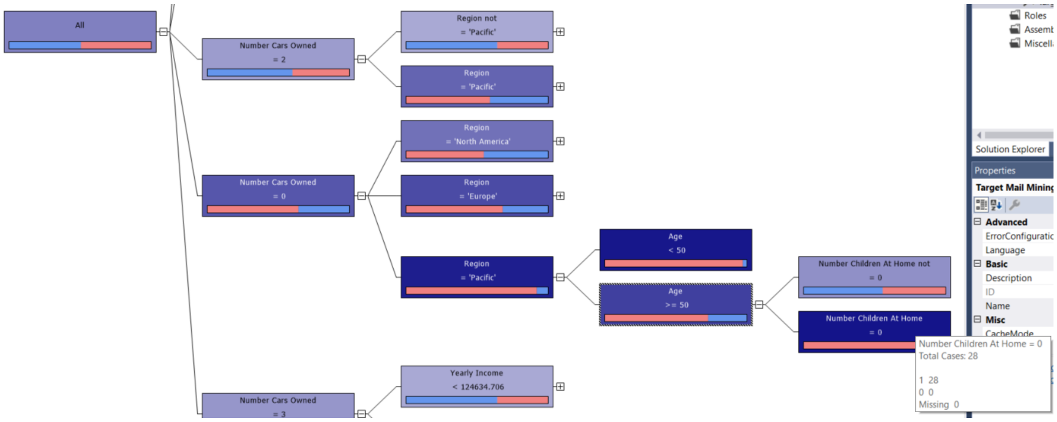
Default Expansion: 3 Levels

Histograms: 6

Background: 1

As you can see in the previous screenshot, you can change the level of a decision tree node to explore the decision path that defines input variables with their values that played the most important role to lead most of the customers to buy that product.

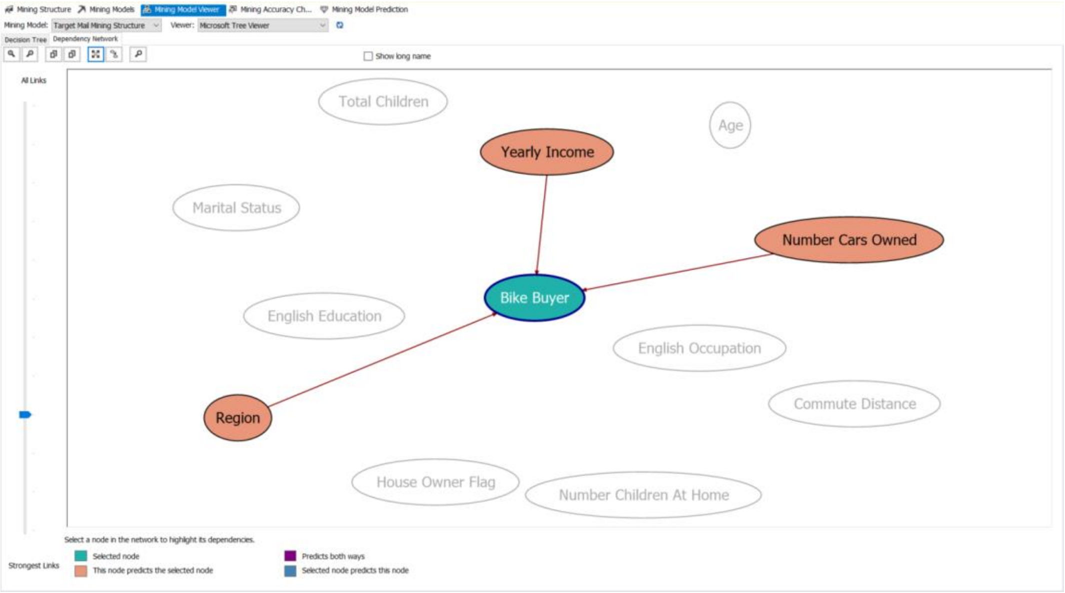
The first decision tree level made in the preceding screenshot is the number of cars owned. As you can see, people with no cars have a greater tendency to buy a bike. Then, if you expand the tree, you will see that among the people who don't own cars, those who are younger than 45 are the subset of customers who bought bikes the most. If you continue the decision path, you will find nodes with very high possibility of bike buyers. While hovering on each node, you can see the number of records in that subset, and whether those subsets are bike buyers or not. The following screenshot shows a sample decision path:



Number Children At Home = 0
Total Cases: 28

1 28
0 0
Missing 0

**20.** Click on the **Dependency Network** tab. You will see a chart of input variables with the `predictable` variable at the middle, which looks like a star schema of variables. **Dependency Network** shows the contribution of each variable in the value of the predictable variable. On the left-hand side, you will see a dependency bar. The highest level in the bar shows all dependency links (which literally means all variables). If you scroll the bar down, you will see only the strongest dependencies. For example, the following screenshot shows that **Number Cars Owned**, **Yearly Income**, and **Region** are the most important directives to buy a bike based on the decision tree model fetched from the existing dataset:



Mining Model: Target Mail Mining Structure
Viewer: Microsoft Tree Viewer

Select a node in the network to highlight its dependencies.

Congratulations! You built your first data mining model in this example. Building data mining models, as you've seen in this example, is easy. You don't need to worry and think about how the data mining algorithm works. You can simply use it with the data mining wizard, and also with some property configurations. Finally, you will see that the results of the mining model are charts; for example, the decision tree viewer and the dependency network viewer that show the impact of input variables and their contribution to the predictable variable.

The following are the steps in the data mining life cycle.

## PROBLEM DEFINITION

The very first step of each data mining solution is the problem definition. The problem in this example was to find prospective bike buyers. In other words, assume this scenario: the AdventureWorks company wants to build a new bike model, and the company would like to know who would buy this product. They want to use data mining algorithms to focus on customers that might want to buy the new bike, not all of them. There are many reasons why a company would want to focus only on high-probability bike buyers; for example, advertising costs will be much lower and effective if it targets only those customers.

## DATA PREPARATION

The data preparation step starts after defining the problem. Data preparation is the process of understanding the existing data and finding variables in the existing data. For example, the database view that was used in this sample (**`vTargetMail`**) contains customer information such as age, income, number of children, and commute distance. On the other hand, existing customers may or may not have bought a bike from this company previously, so the **`BikeBuyer`** field can be gathered based on previous sales data records. The **`vTargetMail`** database contains both information, that is, customer fields and the **`BikeBuyer`** column.

Data preparation can be done not only with SQL Server and database query tools, but also with Excel. Excel is a very good tool to identify columns that play the most important roles  in the result set. For example, the commute distance is an important variable in the decision to buy a bike, but the last name of the customer or the address line is not. In *Chapter 11*, *Power BI*, you will learn how Excel can be used in data analysis.

After data preparation, the next step is to assign input variables from the existing dataset to data mining algorithms and build models. Microsoft's data mining solution is located in the Analysis Services multidimensional project template. This doesn't mean that you need to build a cube to create a mining model. The SSAS project is just a container for the data mining structure. As you've seen in step 1 of the *Time for action – creating a data mining solution with the Microsoft Decision Tree* algorithm section, the project template that is used in this example is the Analysis Service multidimensional and data mining project.

The data mining structure requires a data source and data source views. So, we simply created a data source and a DSV for the **`vTargetMail`** view in steps 2 and 3. Step 4 shows you a view of the existing dataset, which we will use to feed into the mining model.

## THE MINING STRUCTURE

Microsoft uses the mining structure based on the data mining variables and their data structure for the metadata that was built, for example, data types and content types. In steps 5 and 6, we started to create the mining structure. As you've seen in step 6, the mining structure accepts the dataset from both the relational database and from an Analysis Services cube.

In step 7, we can choose to create the mining model beside the mining structure, or not.

## THE MINING MODEL

The mining model contains a data mining algorithm and a set of variables, for example, **input**, **key**, and **predictable**. The mining algorithm will be applied on the existing dataset by considering the variable configuration, and finally the mining model will contain the result of the data analysis. A mining structure may contain one or more mining models. In a data mining solution, you would need to apply multiple mining models on a same mining structure to find the best algorithm that satisfies the test result set. As you've seen in step 7, there are nine data mining algorithms that can be used in mining models.

Steps 8 and 9 show how to set the existing data in **`vTargetMail`** as an input for the mining structure. In step 9, we selected **vTargetMail** as the **Case** table. The **`Case`** table is the main input of the mining structure and algorithm. Sometimes, the **`Case`** table cannot be as simple as one table, and you might need to provide a many-to-one relationship in it. In those scenarios, you should choose the second table as **`nested`** and the first one as **`case`**.

## DATA MINING VARIABLES

Step 10 is an important step in configuring the mining model. There are three types of variables in mining algorithms: key, input, and predictable. The key variable, as its name suggests, is the variable that defines the key field in the dataset to be analyzed. The **Key** column in our previous example is **CustomerKey**. Input variables are columns whose effects we want to see on the result set. As you've seen, we have chosen columns such as **Age**, **Education**, and **Occupation** as input. The reason is that we want to analyze the effect of the values of these variables on the `predictable` variable. You can find out which columns are most useful to be set as input variables based on the analysis done in the data preparation step and also in the problem definition step.

## THE CONTENT TYPE

In step 11, we used the **Detect** button to detect the content type of the input variables. The content type means that the data type is fetched based on the existing values for that column in the provided dataset. There are nine content types, but the most common are discrete, continuous, discretized, cyclical, and ordered. Short notes on these are presented in the following list:

- **Discrete**: Data values that are separate, for example, the red, yellow, and blue color values
- **Continuous**: Data values that change continuously, for example, age or salary
- **Cyclical**: Data values that are in a cyclical order, for example, the days of a week
- **Ordered**: Data values that are in a sequential order, for example, the days of a month
- **Discretized**: Data values that are continuous but bucketed into categories, and as a result behave discretely

In step 11, you saw that age and yearly income are continuous, while other input variables are discrete. Data mining algorithms work with different content types; for example, Time Series have better results with ordered and cyclical content types, decision trees perform well with discrete values, while regression algorithms have better results with continuous and discrete variables.

## THE TRAINING AND TEST SET

When we apply a dataset to the data mining algorithm, a part of the data acts like the training set and the other as the test set. The training set is a subset of the data on which the mining algorithm will be applied; this subset participates directly in generating the mining model. When the mining model is ready, another subset of data will be used to test the model to check the mining model result; this subset is named test set. When you work with prediction algorithms, it is important to set a part of the dataset as the test set, because if you use all of the dataset as a training set, then you have no way to identify the mining model that generates the result correctly (or almost correctly). In this example, we used 70 percent of the data as a training set while the remaining 30 percent was used as a test set (step 12). Defining a test set is only important when you want to perform prediction. For data analysis without the prediction functionality, you won't require a test set.

In steps 15 to 17, we changed the content type of the **Age** and **Yearly Income** columns to **Discretized**. Since the content is continuous, we need to bucket values. The data mining algorithm applies bucketing methods based on the **`DiscretizationMethod`** and **`DiscretizationBucketCount`** properties. We set **`DiscretizationBucketCount`**

as **`16`**, which means age will be categorized in five categories (buckets). In order to apply the data mining algorithm on the training set, the mining structure should be processed (step 18).

## DATA MINING VIEWERS

Each data mining algorithm has a different set of mining viewers. Data mining viewers are visualization tools that show the structure of data based on the training dataset. In this example, you've seen two types of mining viewers: decision tree viewer and dependency network viewer.

The decision tree viewer that you've seen from steps 20 to 22 shows the decision tree built over the training dataset. Each node shows a decision, and nodes can be colored based on the predictable column value (step 20). You've learned how to follow a decision path and analyze the result.

The dependency network viewer shows the dependency of the **`predictable`** variable to all input variables. You can see the strongest links by changing the dependency bar in this viewer.

## MICROSOFT ASSOCIATION RULES

Association rules is a data mining algorithm that identifies relationships between different variables in an existing dataset; this algorithm literally finds rules existing on the association relationships of variables. One of the most common scenarios that can be solved with this algorithm is the market-basket analysis. In the following example, you will learn about the association rule algorithm and how to analyze the results of this algorithm.

## TIME FOR ACTION – THE MICROSOFT ASSOCIATION RULE

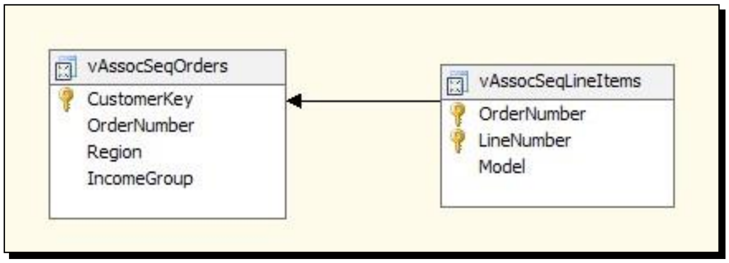
Assume this scenario as an example: the AdventureWorks company wants to identify a set of products that are usually purchased together. The company can use this information to put these products beside each other or add similar items next to each other in a store to achieve higher sales. This is known as market-basket analysis because it relates to the analysis of products in a single basket in each purchase. In this example, we will use the `AdventureWorksDW2012` database.

1. Open the AdventureWorksDW2012 data source view from the previous example.

2. Right-click on an empty area in the data source view and select **Add/Remove Tables**.

3. In the **Add/Remove Tables** window, add **vAssocSeqOrders** and **vAssocSeqLineItems** to the DSV.

4. After adding these views, create a relationship between the two views in this manner: drag-and-drop the **Order Number** column from **vAssocSeqLineItems**  to **vAssocSeqOrders** as shown in the following screenshot:

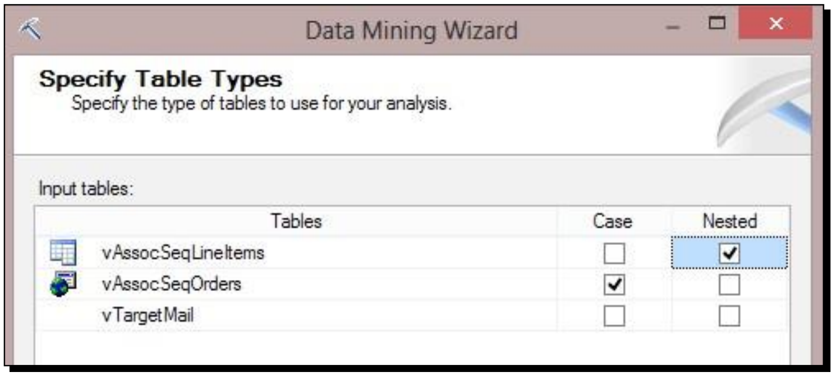


5. Create a new mining structure by right-clicking on the `mining structure` folder  in **Solution Explorer**. Choose a definition method from an existing relational database or data warehouse.

6. In the **Create the Data Mining Structure** window, select **Create Mining Structure** as the mining model and then choose the **Microsoft Association Rules** algorithm from the drop-down list.

7. Choose the data source view as **Adventure Works DW2012**.

8. In the **Specify Table Types** window, set **vAssocSeqLineItems** as **Nested** and **vAssocSeqOrders** as **Case** as shown in the following screenshot:



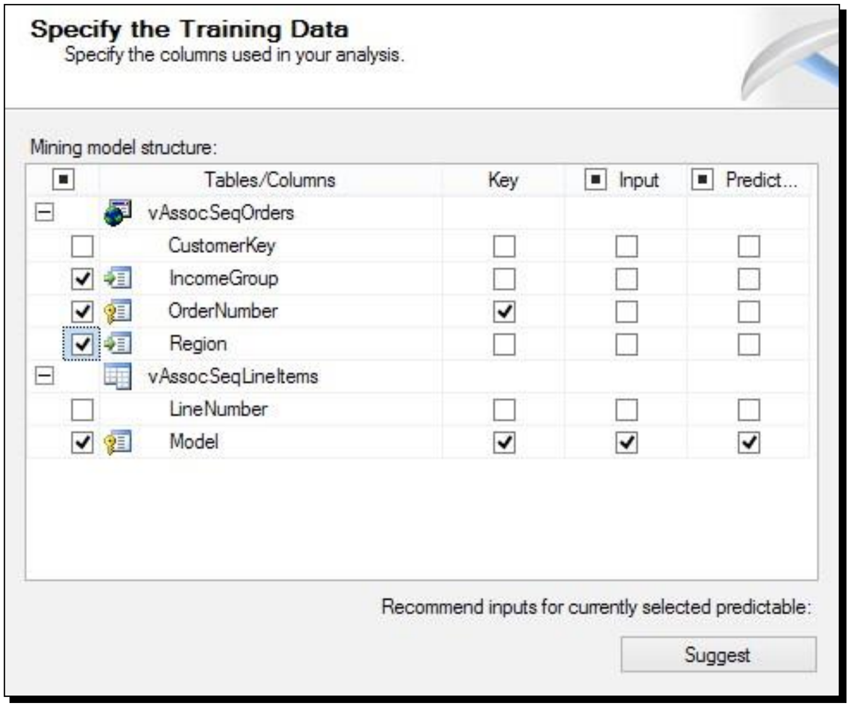
### Specify Table Types

Specify the type of tables to use for your analysis.

Input tables:

| Tables | Case | Nested |
|---|---|---|
| vAssocSeqLineItems | ☐ | ☑ |
| vAssocSeqOrders | ☑ | ☐ |
| vTargetMail | ☐ | ☐ |

9. In the **Specify the Training Data** window, set **OrderNumber** as **Key** and **Model** as **Key**, **Input**, and **Predictable**. The following screenshot shows this configuration:



### Specify the Training Data

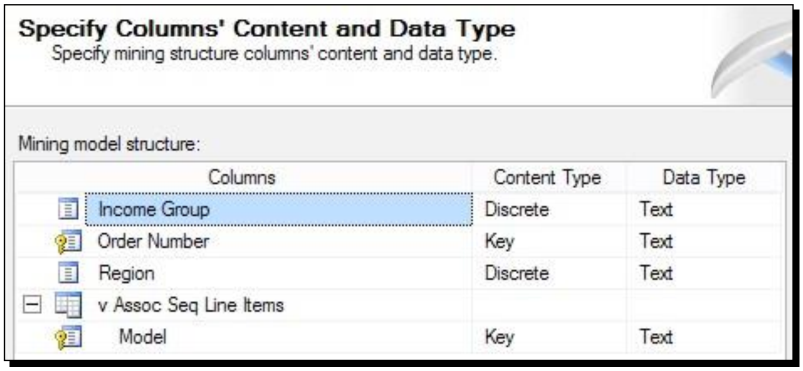
Specify the columns used in your analysis.

Mining model structure:

| ■ | Tables/Columns | Key | ■ Input | ■ Predict... |
|---|---|---|---|---|
| ⊟ | vAssocSeqOrders | | | |
| ☐ | CustomerKey | ☐ | ☐ | ☐ |
| ☑ | IncomeGroup | ☐ | ☐ | ☐ |
| ☑ | OrderNumber | ☑ | ☐ | ☐ |
| ☑ | Region | ☐ | ☐ | ☐ |
| ⊟ | vAssocSeqLineItems | | | |
| ☐ | LineNumber | ☐ | ☐ | ☐ |
| ☑ | Model | ☑ | ☑ | ☑ |

Recommend inputs for currently selected predictable:

10. In the **Specify Column's Content and Data Type** window that appears next,  as shown in the following screenshot, click on **Detect**:



Specify Columns' Content and Data Type

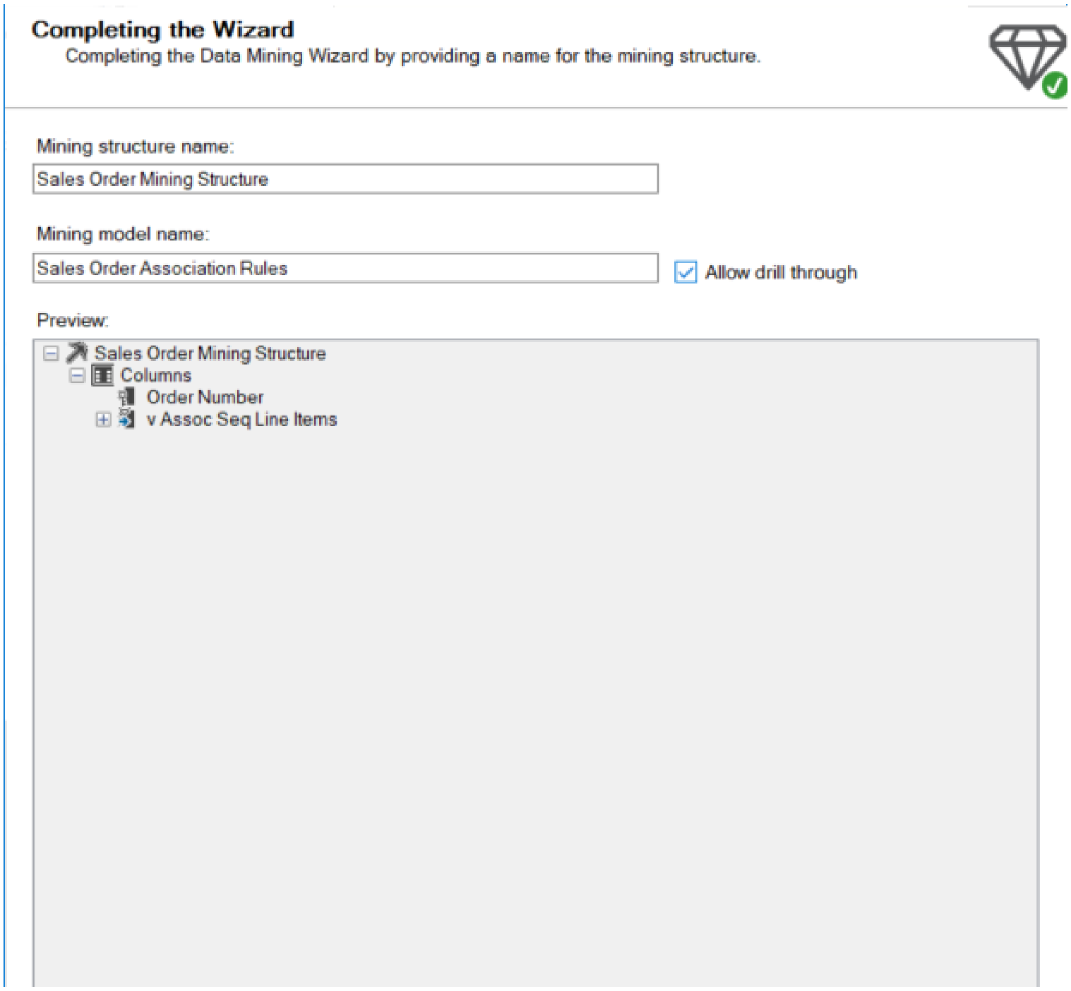
Specify mining structure columns' content and data type.

Mining model structure:

| Columns | Content Type | Data Type |
|---|---|---|
| Income Group | Discrete | Text |
| Order Number | Key | Text |
| Region | Discrete | Text |
| v Assoc Seq Line Items | | |
| Model | Key | Text |

11. In the **Create Testing Set** window, change the percentage of data for testing to `0`.

12. Rename **mining structure** to `Sales Order Mining Structure` and **mining model** to `Sales Order Association Rules`. Check the option for **Allow drill through** and click on **Finish**.



**Completing the Wizard**

Completing the Data Mining Wizard by providing a name for the mining structure.

Mining structure name:

Sales Order Mining Structure

Mining model name:

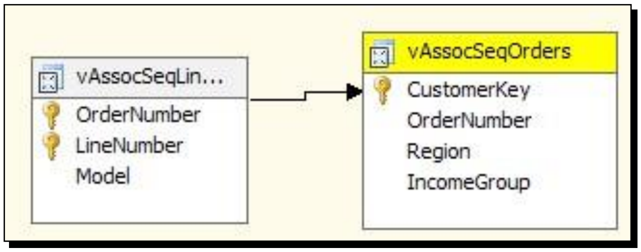
Sales Order Association Rules

☑ Allow drill through

Preview:

- Sales Order Mining Structure
  - Columns
    - Order Number
    - v Assoc Seq Line Items

**13.** After completing the wizard, you will see a diagram of a case and nested table in  the mining structure designer, as shown in the following screenshot:

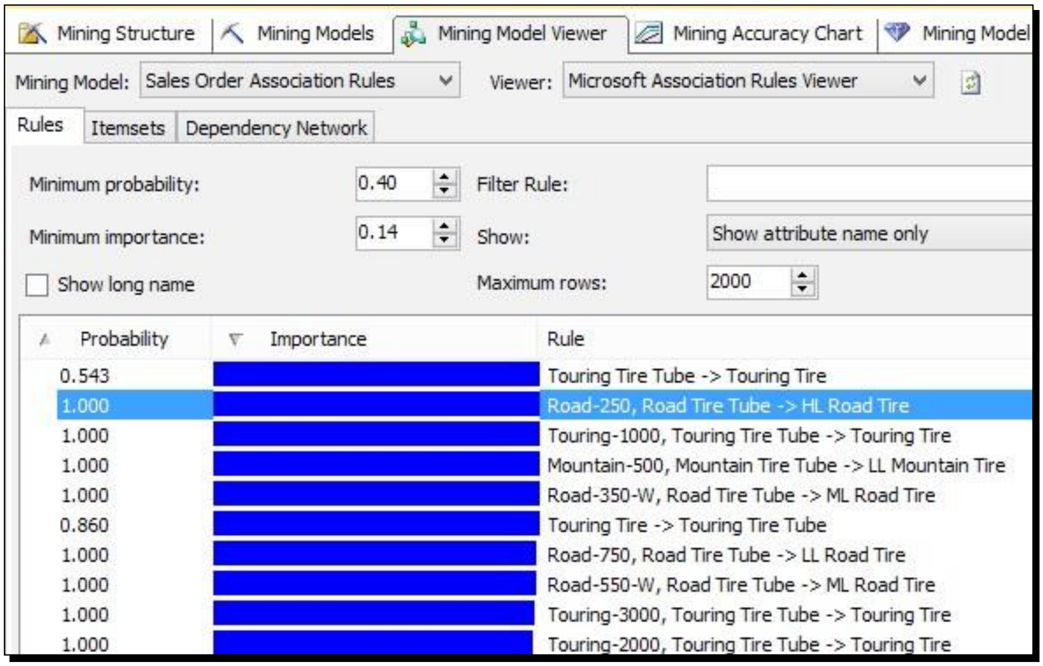


**14.** Right-click on **Sales Order Mining Structure** and process it.

**15.** Go to the **Mining Model Viewer** tab.

**16.** The default viewer is **Microsoft Association Rule Viewer**. In the **Rules** tab,  you can see a list of rules fetched by this algorithm applied on the existing dataset.

**17.** Change the **Show** option to **Show attribute name only** by choosing it from the drop-down list. Change the order of columns to sort by **Importance** or **Probability** (that is descending) and investigate the result set. As you can see in the following screenshot, there are some rules; for example, you wanted to buy Road-250 and Road Tire Tube but you ended up buying HL Road Tire:



Mining Model: Sales Order Association Rules

Viewer: Microsoft Association Rules Viewer

Minimum probability: 0.40

Filter Rule:

Minimum importance: 0.14

Show: Show attribute name only

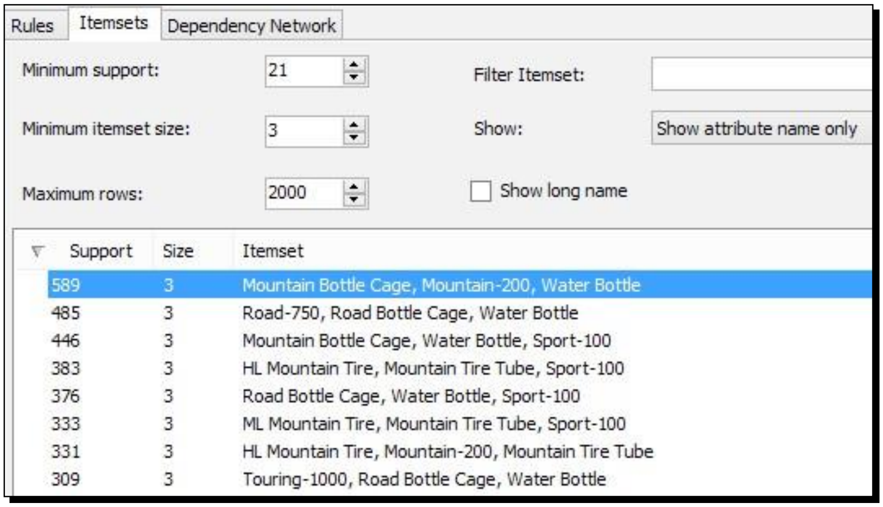
□ Show long name

Maximum rows: 2000

| Probability | Importance | Rule |
|---|---|---|
| 0.543 | ▇ | Touring Tire Tube -> Touring Tire |
| 1.000 | ▇ | Road-250, Road Tire Tube -> HL Road Tire |
| 1.000 | ▇ | Touring-1000, Touring Tire Tube -> Touring Tire |
| 1.000 | ▇ | Mountain-500, Mountain Tire Tube -> LL Mountain Tire |
| 1.000 | ▇ | Road-350-W, Road Tire Tube -> ML Road Tire |
| 0.860 | ▇ | Touring Tire -> Touring Tire Tube |
| 1.000 | ▇ | Road-750, Road Tire Tube -> LL Road Tire |
| 1.000 | ▇ | Road-550-W, Road Tire Tube -> ML Road Tire |
| 1.000 | ▇ | Touring-3000, Touring Tire Tube -> Touring Tire |
| 1.000 | ▇ | Touring-2000, Touring Tire Tube -> Touring Tire |

**18.** To view a list of records from a rule set, right-click on that record and select one of the drill-through options.

**19.** Go to the **Itemsets** tab. In this tab, you will see a set of items based on the set  of items fetched with the association rule algorithm. Change the **Show** drop-down list to **Show attribute name only**, change the size of the set to **3**, and change the order of the result set to the descending order of the **Support** column. For example, the first itemset in the following screenshot shows that, in the provided dataset,  **589** sale order transactions had the **Mountain Bottle Cage, Mountain-200, Water Bottle** option in the same basket:



Rules | Itemsets | Dependency Network

Minimum support: 21

Minimum itemset size: 3

Maximum rows: 2000

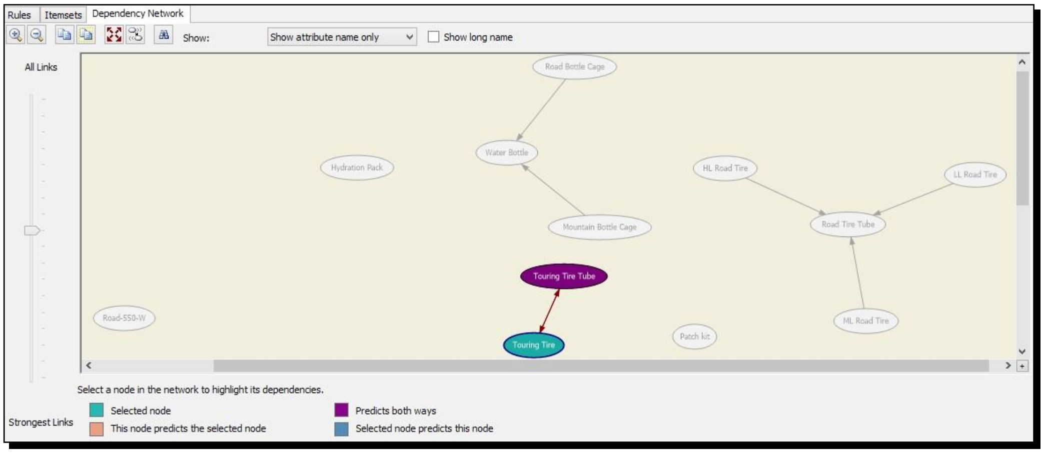
Filter Itemset:

Show: Show attribute name only

☐ Show long name

| Support | Size | Itemset |
|---|---|---|
| 589 | 3 | Mountain Bottle Cage, Mountain-200, Water Bottle |
| 485 | 3 | Road-750, Road Bottle Cage, Water Bottle |
| 446 | 3 | Mountain Bottle Cage, Water Bottle, Sport-100 |
| 383 | 3 | HL Mountain Tire, Mountain Tire Tube, Sport-100 |
| 376 | 3 | Road Bottle Cage, Water Bottle, Sport-100 |
| 333 | 3 | ML Mountain Tire, Mountain Tire Tube, Sport-100 |
| 331 | 3 | HL Mountain Tire, Mountain-200, Mountain Tire Tube |
| 309 | 3 | Touring-1000, Road Bottle Cage, Water Bottle |

**20.** In the last tab of this mining model viewer, you will see the **Dependency Network** viewer. Change the **Show** option to **Show attribute name only** by choosing it in the drop-down list and shift the **Dependency Network** bar to **lower**. You will see the relationship between items in a basket and rules fetched. The following screenshot shows a magnified part of the dependency network:



Show: Show attribute name only

☐ Show long name

Select a node in the network to highlight its dependencies.

🟦 Selected node

🟧 This node predicts the selected node

🟪 Predicts both ways

🟦 Selected node predicts this node

## WHAT JUST HAPPENED?

In this example, you've learned how to use the Microsoft association rules algorithm to solve the market-basket analysis scenario. The AdventureWorksDW2012 database contains two views for sales order transactions, that is, **`vAssocSeqOrders`** and **`vAssocSeqLineItems`**. The **`vAssocSeqOrders`** view contains one record per sales order and **`vAssocSeqLineItems`** contains one record per item in each sales order. The combination of these two tables provides a list of items bought in each sales order. We used this dataset to train the Microsoft association rules algorithm in this example. Steps 2 to 4 show how to add these tables and create a relationship between them in the data source view. The relationship generated is based on the order number.

We used the Microsoft association rules algorithm (step 6) in this example because it identifies rules between variables existing in a dataset and provides a set of items that are associated with each other. In step 8, we set the **`vAssocSeqOrders`** table as a **Case** table because this table contains header records of sales orders. We've set **`vAssocSeqLineItems`** as a **Nested** table as it contains detailed item sales records.

Step 9 is an important step because we set variable types in this step. **Order Number** and **Model** is used as a key because **Order Number** itself cannot identify a record. Moreover, we used **Model** as an input as well because we want to see the effect of buying a model based on another model. We also choose the model to be predictable. The configuration of a variable and a column's type is one of the most important steps in data mining solutions.

We've set the entire dataset to act as a training set, so we've changed the test set (step 11) to `0` because, in this example, we didn't want to predict. The main purpose of this scenario is descriptive data analysis.

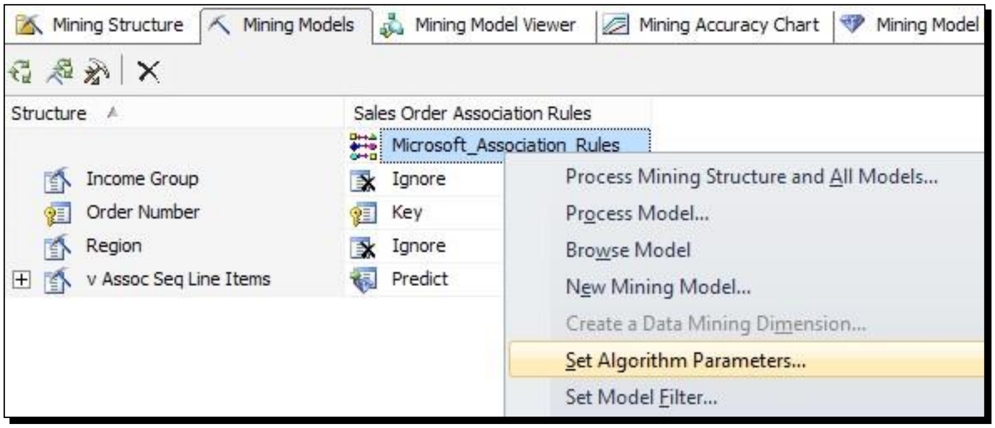
In step 13, you can see that the case table is shown with a yellow header and nested tables with grey headers in the mining structure designer. After processing the mining structure and the mining model in step 14, you can view the result in the **Mining Model Viewer** tab of the **Mining Structure Designer** window.

The Microsoft association rule viewer contains three types of viewers, namely rules, itemsets, and the dependency network. In step 17, you can see a sample screenshot of the **Rules** tab. The association rules algorithm is applied on the existing dataset and, as a result, it generates rules that are based on the input and **`predictable`** variables. As a result, in this viewer, you can see a list of rules that you can filter using some parameters. First of all, rules show attribute names and values by default (which we've changed to **Show attribute name only** in step 17).

## ALGORITHM PARAMETERS

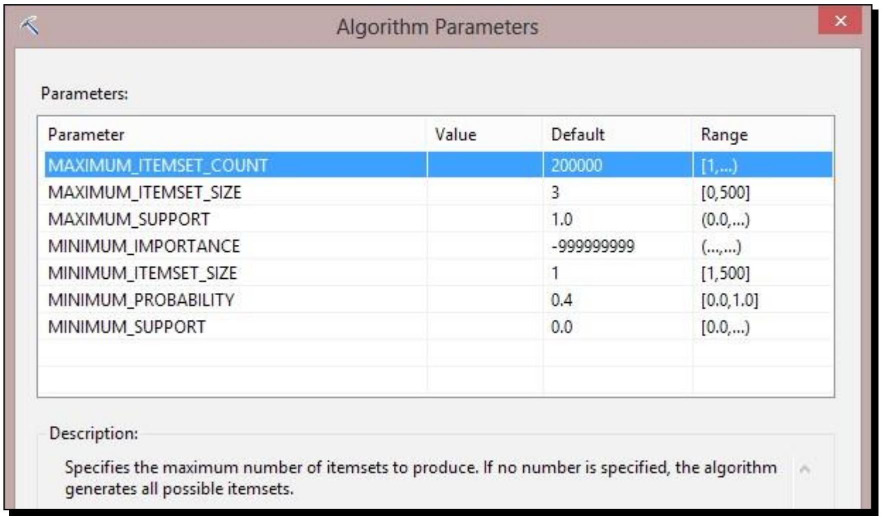
You can filter the required options in the minimum probability or the minimum importance tools. Probability, importance, and support are the main parameters in the association rules algorithm. Support shows the occurrence of an item or itemset; it is also called frequency. Probability is a value calculated by Analysis Services that shows the confidence of the prediction of a predictable value based on the items of input values. The importance parameter is calculated based on the likelihood of the condition *if X, then Y* resulting in *Y*. The importance parameter is also called lift. A detailed description of these algorithm parameters is out of the scope of this book. You can read more about them with a brief description in MSDN books, at **http://technet.microsoft.com/en-us/library/ cc280428.aspx**.

In the **Rules** tab, as you've seen in step 17, you can look at rules and filter them on the  basis of minimum probability or importance to get rid of rules that are not reliable enough  to be considered. Algorithm parameters can also be changed in the **Mining Model** tab by right-clicking on the **mining model** column and choosing **Set Algorithm Parameters…** as shown in the following screenshot:



| Structure | Sales Order Association Rules |
|---|---|
| | Microsoft_Association_Rules |
| Income Group | Ignore |
| Order Number | Key |
| Region | Ignore |
| v Assoc Seq Line Items | Predict |

Each data mining algorithm has a different set of parameters. For example, the clustering algorithm can be modified with a number of clusters while association rules can be modified with support, probability, and importance. The following screenshot shows a list of parameters that can be modified when you work with the association rules algorithm:



Algorithm Parameters

Parameters:

| Parameter | Value | Default | Range |
|---|---|---|---|
| MAXIMUM_ITEMSET_COUNT | | 200000 | [1,...) |
| MAXIMUM_ITEMSET_SIZE | | 3 | [0,500] |
| MAXIMUM_SUPPORT | | 1.0 | (0.0,...) |
| MINIMUM_IMPORTANCE | | -999999999 | (...,...) |
| MINIMUM_ITEMSET_SIZE | | 1 | [1,500] |
| MINIMUM_PROBABILITY | | 0.4 | [0.0,1.0] |
| MINIMUM_SUPPORT | | 0.0 | [0.0,...) |

Description:

Specifies the maximum number of itemsets to produce. If no number is specified, the algorithm generates all possible itemsets.

Step 18 shows how you can use the **drill-through** option in the viewer to see the actual records that cover each rule or itemset. Step 19 shows a different view, which is the itemset viewer. You can change the size of an item set; for example, size **3** will show only those itemsets with at least three items. You can also view the frequency of the itemset in the **Support** column, and you can filter it by changing the minimum support.

Microsoft association rules give us the benefit of seeing the result set in the dependency network viewer. The dependency network viewer, in this mining model, shows how items are dependent on each other based on the market-basket analysis. When you change the dependency bar (step 20), you can see that the lower you go through the dependency bar, you will see more dependent items; this means that those items are bought together most of the times (such as **Touring Tire Tube** and **Touring Tire** shown in the screenshot of step 20).

## SUMMARY

In this chapter, you've learned some fundamentals about data mining algorithms. You learned that Microsoft Analysis Services provides nine algorithms for data mining. Data mining algorithms are divided into five categories based on their functionality. You also learned that a data mining solution is not a one-way solution. It is a circular life cycle that starts with problem definition, then data preparation, continues to model design and implementation, followed by testing the results and deployment, and finally finding the next problem.

We went through two common problems that could be solved with descriptive data mining algorithms, namely the analysis of existing bike buyers and market-basket analysis. For the analysis of existing bike buyers, we used the Microsoft Decision Tree algorithm that provided a tree of decisions made based on the value of the variables. You've learned how to analyze the result set of an algorithm in mining model viewers. The market-basket analysis utilized the Microsoft association rules algorithm to identify a set of products or items that were purchased together. You also learned that each data mining algorithm has a different set of parameters, which can be filtered or configured to help get better result out of the mining model.

In this chapter, you've learned about data mining algorithms, also called descriptive models, that analyze existing dataset. In the next chapter, you will learn how to use prediction in these algorithms and also use algorithms such as Time Series to identify patterns over a period of time. You will also learn about DMX, which is the query language for data mining scripts.

# 8

# Identifying Data Patterns – Predictive Models in SSAS

*In the previous chapter, you learned how to use data mining for data analysis. In this chapter, we will go one step further to reveal predictions based on an existing pattern. Prediction is one of the most interesting topics of Business Intelligence, because it gives us an estimate of the future behavior of the business.*

*The Microsoft SSAS engine understands a specific language for prediction and works with data mining algorithms called* **Data Mining Extensions** *(***DMX***). In this chapter, you will learn how to compare mining models and how to apply prediction on an existing data pattern with DMX. The examples of this chapter will use the scenarios covered in the previous chapter. Also, an example of Microsoft Time Series algorithm usage will be discussed at the end of this chapter.*

## FINDING THE BEST ALGORITHM

As a data mining developer, you apply different data mining algorithms and models on  a dataset. Different mining models will return different patterns and different result sets because each mining algorithm works in a different way. Finding the best mining model is  a process that depends on the problem, dataset, and the output of algorithm.

Microsoft provides tools such as Lift Chart and Profit Chart for comparing the result of  mining models. As you may recall from the previous chapter, the test set is part of the data that is retained intact to be tested with the recognized pattern of the mining model. Lift Chart and Profit Chart will show how the result of the mining model works compared to actual data in the test set.

## TIME FOR ACTION – FINDING THE BEST MINING MODEL WITH LIFT CHART AND PROFIT CHART

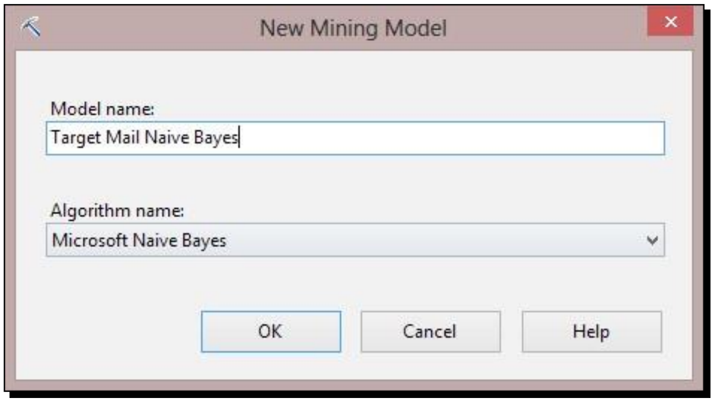
In this example, we will add two other data mining algorithms, Naïve Bayes and Clustering, to the Target mail mining Structure example from the previous chapter. Then, we compare Lift Chart and Profit Chart for these algorithms to see which one works better compared to the test set. Perform the following steps to add the algorithms:

1. Open Target mail mining Structure from the first example of the previous chapter.

2. In the mining structure designer, go to the **Mining Models** tab and create a  new mining model in one of the following ways:

   - ⑨ Right-click anywhere on the **Mining Models** tab and choose **New Mining Model**
   - ⑨ Click on the icon that shows the **Create a related mining model** option  on hovering

3. In the **New Mining Model** window, choose **Microsoft Naïve Bayes** as the algorithm and name it **`Target Mail Naive Bayes`**, as shown in the following screenshot:



New Mining Model

Model name:
Target Mail Naive Bayes

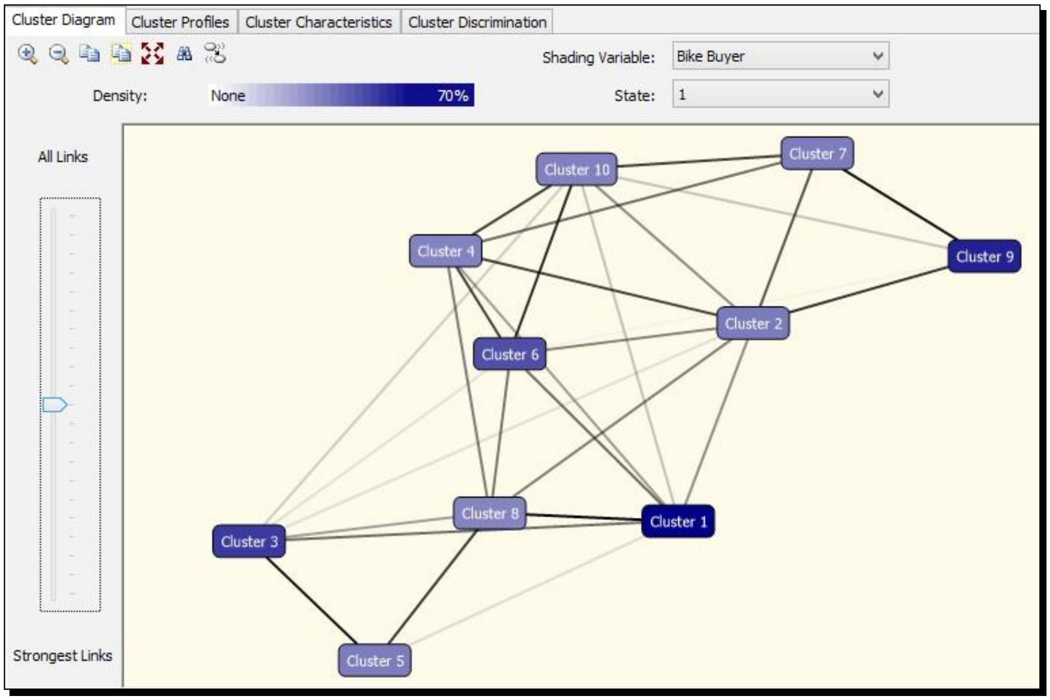
Algorithm name:
Microsoft Naive Bayes

4. Repeat steps 2 and 3 to create a new mining model with the Microsoft Clustering algorithm and name this mining model **`Target Mail Clustering`**.

5. Deploy and process the SSAS project.

6. Go to the **Mining Model Viewer** tab and choose **Target Mail Clustering** in the  mining model drop-down list.

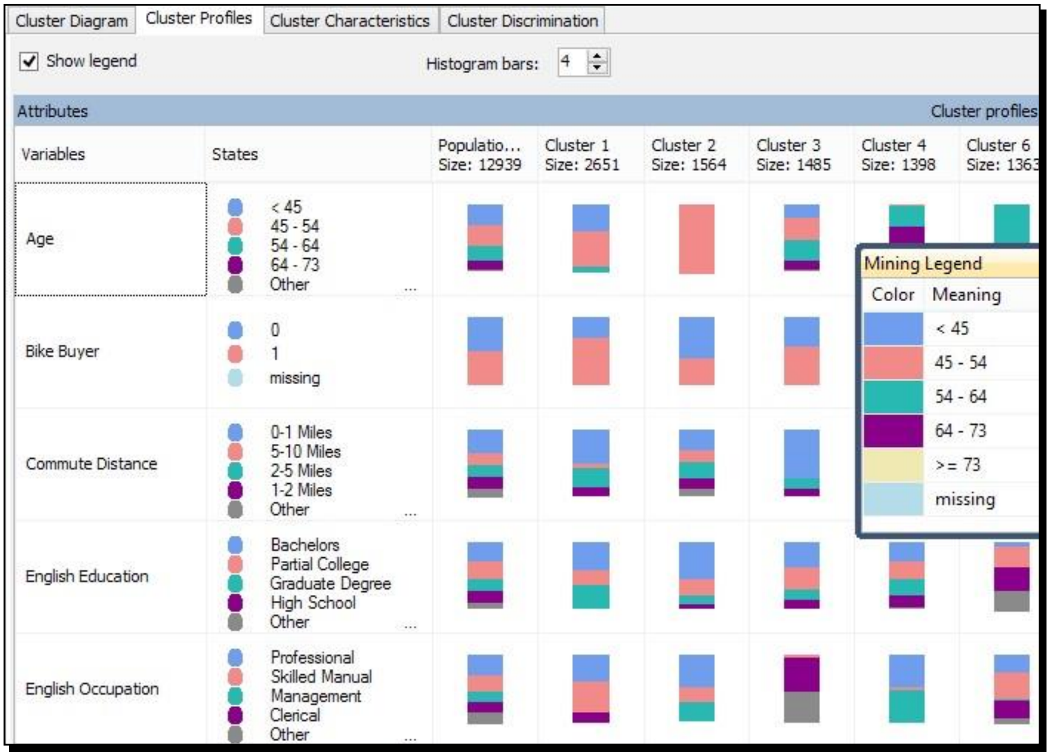
7. You will now see four viewer tabs—**Cluster Diagram**, **Cluster Profiles**, **Cluster Characteristics**, **and Cluster Discrimination**. The first tab is **Cluster Diagram**.  Change the **Shading Variable** option to **Bike Buyer** and change the **State** option  to **1**. You will see a list of clusters with links between them. You can change the bar on the left-hand side to show the stronger or weaker links between the clusters.  The shading configuration that we created means that the cluster with a darker  blue background will be the cluster with more portions of bike buyers. In the following screenshot, **Cluster 1**, **Cluster 9**, and **Cluster 3** have the most density of bike buyers. By hovering the mouse on any of the clusters, you will see the density of the shading variable (which bike buyer is to be numbered as 1) in percentage.



Shading Variable: Bike Buyer

State: 1

**8.** Click on the **Cluster Profiles** tab. In this tab, you can see how each cluster is populated. You would see all input variables and the proportion of values for  each variable in each cluster. For example, the following screenshot shows that **Cluster 1** contains more bike buyers, and the **English Occupation** tab of this cluster shows that most of them are professionals or skilled manual laborers. You can also see that all customers in **Cluster 2** (which has low density of bike buyers) are in  the age group 45 to 54.



☑ Show legend

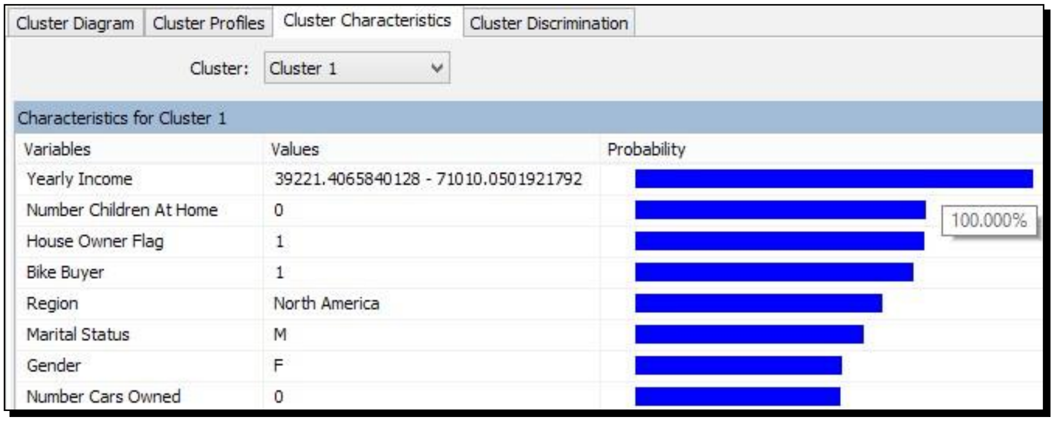
Histogram bars: 4

| Variables | States | Populatio... Size: 12939 | Cluster 1 Size: 2651 | Cluster 2 Size: 1564 | Cluster 3 Size: 1485 | Cluster 4 Size: 1398 | Cluster 6 Size: 1363 |
|---|---|---|---|---|---|---|---|
| Age | < 45<br>45 - 54<br>54 - 64<br>64 - 73<br>Other ... | | | | | | |
| Bike Buyer | 0<br>1<br>missing | | | | | | |
| Commute Distance | 0-1 Miles<br>5-10 Miles<br>2-5 Miles<br>1-2 Miles<br>Other ... | | | | | | |
| English Education | Bachelors<br>Partial College<br>Graduate Degree<br>High School<br>Other ... | | | | | | |
| English Occupation | Professional<br>Skilled Manual<br>Management<br>Clerical<br>Other ... | | | | | | |

Mining Legend

| Color | Meaning |
|---|---|
| | < 45 |
| | 45 - 54 |
| | 54 - 64 |
| | 64 - 73 |
| | >= 73 |
| | missing |

***9.*** The **Cluster Characteristics** tab shows the percentage of each variable value in each cluster. If you choose **Cluster 1** in the cluster's drop-down list, you will see the probability of the variable's values for this cluster. For example, the following screenshot shows that all customers in this cluster are in the yearly income range of 39K to 71K. If you hover your mouse on the **Probability** bar, you will see the percentage of probability. You can choose other clusters to find out the characteristics of those clusters in the diagram. The following screenshot also  shows that most of the customers in this cluster have no children at home:

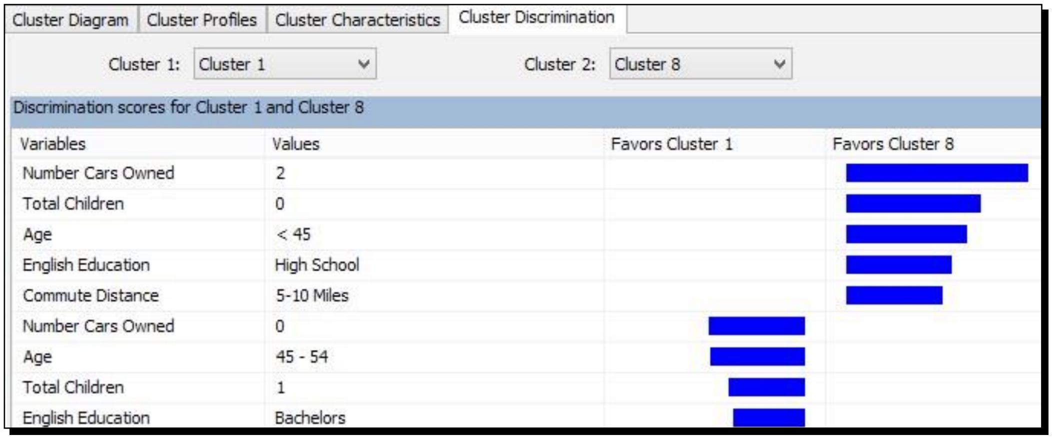


Cluster: Cluster 1

Characteristics for Cluster 1

| Variables | Values | Probability |
|---|---|---|
| Yearly Income | 39221.4065840128 - 71010.0501921792 | |
| Number Children At Home | 0 | |
| House Owner Flag | 1 | |
| Bike Buyer | 1 | |
| Region | North America | |
| Marital Status | M | |
| Gender | F | |
| Number Cars Owned | 0 | |

***10.*** Open the **Cluster Discrimination** tab. choose **Cluster 1** and **Cluster 8** sequentially. You will see the distribution of variable values in these two different clusters. The following screenshot shows that **Cluster 8** contains the customers who have two cars at most, but **Cluster 1** contains most of the customers who have no cars:



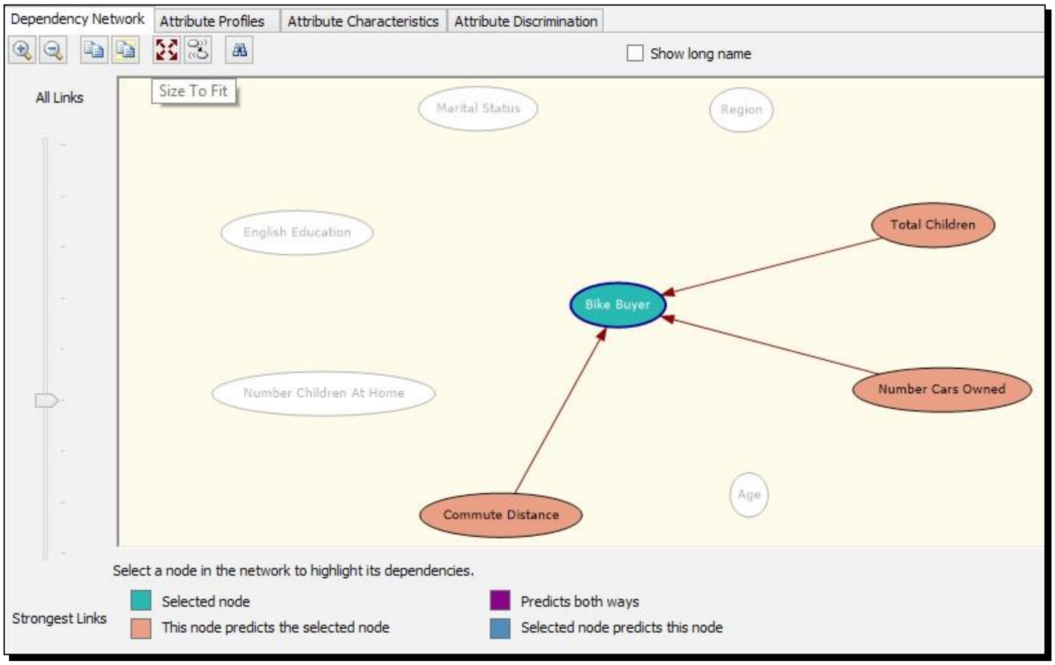
Cluster 1: Cluster 1 Cluster 2: Cluster 8

Discrimination scores for Cluster 1 and Cluster 8

| Variables | Values | Favors Cluster 1 | Favors Cluster 8 |
|---|---|---|---|
| Number Cars Owned | 2 | | |
| Total Children | 0 | | |
| Age | < 45 | | |
| English Education | High School | | |
| Commute Distance | 5-10 Miles | | |
| Number Cars Owned | 0 | | |
| Age | 45 - 54 | | |
| Total Children | 1 | | |
| English Education | Bachelors | | |

***11.*** Select **Target Mail Naïve Bayes** as **Mining Model** and you will see four different  tabs as the mining model viewers.

***12.*** The first tab shows the dependency diagram (which you are familiar with from the previous chapter). As you see in the following screenshot, scrolling the dependency bar down will show only the strongest links. If you compare the results of the **Dependency Network** tab of the Naive Bayes mining model with the dependency network of the decision tree mining model from the previous chapter, then you  will find some differences in the strongest links.

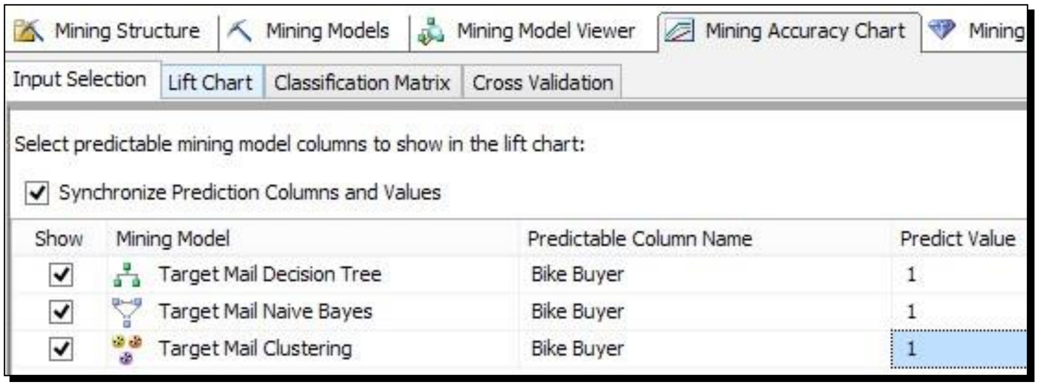


**13.** The **Attribute Profiles** tab is similar to the **Cluster Profile** tab. It shows the distribution of the attribute values that end up with the value **0** or **1** for the `predictable` variable (**Bike Buyer**).

**14.** The **Attribute Characteristics** tab shows the probability of attribute values for bike buyers or non-bike buyers. You can configure it by setting the `predictable` variable and its value.

**15.** The last mining viewer tab is **Attribute Discrimination**. Choose the value **0** and **1** for **Bike Buyer**. And compare the attribute's probability in both sections. You will see that the attribute value **0** for the number of cars owned mostly favors bike buyers, but those who have two cars are less likely to buy a bike.

**16.** Click on the **Mining Accuracy Chart** tab. In the **Input Selection** tab, change the **Predict Value** of all mining models to **1**. Leave other options as is, as shown in the following screenshot:

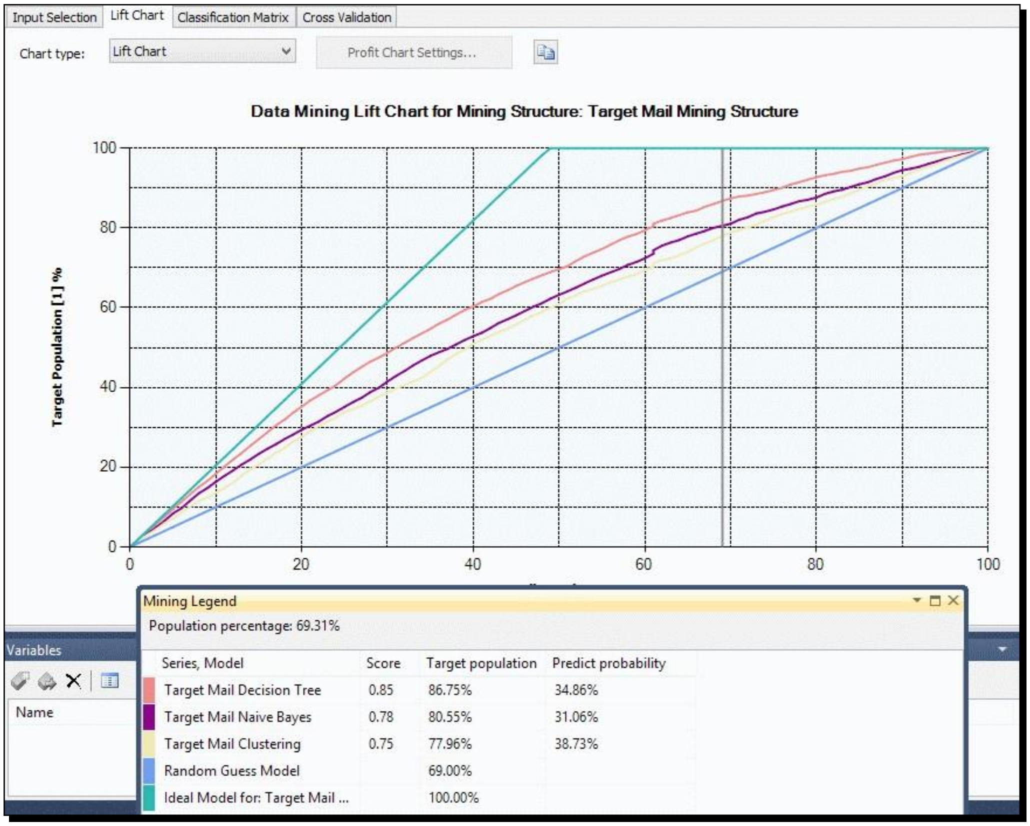


Select predictable mining model columns to show in the lift chart:

☑ Synchronize Prediction Columns and Values

| Show | Mining Model | Predictable Column Name | Predict Value |
|---|---|---|---|
| ☑ | Target Mail Decision Tree | Bike Buyer | 1 |
| ☑ | Target Mail Naive Bayes | Bike Buyer | 1 |
| ☑ | Target Mail Clustering | Bike Buyer | 1 |

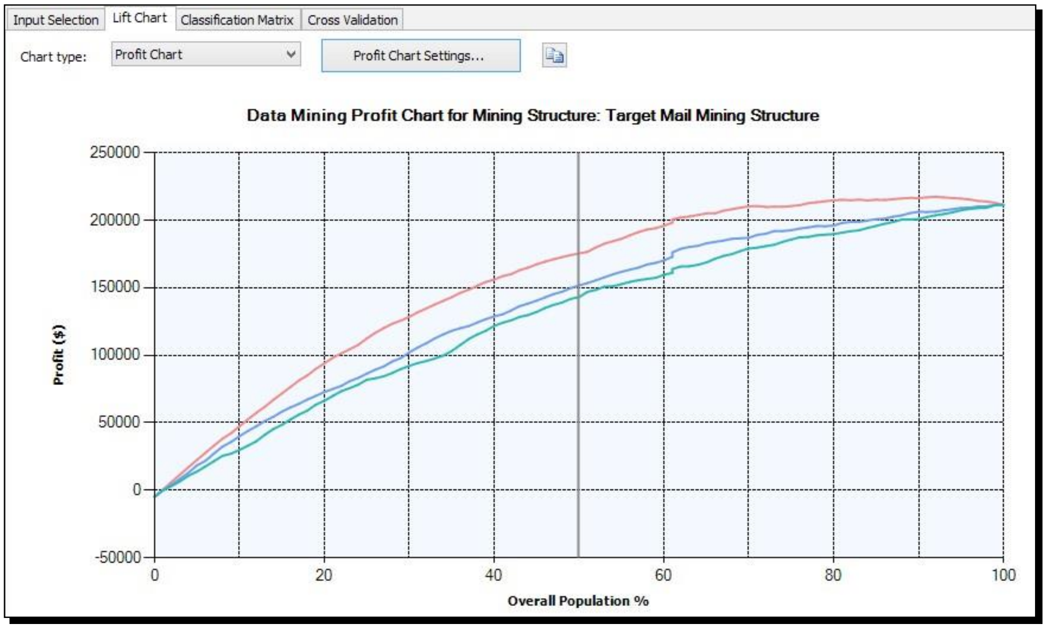
**17.** Go to the **Lift Chart** tab and you will see a line chart that shows one line per mining model. There are also two other lines: one for random guesses and another for an ideal model. The overall population is listed on the horizontal axis and the vertical axis shows the target population. The mining algorithm that produces most of the target with a lower number of overall population works best. For a more detailed discussion on this diagram, go to the *What just happened?* section. In the following screenshot, you see that the decision Tree mining model is the best algorithm based on this chart:



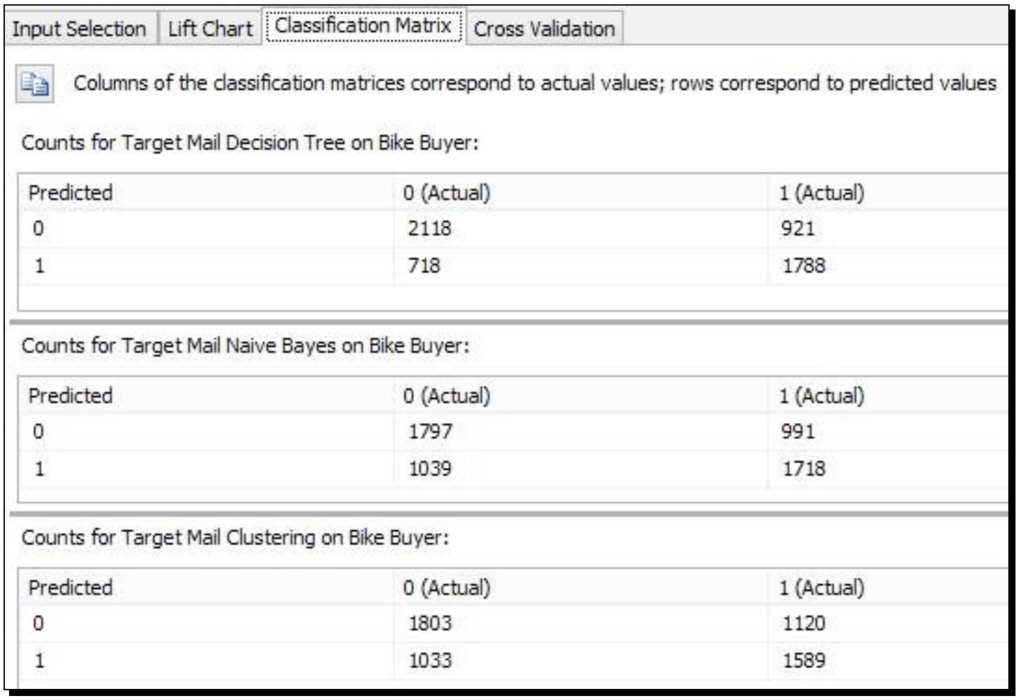
Population percentage: 69.31%

| | Series, Model | Score | Target population | Predict probability |
|---|---|---|---|---|
| 🟥 | Target Mail Decision Tree | 0.85 | 86.75% | 34.86% |
| 🟪 | Target Mail Naive Bayes | 0.78 | 80.55% | 31.06% |
| 🟨 | Target Mail Clustering | 0.75 | 77.96% | 38.73% |
| 🟦 | Random Guess Model | | 69.00% | |
| 🟩 | Ideal Model for: Target Mail ... | | 100.00% | |

**18.** Change **Chart type** in the drop-down list to **Profit Chart**, and in the **Profit Chart Settings…** dialog box, accept everything with the default values. When the profit chart is drawn, you will see another line chart that shows one line per mining model. Profit charts show the profit calculated based on the cost of overall population. For example, the profit of using a mining model for sending an e-mail to prospective bike buyers will be calculated based on the cost of sending e-mail to the prospective buyers minus the cost of sending e-mail to those customers who bought a bike. For more information about profit charts, refer to the *What just happened?* section. The following screenshot shows the profit chart for the mining structure:



19. Click on the **Classification Matrix** tab. This tab shows one matrix per mining model. In each matrix, you can see the value predicted by actual style matrix. For example, if the value for **Predicted** is **1**, it means that the mining models predicted that this customer would be a bike buyer, but if the value for **Actual** is **0**, it means that the customer actually didn't buy a bike. So the best answer for our prediction would be **1** for **Predicted** and **Actual**. In the following screenshot, you can see that the decision tree generates a better result again by predicting **1788** cases to be the actual bike buyers:



Columns of the classification matrices correspond to actual values; rows correspond to predicted values

Counts for Target Mail Decision Tree on Bike Buyer:

| Predicted | 0 (Actual) | 1 (Actual) |
|---|---|---|
| 0 | 2118 | 921 |
| 1 | 718 | 1788 |

Counts for Target Mail Naive Bayes on Bike Buyer:

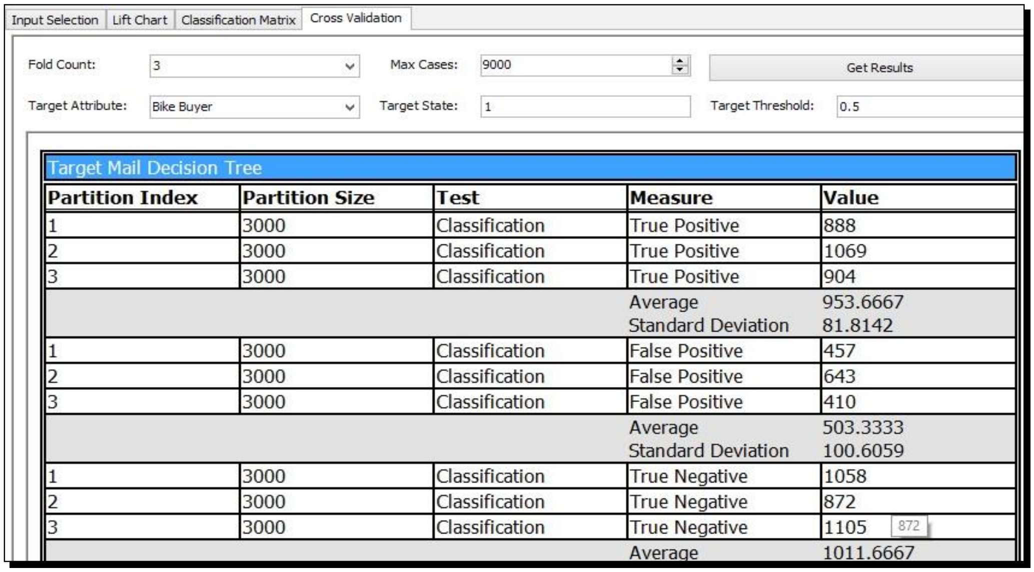
| Predicted | 0 (Actual) | 1 (Actual) |
|---|---|---|
| 0 | 1797 | 991 |
| 1 | 1039 | 1718 |

Counts for Target Mail Clustering on Bike Buyer:

| Predicted | 0 (Actual) | 1 (Actual) |
|---|---|---|
| 0 | 1803 | 1120 |
| 1 | 1033 | 1589 |

20. Go to the **Cross Validation** tab, set **Fold Count** to **3**, enter 9000 as **Max Cases**, leave **Target Attribute** as **Bike Buyer**, and set **Target State** as **1**. Next, enter 0.5 as **Target Threshold** and then click on **Get Results**. You will

see that 9,000 cases would be divided into three folds. A description of this result set can be found in the *What just happened?* section. The following screenshot shows the settings for the **Cross Validation** tab:



Fold Count: 3
Max Cases: 9000
Target Attribute: Bike Buyer
Target State: 1
Target Threshold: 0.5

Target Mail Decision Tree

| Partition Index | Partition Size | Test | Measure | Value |
|---|---|---|---|---|
| 1 | 3000 | Classification | True Positive | 888 |
| 2 | 3000 | Classification | True Positive | 1069 |
| 3 | 3000 | Classification | True Positive | 904 |
| | | | Average<br>Standard Deviation | 953.6667<br>81.8142 |
| 1 | 3000 | Classification | False Positive | 457 |
| 2 | 3000 | Classification | False Positive | 643 |
| 3 | 3000 | Classification | False Positive | 410 |
| | | | Average<br>Standard Deviation | 503.3333<br>100.6059 |
| 1 | 3000 | Classification | True Negative | 1058 |
| 2 | 3000 | Classification | True Negative | 872 |
| 3 | 3000 | Classification | True Negative | 1105 |
| | | | Average | 1011.6667 |

## WHAT JUST HAPPENED?

In this example, we've continued the scenario from the previous chapter to find out the best data mining model that generates better results for the problem (which was finding prospective bike buyers for the new product). The best mining model would be the one that generates results that are closest to the test dataset.

As a data mining developer, you are required to train multiple algorithms with the existing dataset. The reason for using multiple algorithms is that different mining algorithms generate patterns differently and give results differently. We will use multiple mining models to find the best results for the defined problem, so we need to find the best algorithm that produces the best results compared to the test dataset. After creating multiple algorithms and training them, you can use a set of mining accuracy charts to figure out which algorithm performs best compared to the real data in the test dataset. In this example, we've added two new mining models to the previous **Target Mail Mining Structure**: clustering and Naïve Bayes (steps 1 to 4).

## MICROSOFT CLUSTERING

In step 6, we added the Microsoft clustering algorithm to the mining structure. The clustering algorithm uses methods to categorize cases into clusters. The number of clusters can be changed with changing algorithm parameters. Each cluster will contain a number of cases. In step 7, you saw the first diagram of clustering, which is **Cluster Diagram**; this diagram shows the density of attributes in a cluster and shows the links between clusters, which show how close the clusters are to each other. For example, in step 7, we've shaded clusters after setting the **Bike Buyer** attribute to be **1**, which means the cluster with the darker blue shade has more cases with the **Bike Buyer** property as **1**.

Step 8 shows another viewer of the clustering algorithm, which is **Cluster Profiles**. **Cluster Profiles** shows the distribution of attributes and variables in each cluster compared to other clusters. For each attribute, there is a mining legend in the diagram that shows the definition of colors in the stacked column chart. For example, the screenshot of step 8 shows that most of the cases that **Cluster 3** contains has a commute distance of less than 1 mile.

The **Cluster Characteristic** tab, as you saw in step 9, shows the probability of each attribute's value in the selected cluster. This diagram is useful to understand the specific characteristic of each cluster. In step 10, the screenshot of **Cluster Discrimination** shows a comparison of two clusters based on the probability of attribute values. For example, when we compare **Cluster 1** (which is one of the clusters with customers who are most likely to buy bikes) with **Cluster 8** (which contains cases with a lesser number of bike buyers), you will see that most of the bike buyers do not own cars, and on the other hand most customers without bikes own two cars (as shown in the screenshot of step 10).

## MICROSOFT NAÏVE BAYES

The Microsoft Naïve Bayes algorithm performs the classification based on the Bayes algorithm. In step 11, we started to review the viewers for the Naïve Bayes mining model. The first viewer of the Naïve Bayes mining model is the **Dependency Network** tab, which we are familiar with from the previous chapter. This diagram shows the relationship of input algorithms and their contribution to leading to the `predictable` variable. In step 12, you can see that this algorithm identifies **Number Cars Owned**, **Total Children**, and **Commute Distance** as the main three variables that derive the bike buyers' prediction. One of the important considerations about mining algorithms is that each mining algorithm may return different results on the same dataset. If you compare the **Dependency Network** of step 12 of this example to the diagram of step 23 in the *Time for action – creating a data mining solution with the Microsoft Decision Tree* algorithm section of *Chapter 7*, *Data Mining – Descriptive Models in SSAS*, which was the **Dependency Network** of the decision tree algorithm, you will find that in the decision tree algorithm, the three main variables were **Number Cars of Owned**, **Region**, and **Yearly Income**.

> Different mining algorithms generate different results because they apply different mathematical calculations and methods on the dataset.

We've explored the **Attribute Profiles** tab in step 13. The **Attribute Profile** tab's diagram is very similar to the **Cluster Profile** tab's diagram. Both of these diagrams show the distribution of attributes in each classification. The only difference is that classifications in **Naïve Bayes Attribute Profile** are one for each value in the predictable variable. In this example, there were two clusters, one for **Bike Buyer** as **1** and another as **0**.

Step 14 shows the **Attribute Characteristics** diagram and step 15 shows the **Attribute Discrimination** diagram, which are similar to the **Cluster Characteristics** and **Cluster Discrimination** diagrams from Microsoft clustering mining viewers.

## MICROSOFT ACCURACY CHART

As you've learned so far, the pattern recognized by each mining model will be different from other mining models, even for the same input dataset. This is the nature of work for a data mining developer. As a mining developer, you need to apply different algorithms on the same dataset and then compare them to find out which algorithm provides the best results. The best results will be based on the pattern recognized by the algorithm (when you trained it with the training set) when applied on the test dataset. Microsoft Accuracy Chart is a set of diagrams and charts to evaluate how each algorithm performs against the test dataset and against other mining models. As a data mining developer, you need to assess the mining models against each other with Microsoft Accuracy Chart and find out the algorithm with the best results. This algorithm can be deployed to the live server for real prediction.

Microsoft Accuracy Chart contains a list of charts and diagrams such as Lift Chart and Profit Chart, Classification Matrix, and Cross Validation. In the **Input Selection** tab (step 16), we've set the predict value of the **`predictable`** variable (**Bike Buyer**) to **1**. It means that the mining models will predict customers who want to buy a bike. In this step, you can also configure the test dataset. By default, the test dataset would be the percentage of data that you've configured as test set in the mining model (in this case, 30 percent). But there is also an option for you to specify a test dataset in this tab directly.

### LIFT CHARTS AND PROFIT CHARTS

A Lift Chart shows how mining models act on prediction against each other when their patterns are applied on the test dataset. In the Lift Chart that you saw in step 17, there are five lines. There is a line for each mining model: one line for ideal prediction and another for random guesses (worst prediction). Each line in the Lift Chart shows how the overall population led to predict the target population. The best model, as is shown in the ideal mode, is to predict 100 percent of the target with only 50 percent of the overall population. And the worst model is to predict 100 percent of the target with 100 percent of the overall population. So the best mining model would be the one that is closest to the ideal mode. In the screenshot of step 17, you can see that the Target Mail decision tree performs better than the other two models.

The Profit Chart that was shown in step 18 shows the amount of profit calculated based on the usage of any of the mining models on the test dataset. You can configure the Profit Chart settings with changing values of **Population**, **Fixed Cost**, **Individual Cost**, and **Revenue per Individual**. The profit in our example can identify the most correct list of prospective buyers because it will reduce the need for sending e-mails to all customers.

### CLASSIFICATION MATRIX

**Classification Matrix** shows a matrix of prediction values against actual values for each mining model. In step 19, you saw the **Classification Matrix** tab, which shows the values **0** and **1** as row headers (for **Prediction**) and values **0** and **1** as column headers (for **Actual**). The values inside the matrix cells show the number of cases that were predicted against the actual values. For example, the screenshot of step 19 shows 1,788 cases predicted correctly (the result was predicted to be a bike buyer and he actually bought a bike). However, there are also 718 cases that were predicted to be bike buyers but who didn't actually buy a bike. By comparing the classification matrix of the three mining models, it is obvious that the decision tree performs better than other algorithms.

### CROSS VALIDATION

The last step of the example shows the **Cross Validation** report (step 20). **Cross Validation** performs mining algorithms on classifications of the input dataset (max cases) with each classification named as a fold (**Fold Count**). You can define the target attribute and its value in that. **Threshold** shows the level of confidence in prediction.

## PREDICTING DATA WITH DMX

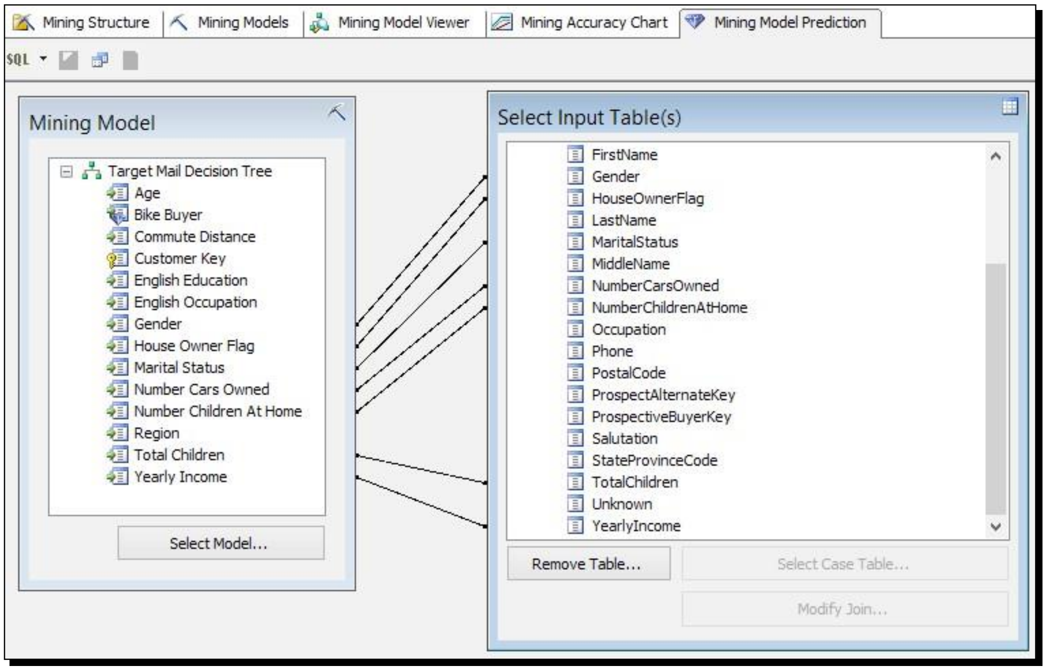
In this section, we will talk about the most attractive part of data mining, prediction. Microsoft provided a language to query and work with data mining algorithms and perform predictions: DMX. DMX is very similar to T-SQL in terms of the **`SELECT`**, **`FROM`**, and **`WHERE`** clauses. The process of prediction with mining algorithms is that the pattern fetched out of the training from a mining model will be joined to a case table. In other words, prediction is a result of applying this pattern with some DMX functions such as **`PredictProbability`**.

In this section, you will learn about the DMX language and how to use the **Mining Model Prediction** options to generate a prediction result set out of the mining model and case table.

## TIME FOR ACTION – PREDICTING PROSPECTIVE BIKE BUYERS

In this example, we will add a case table that contains only a list of new customers with attributes such as yearly income, total children, gender, and so on. Then we will apply DMX queries on the mining model and the case table to find out the list of prospective bike buyers. The prerequisite for running this example is the decision tree example mentioned in the previous chapter. Perform the following steps to predict prospective bike buyers:

1. Open the `Adventure Works DW2012.dsv` designer.
2. Right-click on a blank area in it and click on **Add/Remove Tables**. Select the **ProspectiveBuyer(dbo)** table and add it to the DSV.
3. Right-click on the **ProspectiveBuyer** table in the DSV and explore its data. You will see that most of the input variables of the Target mail mining algorithms are supported as columns of this table.
4. Save the DSV and open **Target Mail Mining Structure** and go to the **Mining Model Prediction** tab.
5. In the **Mining Model** pane, on the top left-hand side of this tab, click on **Select Model** and choose **Target Mail Decision Tree** under the **Target Mail Mining Structure** options.
6. In the **Select Case Table/Select Input** table pane (on the right-hand side), select the **ProspectiveBuyer** table. You will see that joins between the input variables of the mining model and case table will be generated based on the column names. The following screenshot shows the **Mining Model Prediction** tab:

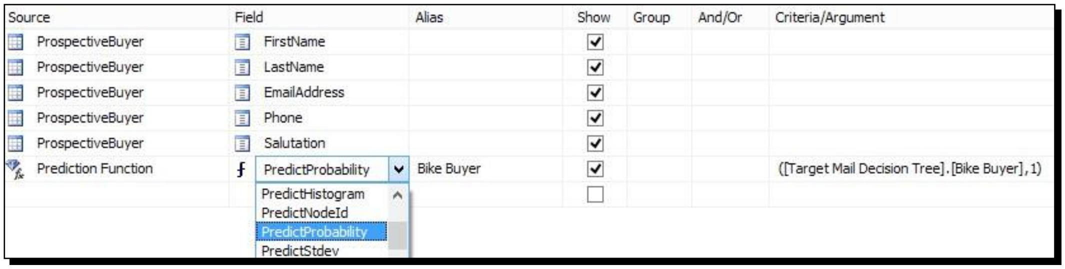


7. Drag-and-drop **FirstName**, **LastName**, **EmailAddress**, **Phone**, and **Salutation** from the case table to the grid under it.

8. In the last record of the grid, add a new row by choosing the **Source** column as **Prediction Function**. Then choose the **PredictProbability** function in the **Field** column. Set **Alias** as **Bike Buyer** and enter the following expression in **Criteria** column (you can also use drag-and-drop to facilitate the writing of the expression):

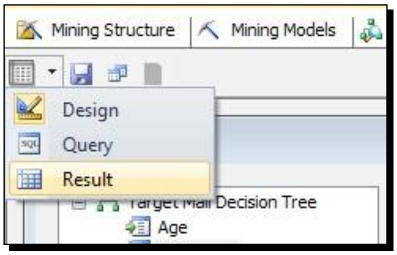
```
([Target Mail Decision tree].[Bike Buyer],1)
```

Take a look at the screenshot of a record of the grid:

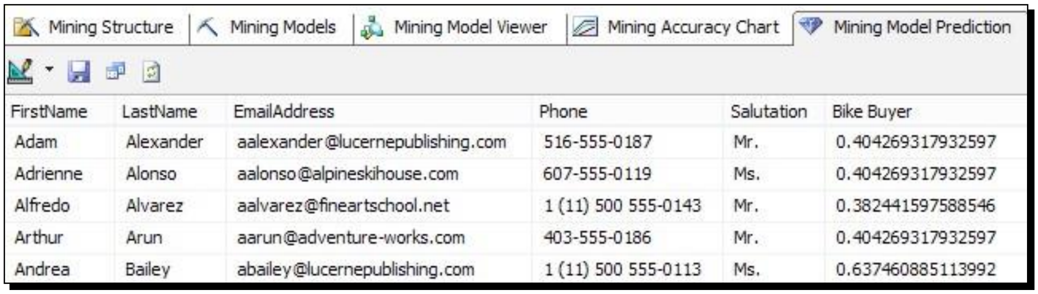


| Source | Field | Alias | Show | Group | And/Or | Criteria/Argument |
|---|---|---|---|---|---|---|
| ProspectiveBuyer | FirstName | | ☑ | | | |
| ProspectiveBuyer | LastName | | ☑ | | | |
| ProspectiveBuyer | EmailAddress | | ☑ | | | |
| ProspectiveBuyer | Phone | | ☑ | | | |
| ProspectiveBuyer | Salutation | | ☑ | | | |
| Prediction Function | PredictProbability | Bike Buyer | ☑ | | | ([Target Mail Decision Tree].[Bike Buyer],1) |
| | | | □ | | | |

9. Click on the **Mode change** icon. The icon is located on the top left-hand side of the **Mining Model Prediction** tab. Select **Result** in the drop-down menu, as shown in the following screenshot:



10. You will see the data mining query execution result. The result set shows a list of bike buyers with the probability of buying bikes. The result set is not filtered or ordered. So it shows even customers with less than 0.5 probability of buying a bike, as shown in the following screenshot:



| FirstName | LastName | EmailAddress | Phone | Salutation | Bike Buyer |
|---|---|---|---|---|---|
| Adam | Alexander | aalexander@lucernepublishing.com | 516-555-0187 | Mr. | 0.404269317932597 |
| Adrienne | Alonso | aalonso@alpineskihouse.com | 607-555-0119 | Ms. | 0.404269317932597 |
| Alfredo | Alvarez | aalvarez@fineartschool.net | 1 (11) 500 555-0143 | Mr. | 0.382441597588546 |
| Arthur | Arun | aarun@adventure-works.com | 403-555-0186 | Mr. | 0.404269317932597 |
| Andrea | Bailey | abailey@lucernepublishing.com | 1 (11) 500 555-0113 | Ms. | 0.637460885113992 |

11. Click on the **Mode change** icon and change the mode to **Query**. You will see the SQL-like query. This is the DMX query generated based on the configuration you've performed in the design mode.

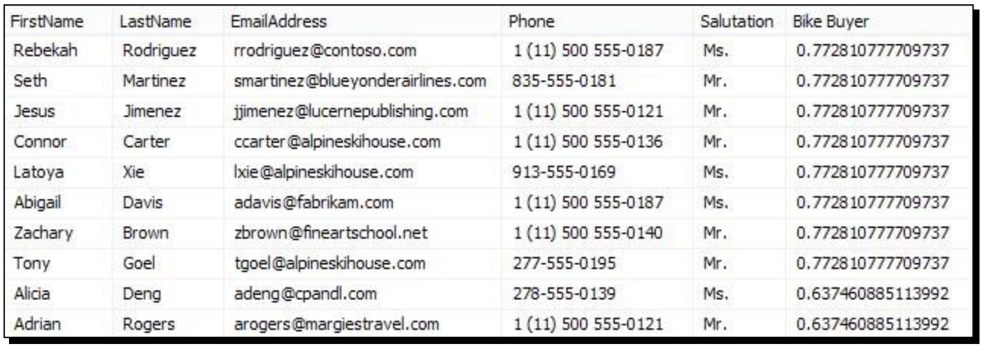
12. Add the following expression (`where` clause and `order by`) at the end of DMX query:

```
where PredictProbability([Target Mail Decision Tree].
  [Bike Buyer],1)>0.5 order by PredictProbability([Target Mail Decision
Tree].
  [Bike Buyer],1) desc
```

**13.** The whole DMX query should look like the following code:

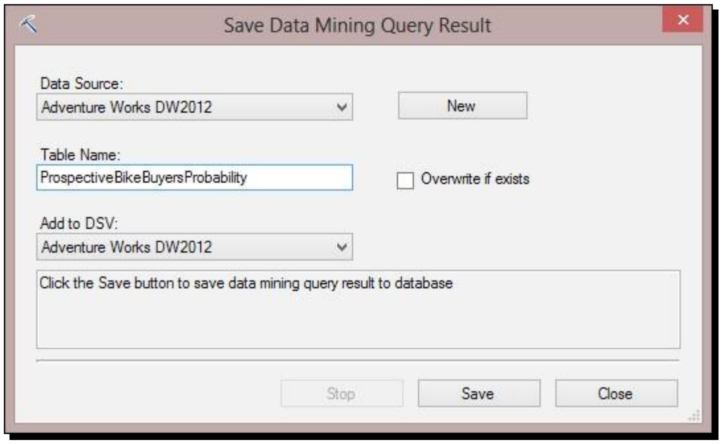
```
SELECT
  t.[FirstName],
  t.[LastName],
  t.[EmailAddress],
  t.[Phone],
  t.[Salutation],
  (PredictProbability([Target Mail Decision Tree].[Bike Buyer],1)) as [Bike Buyer]
From
  [Target Mail Decision Tree]
PREDICTION JOIN
  OPENQUERY([Adventure Works DW2012],
    'SELECT
      [FirstName],
      [LastName],
      [EmailAddress],
      [Phone],
      [Salutation],
      [MaritalStatus],
      [Gender],
      [YearlyIncome],
      [TotalChildren],
      [NumberChildrenAtHome],
      [HouseOwnerFlag],
      [NumberCarsOwned]
    FROM
      [dbo].[ProspectiveBuyer]
    ') AS t
ON
  [Target Mail Decision Tree].[Marital Status] = t.
  [MaritalStatus] AND
  [Target Mail Decision Tree].[Gender] = t.[Gender] AND
  [Target Mail Decision Tree].[Yearly Income] = t.
  [YearlyIncome] AND
  [Target Mail Decision Tree].[Total Children] = t.
  [TotalChildren] AND
  [Target Mail Decision Tree].[Number Children At Home] = t.
[NumberChildrenAtHome] AND
  [Target Mail Decision Tree].[House Owner Flag] = t.
  [HouseOwnerFlag] AND
  [Target Mail Decision Tree].[Number Cars Owned] = t.
  [NumberCarsOwned]
where PredictProbability([Target Mail Decision Tree].[Bike
Buyer],1)>0.5
order by PredictProbability([Target Mail Decision Tree].[Bike Buyer],1) desc
```

**14.** Go to the **Result** mode. You will see a list of prospective bike buyers filtered and sorted this time. The higher the value in the bike buyer columns, the higher the probability of that customer buying a bike, as shown in the following screenshot:



| FirstName | LastName | EmailAddress | Phone | Salutation | Bike Buyer |
|---|---|---|---|---|---|
| Rebekah | Rodriguez | rrodriguez@contoso.com | 1 (11) 500 555-0187 | Ms. | 0.772810777709737 |
| Seth | Martinez | smartinez@blueyonderairlines.com | 835-555-0181 | Mr. | 0.772810777709737 |
| Jesus | Jimenez | jjimenez@lucernepublishing.com | 1 (11) 500 555-0121 | Mr. | 0.772810777709737 |
| Connor | Carter | ccarter@alpineskihouse.com | 1 (11) 500 555-0136 | Mr. | 0.772810777709737 |
| Latoya | Xie | lxie@alpineskihouse.com | 913-555-0169 | Ms. | 0.772810777709737 |
| Abigail | Davis | adavis@fabrikam.com | 1 (11) 500 555-0187 | Ms. | 0.772810777709737 |
| Zachary | Brown | zbrown@fineartschool.net | 1 (11) 500 555-0140 | Mr. | 0.772810777709737 |
| Tony | Goel | tgoel@alpineskihouse.com | 277-555-0195 | Mr. | 0.772810777709737 |
| Alicia | Deng | adeng@cpandl.com | 278-555-0139 | Ms. | 0.637460885113992 |
| Adrian | Rogers | arogers@margiestravel.com | 1 (11) 500 555-0121 | Mr. | 0.637460885113992 |

**15.** Click on the **Save** icon on the right-hand side of the **Mode change** icon and save the result set to a new table named **`ProspectiveBikeBuyersProbability`** under the same data source and data source view, as shown in the following screenshot:



Data Source: Adventure Works DW2012

Table Name: ProspectiveBikeBuyersProbability

☐ Overwrite if exists

Add to DSV: Adventure Works DW2012

Click the Save button to save data mining query result to database

**16.** If you go back to the DSV designer, you will see the new table created there. You can explore the data of that table and view the result of the data mining query there.

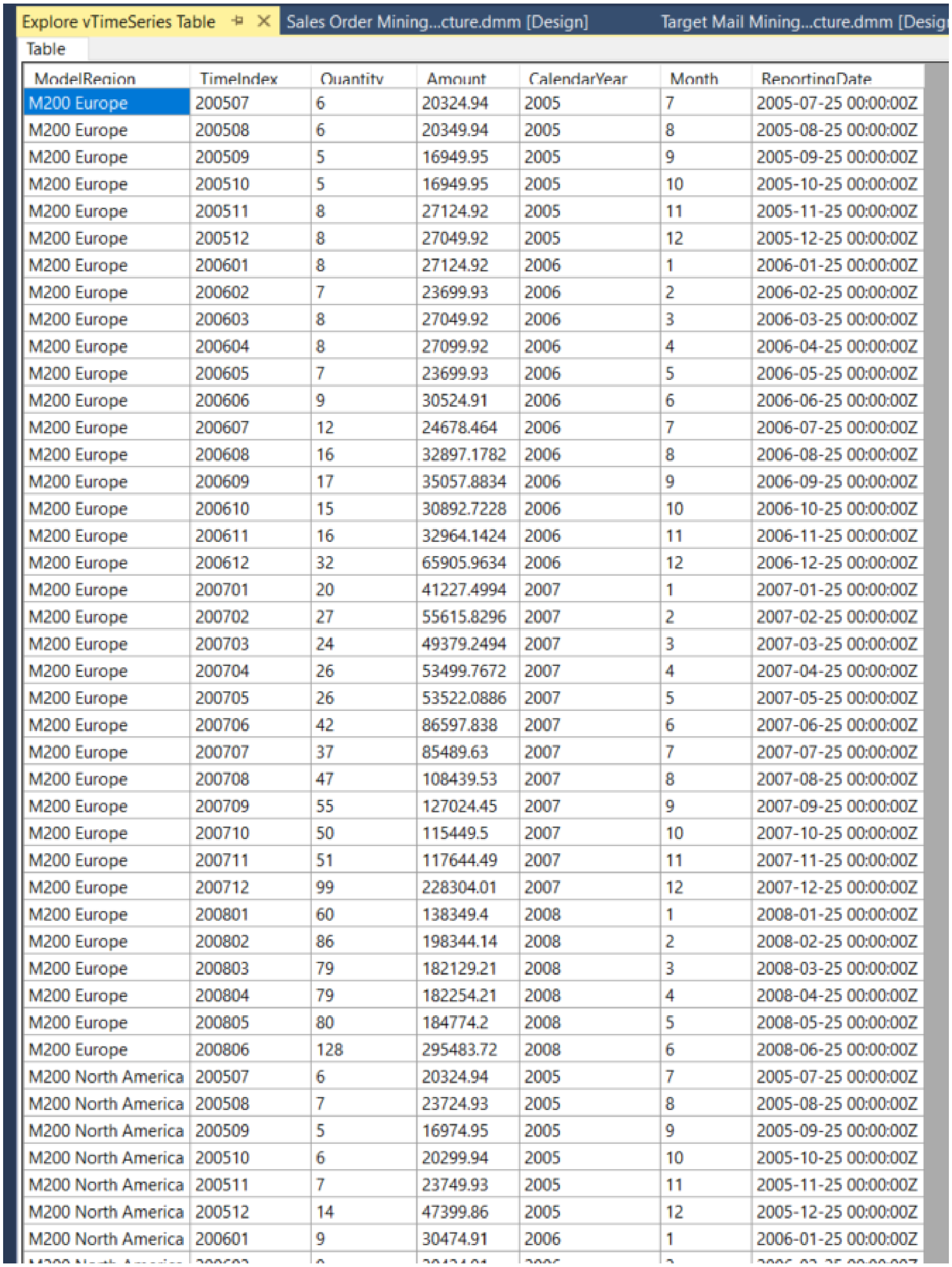
## MICROSOFT TIME SERIES

In this last section of the data mining discussion, we will reveal the Time Series mining model. The Time Series algorithm is very useful to predict time-based facts such as prediction of future sales of a product based on the information of the last two years, or more.

## TIME FOR ACTION – PREDICTING FUTURE SALES WITH MICROSOFT TIME SERIES

In this example, we use a dataset to train a Time Series algorithm with sales information of previous years and then we will see how a Time Series algorithm will help to predict future sales based on the trained pattern.

1. Open `Adventure Works DW2012.dsv` and add a new table to it. Choose **vTimeSeries** from the list of tables.
2. Right-click on the **vTimeSeries** table in DSV and explore the data. You will see that this table contains the sales amount and quantity based on the product model and region and month.



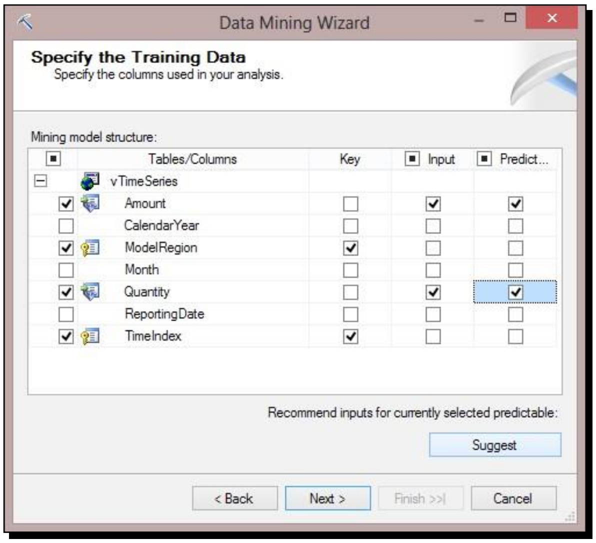
Explore vTimeSeries Table
Sales Order Mining...cture.dmm [Design]
Target Mail Mining...cture.dmm [Desig
Table

| ModelRegion | TimeIndex | Quantity | Amount | CalendarYear | Month | ReportingDate |
|---|---|---|---|---|---|---|
| M200 Europe | 200507 | 6 | 20324.94 | 2005 | 7 | 2005-07-25 00:00:00Z |
| M200 Europe | 200508 | 6 | 20349.94 | 2005 | 8 | 2005-08-25 00:00:00Z |
| M200 Europe | 200509 | 5 | 16949.95 | 2005 | 9 | 2005-09-25 00:00:00Z |
| M200 Europe | 200510 | 5 | 16949.95 | 2005 | 10 | 2005-10-25 00:00:00Z |
| M200 Europe | 200511 | 8 | 27124.92 | 2005 | 11 | 2005-11-25 00:00:00Z |
| M200 Europe | 200512 | 8 | 27049.92 | 2005 | 12 | 2005-12-25 00:00:00Z |
| M200 Europe | 200601 | 8 | 27124.92 | 2006 | 1 | 2006-01-25 00:00:00Z |
| M200 Europe | 200602 | 7 | 23699.93 | 2006 | 2 | 2006-02-25 00:00:00Z |
| M200 Europe | 200603 | 8 | 27049.92 | 2006 | 3 | 2006-03-25 00:00:00Z |
| M200 Europe | 200604 | 8 | 27099.92 | 2006 | 4 | 2006-04-25 00:00:00Z |
| M200 Europe | 200605 | 7 | 23699.93 | 2006 | 5 | 2006-05-25 00:00:00Z |
| M200 Europe | 200606 | 9 | 30524.91 | 2006 | 6 | 2006-06-25 00:00:00Z |
| M200 Europe | 200607 | 12 | 24678.464 | 2006 | 7 | 2006-07-25 00:00:00Z |
| M200 Europe | 200608 | 16 | 32897.1782 | 2006 | 8 | 2006-08-25 00:00:00Z |
| M200 Europe | 200609 | 17 | 35057.8834 | 2006 | 9 | 2006-09-25 00:00:00Z |
| M200 Europe | 200610 | 15 | 30892.7228 | 2006 | 10 | 2006-10-25 00:00:00Z |
| M200 Europe | 200611 | 16 | 32964.1424 | 2006 | 11 | 2006-11-25 00:00:00Z |
| M200 Europe | 200612 | 32 | 65905.9634 | 2006 | 12 | 2006-12-25 00:00:00Z |
| M200 Europe | 200701 | 20 | 41227.4994 | 2007 | 1 | 2007-01-25 00:00:00Z |
| M200 Europe | 200702 | 27 | 55615.8296 | 2007 | 2 | 2007-02-25 00:00:00Z |
| M200 Europe | 200703 | 24 | 49379.2494 | 2007 | 3 | 2007-03-25 00:00:00Z |
| M200 Europe | 200704 | 26 | 53499.7672 | 2007 | 4 | 2007-04-25 00:00:00Z |
| M200 Europe | 200705 | 26 | 53522.0886 | 2007 | 5 | 2007-05-25 00:00:00Z |
| M200 Europe | 200706 | 42 | 86597.838 | 2007 | 6 | 2007-06-25 00:00:00Z |
| M200 Europe | 200707 | 37 | 85489.63 | 2007 | 7 | 2007-07-25 00:00:00Z |
| M200 Europe | 200708 | 47 | 108439.53 | 2007 | 8 | 2007-08-25 00:00:00Z |
| M200 Europe | 200709 | 55 | 127024.45 | 2007 | 9 | 2007-09-25 00:00:00Z |
| M200 Europe | 200710 | 50 | 115449.5 | 2007 | 10 | 2007-10-25 00:00:00Z |
| M200 Europe | 200711 | 51 | 117644.49 | 2007 | 11 | 2007-11-25 00:00:00Z |
| M200 Europe | 200712 | 99 | 228304.01 | 2007 | 12 | 2007-12-25 00:00:00Z |
| M200 Europe | 200801 | 60 | 138349.4 | 2008 | 1 | 2008-01-25 00:00:00Z |
| M200 Europe | 200802 | 86 | 198344.14 | 2008 | 2 | 2008-02-25 00:00:00Z |
| M200 Europe | 200803 | 79 | 182129.21 | 2008 | 3 | 2008-03-25 00:00:00Z |
| M200 Europe | 200804 | 79 | 182254.21 | 2008 | 4 | 2008-04-25 00:00:00Z |
| M200 Europe | 200805 | 80 | 184774.2 | 2008 | 5 | 2008-05-25 00:00:00Z |
| M200 Europe | 200806 | 128 | 295483.72 | 2008 | 6 | 2008-06-25 00:00:00Z |
| M200 North America | 200507 | 6 | 20324.94 | 2005 | 7 | 2005-07-25 00:00:00Z |
| M200 North America | 200508 | 7 | 23724.93 | 2005 | 8 | 2005-08-25 00:00:00Z |
| M200 North America | 200509 | 5 | 16974.95 | 2005 | 9 | 2005-09-25 00:00:00Z |
| M200 North America | 200510 | 6 | 20299.94 | 2005 | 10 | 2005-10-25 00:00:00Z |
| M200 North America | 200511 | 7 | 23749.93 | 2005 | 11 | 2005-11-25 00:00:00Z |
| M200 North America | 200512 | 14 | 47399.86 | 2005 | 12 | 2005-12-25 00:00:00Z |
| M200 North America | 200601 | 9 | 30474.91 | 2006 | 1 | 2006-01-25 00:00:00Z |

3. In the DSV designer, select both the **TimeIndex** and **ModelRegion** columns from **vTimeSeries** and right-click and choose **Set Logical Primary Key**.

4. Create a new mining structure from the existing relational database and choose **Microsoft Time Series** as the mining model. Set the **vTimeSeries** table as **Case table**.

5. In the **Specify the Training Data** window, set **TimeIndex** and **ModelRegion** as the **Key** columns and set **Amount** and **Quantity** as the **Input** and **Predictable** columns, as shown in the following screenshot:



Data Mining Wizard

**Specify the Training Data**
Specify the columns used in your analysis.

Mining model structure:

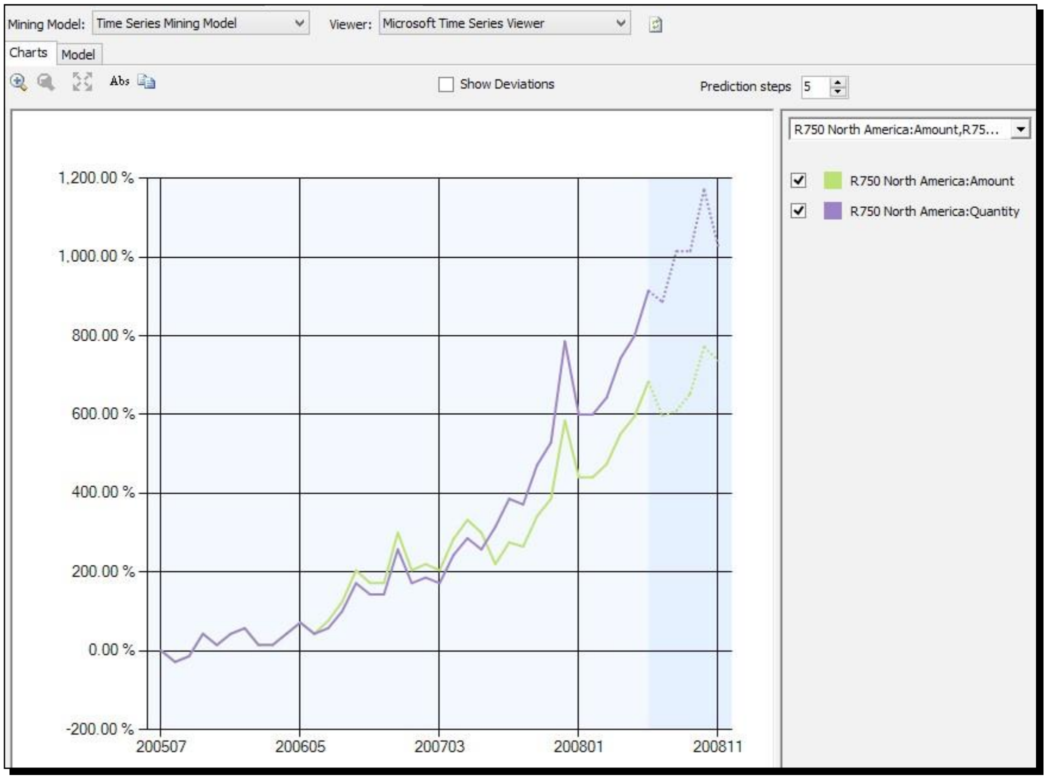
| ■ | Tables/Columns | Key | ■ Input | ■ Predict... |
|---|---|---|---|---|
| ⊟ | vTimeSeries | | | |
| ☑ | Amount | □ | ☑ | ☑ |
| □ | CalendarYear | □ | □ | □ |
| ☑ | ModelRegion | ☑ | □ | □ |
| □ | Month | □ | □ | □ |
| ☑ | Quantity | □ | ☑ | ☑ |
| □ | ReportingDate | □ | □ | □ |
| ☑ | TimeIndex | ☑ | □ | □ |

Recommend inputs for currently selected predictable:

**6.** Detect the data types in the next step and name the mining structure `Time Series Mining Structure` and the model `Time Series Mining Model`.

**7.** Deploy and process the mining model.

**8.** Go to **Mining Model Viewer**. You will see a line chart in the **Charts** tab of **Microsoft Time Series Viewer**. This diagram shows the changing of variables over time. Choose **R 750 North America: Amount** and **R 750 North America: Quantity** from the dropdown list. You can also reduce or increase prediction steps to see how the diagram changes. The following screenshot shows how the R 750 product will sell (amount and quantity) in the next five months:



Mining Model: Time Series Mining Model
Viewer: Microsoft Time Series Viewer

□ Show Deviations
Prediction steps 5

R750 North America:Amount,R75...

☑ 🟩 R750 North America:Amount
☑ 🟪 R750 North America:Quantity

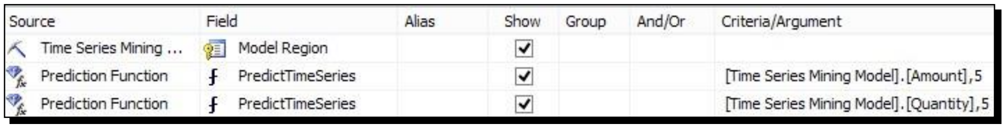
**9.** Go to the **Mining Model Prediction** tab and verify that the model is **Time Series Mining Model**.

**10.** Drag-and-drop **ModelRegion** from model to the grid as shown in the next screenshot.

**11.** Add a **Prediction Function** named **PredictTimeSeries** with the criteria shown in the following line of code:

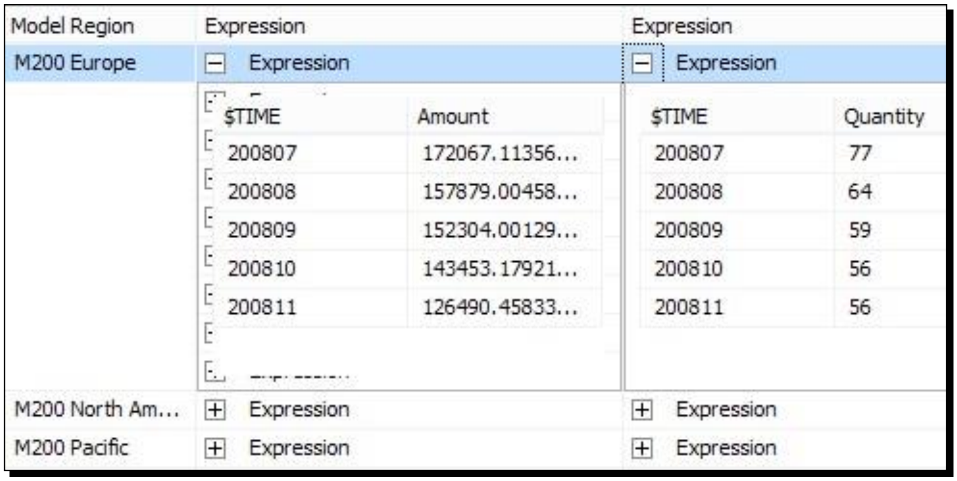
```
[Time Series Mining Model].[Amount],5
```

**12.** Add another **PredictTimeSeries** function for **Quantity**. The following screenshot shows how the functions are to be configured:



| Source | Field | Alias | Show | Group | And/Or | Criteria/Argument |
|---|---|---|---|---|---|---|
| Time Series Mining ... | Model Region | | ☑ | | | |
| Prediction Function | PredictTimeSeries | | ☑ | | | [Time Series Mining Model].[Amount],5 |
| Prediction Function | PredictTimeSeries | | ☑ | | | [Time Series Mining Model].[Quantity],5 |

**13.** Go to the **Result** mode. You will see the result set with two hierarchical columns (one for each prediction result). If you expand the **Expression** columns, you will see a prediction for the **Amount** (second column) and **Quantity** (third column) for the next five steps (months), as shown in the following screenshot:



| Model Region | Expression | Expression |
|---|---|---|
| M200 Europe | ⊟ Expression | ⊟ Expression |
| M200 North Am... | ⊞ Expression | ⊞ Expression |
| M200 Pacific | ⊞ Expression | ⊞ Expression |

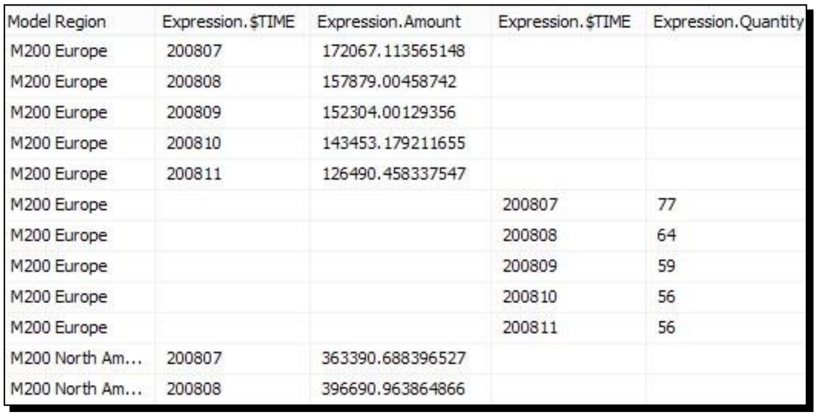
| $TIME | Amount |
|---|---|
| 200807 | 172067.11356... |
| 200808 | 157879.00458... |
| 200809 | 152304.00129... |
| 200810 | 143453.17921... |
| 200811 | 126490.45833... |

| $TIME | Quantity |
|---|---|
| 200807 | 77 |
| 200808 | 64 |
| 200809 | 59 |
| 200810 | 56 |
| 200811 | 56 |

**14.** Go to the **Query** mode and add the `flattened` keyword after `SELECT` and before the first column of the query, as shown in the following code snippet:

```
SELECT flattened
  [Time Series Mining Model].[Model Region],
  PredictTimeSeries([Time Series Mining Model].[Amount],5),
  PredictTimeSeries([Time Series Mining Model].[Quantity],5)
From
  [Time Series Mining Model]
```

**15.** Go to the **Result** mode. This time, you will see that the **Expression** column's values are flattened into columns and rows. This result is more useful when you want to export the result set into a table. The following screenshot shows the results:



| Model Region | Expression.$TIME | Expression.Amount | Expression.$TIME | Expression.Quantity |
|---|---|---|---|---|
| M200 Europe | 200807 | 172067.113565148 | | |
| M200 Europe | 200808 | 157879.00458742 | | |
| M200 Europe | 200809 | 152304.00129356 | | |
| M200 Europe | 200810 | 143453.179211655 | | |
| M200 Europe | 200811 | 126490.458337547 | | |
| M200 Europe | | | 200807 | 77 |
| M200 Europe | | | 200808 | 64 |
| M200 Europe | | | 200809 | 59 |
| M200 Europe | | | 200810 | 56 |
| M200 Europe | | | 200811 | 56 |
| M200 North Am... | 200807 | 363390.688396527 | | |
| M200 North Am... | 200808 | 396690.963864866 | | |

## CONCLUSION

Through the hands-on projects of this assignment, we were able to come to a deeper understanding of how we can leverage the capabilities of data mining to benefit a business' continued success. When we were first learning about the concepts of the different data mining models and how they can be applied to help us predict data, I was a bit worried that creating data mining models would be a very tedious task. This is because data mining models are really incredible and must hold a high level of complexity in order to provide us with predictions based on the training data. However, after going through this lab, I have discovered that it is a relatively easy and straightforward process to develop data mining models. This is thanks to all of the wonderful work that has been put into developing the SSDT (SQL Server Development Tools), and Visual Studio. Following the instructions provided in the assignment document, not only were we able to easily create the three data mining models (Decision Tree, Naïve Bayes, and Clustering), but we were able to create other data mining structures such as Association Rules, and we were able to use Lift Charts, Profit Charts and Classification Matrices to help us compare the effectiveness and accuracy of the various mining models to select the best data mining model for the data at hand. With the vast amount of ever-growing data we generate every day, data mining plays a more and more important rule. We need to be identified what attributes in data are relevant, and proactively identify trends in the data in order to make well informed decisions based on accurate predictions.  From this assignment, we were able to successfully generate three data mining models to help us predict the number of potential customers would purchase bikes from our company. Furthermore, we were able to compare the costs and profits we would be able to save and generate between the different prediction accuracies provided by these data mining models. From this gained knowledge, we were able to identify that the Decision Tree model would be best suited to be deployed as the data mining model for the given dataset and scenario.