SQL Server 2012 Tutorials: Analysis Services - Data Mining

SQL Server 2012 Books Online

Step-by-Step

Microsoft

SQL Server 2012 Tutorials: Analysis Services - Data Mining

SQL Server 2012 Books Online

Summary: Microsoft SQL Server Analysis Services makes it easy to create sophisticated data mining solutions. The step-by-step tutorials in the following list will help you learn how to get the most out of Analysis Services, so that you can perform advanced analysis to solve business problems that are beyond the reach of traditional business intelligence methods.

Category: Step-by-Step
Applies to: SQL Server 2012
Source: SQL Server Books Online (link to source content)
E-book publication date: June 2012

Contents

Data Mining Tutorials (Analysis Services)

Microsoft SQL Server Analysis Services makes it easy to create sophisticated data mining solutions. The tools in Analysis Services help you design, create, and manage data mining models that use either relational or cube data. You can manage client access to data mining models and create prediction queries from multiple clients.

The step-by-step tutorials in the following list will help you learn how to get the most out of Analysis Services, so that you can perform advanced analysis to solve business problems that are beyond the reach of traditional business intelligence methods.

In this Section

- Basic Data Mining Tutorial

  This tutorial walks you through a targeted mailing scenario. It demonstrates how to use the data mining algorithms, mining model viewers, and data mining tools that are included in Analysis Services. You will build three data mining models to answer practical business questions while learning data mining concepts and tools.

- Intermediate Data Mining Tutorial (Analysis Services - Data Mining)

  This tutorial contains a collection of lessons that introduce more advanced data mining concepts and techniques. The scenarios include these model types:

  - forecasting
  - market basket analysis
  - neural networks and logistic regression
  - sequence clustering

  The lessons are independent and can be done in any order, but you should have a basic knowledge of how to build data sources.

  Advanced concepts covered in these lessons include the use of nested tables, cross-prediction, custom data source views and named queries, and filtering in data mining queries. You will also gain proficiency in using the prediction query tools that are included in Analysis Services.

Reference

Data Mining Algorithms (Analysis Services - Data Mining)

Data Mining Extensions (DMX) Reference

Related Sections

Using the Data Mining Tools

Logical Architecture (Analysis Services - Data Mining)

Logical Architecture (Analysis Services - Multidimensional Data)

Designing and Implementing (Analysis Services - Data Mining)

See Also

Working with Data Mining

Microsoft SQL Server Data Mining resources

Creating and Querying Data Mining Models with DMX: Tutorials (Analysis Services - Data Mining)

Basic Data Mining Tutorial

Welcome to the Microsoft Analysis Services Basic Data Mining Tutorial. Microsoft SQL Server provides an integrated environment for creating and working with data mining models. In this tutorial, you will complete a scenario for a targeted mailing campaign in which you create models for analyzing and predicting customer purchasing behavior and for targeting potential buyers. The tutorial demonstrates how to use three of the most important data mining algorithms, how to analyze your findings using the mining model viewers, create predictions and accuracy charts, using the data mining tools that are included in Microsoft SQL Server Analysis Services. The fictitious company, Adventure Works Cycles, is used for all examples.

When you are comfortable using the data mining tools, we recommend that you also complete the Intermediate Data Mining Tutorial, which demonstrates how to use forecasting, market basket analysis, time series, association models, nested tables, and sequence clustering.

Tutorial Scenario

In this tutorial, you are an employee of Adventure Works Cycles who has been tasked with learning more about the company's customers based on historical purchases, and then using that historical data to make predictions that can be used in marketing. The company has never done data mining before, so you must create a new database specifically for data mining and set up several data mining models.

What You Will Learn

This tutorial teaches you how to create and work with several different types of data mining models. It also teaches you how to create a copy of a mining model, and apply a filter to the mining model. You then process the new model and evaluate the model using a lift chart. After the model is complete, you use drillthrough to retrieve additional data from the underlying mining structure.

Microsoft Analysis Services Data Mining includes the following features that help you easily develop and compare multiple predictive models and then take actions on the results :

- *Holdout Test Sets* - When you create a mining structure, you can now divide the data in the mining structure into training and testing sets. This lets you test models on similar data sets, and compare the accuracy of related models.
- *Mining model filters* - You can now attach filters to a mining model, and apply the filter during both training and testing. This lets you easily build related models on different subsets of the data.
- *Drillthrough to Structure Cases and Structure Columns* - You can now easily move from the general patterns in the mining model to actionable detail in the data source.

This tutorial is divided into the following lessons:

Lesson 1: Preparing the Analysis Services Database

In this lesson, you will learn how to create a new Analysis Services database, add a data source and data source view, and prepare the new database to be used with data mining.

Lesson 2: Building the Targeted Mailing Scenario

In this lesson, you will learn how to create a mining model structure that can be used as part of a targeted mailing scenario.

Lesson 3: Adding and Processing Models

In this lesson you will learn how to add models to a structure. The models you create are built with the following algorithms:

- Microsoft Decision Trees
- Microsoft Clustering
- Microsoft Naive Bayes

Lesson 4: Exploring the Targeted Mailing Models (Basic Data Mining Tutorial)

In this lesson you will learn how to explore and interpret the findings of each model using the Viewers.

Lesson 5: Testing Models (Basic Data Mining Tutorial)

In this lesson, you make a copy of one of the targeted mailing models, add a mining model filter to restrict the training data to a particular set of customers, and then assess the viability of the model.

Lesson 6: Creating and Working with Predictions (Basic Data Mining Tutorial)

In this final lesson of the Basic Data Mining Tutorial, you use the model to predict which

customers are most likely to purchase a bike. You then drill through to the underlying cases to obtain contact information.

Requirements

Make sure that the following are installed:

- Microsoft SQL Server 2012
- Microsoft SQL Server Analysis Services in multidimensional mode
- The database.

To enhance security, the sample databases are not installed with SQL Server. To install the official databases for Microsoft SQL Server, visit the Microsoft SQL Sample Databases page and select SQL Server 2012.

Note
When you are working through a tutorial, you might find it easier to move back and forth between the steps if you add the **Next topic** and **Previous topic** buttons to the document viewer toolbar. For more information, see Adding Next and Previous Buttons to Help.

See Also

Working with Data Mining

Mining Models Tab: How-to Topics

Creating and Querying Data Mining Models with DMX: Tutorials (Analysis Services - Data Mining)

Lesson 1: Preparing the Analysis Services Database (Basic Data Mining Tutorial)

You are a new employee of Adventure Works Cycles who has been tasked with designing a business intelligence application in SQL Server 2012. Adventure Works Cycles hopes to leverage your Analysis Services data mining experience to discover interesting and actionable information about people who have purchased bicycles. They then want you to predict which prospective customers are most likely to purchase a bicycle in the future.

Designing this application in SQL Server starts with the creation in SQL Server Data Tools (SSDT) of a SQL Server Analysis Services project based on the Analysis Services project template for multidimensional modeling and data mining. After you create an Analysis Services project, you define one or more data sources. Then, you define a view of the metadata, called a *data source view*, from selected tables and views from the data sources.

In this lesson, you will create an Analysis Services project, define a single data source, and add a subset of tables to a data source view. This lesson includes the following tasks:

Creating an Analysis Services Project (Basic Data Mining Tutorial)

Creating a Data Source (Basic Data Mining Tutorial)

Creating a Data Source View (Basic Data Mining Tutorial)

First Task in Lesson

Creating an Analysis Services Project (Basic Data Mining Tutorial)

Next Lesson

Lesson 2: Building a Targeted Mailing Scenario (Basic Data Mining)

See Also

Designing Data Source Views (Analysis Services)

Defining Data Sources (Analysis Services)

Building Analysis Services Projects

Creating an Analysis Services Project

Creating an Analysis Services Project (Basic Data Mining Tutorial)

Each Microsoft SQL Server Analysis Services project defines the schema for the objects in a single Analysis Services database. An Analysis Services database contains mining structures and mining models, multidimensional models (cubes), and supporting objects such as data sources and data source views. In this tutorial you will be using the database as a data source. You will deploy the data mining objects to an Analysis Services database named **BasicDataMining**.

By default, Analysis Services uses the **localhost** instance for new projects. If you are using a named instance or a different server, you must first create and open the project and then change the instance name.

For more information about Analysis Services projects, see Creating an Analysis Services Project.

Procedures

To create an Analysis Services project

1. Open SQL Server Data Tools (SSDT).
2. On the **File** menu, point to **New**, and then select **Project**.
3. Verify that **Business Intelligence Projects** is selected in the **Project types** pane.
4. In the **Templates** pane, select **Analysis Services Multidimensional and Data Mining Project**.
5. In the **Name** box, name the new project **BasicDataMining**.

6. Click .

To change the instance where data mining objects are stored

1. In SQL Server Data Tools (SSDT), on the **Project** menu, select **Properties**.
2. On the left side of the **Property Pages** pane, under **Configuration Properties**, click **Deployment**.
3. On the right side of the **Property Pages** pane, under **Target**, verify that the **Server** name is **localhost**. If you are using a different instance, type the name of the instance. Click .

Next Task in Lesson

Creating a Data Source (Data Mining Tutorial)

See Also

Building Analysis Services Projects

Defining an Analysis Services Project

How to: Build and Deploy an Analysis Services Project

Creating a Data Source (Basic Data Mining Tutorial)

A *data source* is a data connection that is saved and managed in your project and deployed to your Microsoft SQL Server Analysis Services database. The data source contains the names of the server and database where your source data resides, in addition to any other required connection properties.

Important
The name of the database is . If you have not already installed this database, see the Microsoft SQL Sample Databases page.

Procedures

To create a data source

1. In **Solution Explorer**, right-click the **Data Sources** folder and select **New Data Source**.
2. On the **Welcome to the Data Source Wizard** page, click **Next**.
3. On the **Select how to define the connection** page, click **New** to add a connection to the database.
4. In the **Provider** list in **Connection Manager**, select **Native OLE DB\SQL Server Native Client 11.0**.
5. In the **Server name** box, type or select the name of the server on which you installed .

   For example, type **localhost** if the database is hosted on the local server.

6. In the **Log onto the server** group, select **Use Windows Authentication**.

   Important
   Whenever possible, implementers should use Windows Authentication, as it provides a more secure authentication method than SQL Server Authentication. However, SQL Server Authentication is provided for backward compatibility. For more information about authentication methods, see Database Engine Configuration - Account Provisioning.

7. In the **Select or enter a database name** list, select and then click **OK**.
8. Click **Next**.
9. On the **Impersonation Information** page, click **Use the service account**, and then click **Next**.

   On the **Completing the Wizard** page, notice that, by default, the data source is named Adventure Works DW 2012.

10. Click **Finish**.

    The new data source, Adventure Works DW 2012, appears in the **Data Sources** folder in Solution Explorer.

Next Task in Lesson

Creating a Data Source View (Data Mining Tutorial)

Previous Task in Lesson

Creating an Analysis Services Project (Basic Data Mining Tutorial)

See Also

Defining a Data Source Using the Data Source Wizard (Analysis Services)

Creating Data Sources How-to Topics

Defining a Data Source

Impersonation Information Dialog Box (Analysis Services - Multidimensional Data)

Creating a Data Source View (Basic Data Mining Tutorial)

A data source view is built on a data source and defines a subset of the data, which you can then use in your mining structures. You can also use the data source view to add columns, create calculated columns and aggregates, and add named views. By using data source views, you can select the data that relates to your project, establish relationships between tables, and modify the structure of the data, without modifying the original data source. For more information, see Designing Data Source Views (Analysis Services).

Procedures

To create a data source view

1. In **Solution Explorer**, right-click **Data Source Views**, and select **New Data Source View**.
2. On the **Welcome to the Data Source View Wizard** page, click **Next**.
3. On the **Select a Data Source** page, under **Relational data sources**, select the Adventure Works DW 2012 data source that you created in the last task. Click **Next**.

   Note
   If you want to create a data source, right-click **Data Sources** and then click **New Data Source** to start the Data Source Wizard.
4. On the **Select Tables and Views** page, select the following objects, and then click the right arrow to include them in the new data source view:
   - **ProspectiveBuyer (dbo)** - table of prospective bike buyers
   - **vTargetMail (dbo)** - view of historical data about past bike buyers
5. Click **Next**.
6. On the **Completing the Wizard** page, by default the data source view is named Adventure Works DW 2012. Change the name to **Targeted Mailing**, and then click **Finish**.

   The new data source view opens in the **Targeted Mailing.dsv [Design]** tab.

Previous Task in Lesson

Creating a Data Source (Basic Data Mining Tutorial)

Next Lesson

Lesson 2: Building a Targeted Mailing Scenario (Basic Data Mining Tutorial)

See Also

Defining a Data Source View (Analysis Services)

How to: Define a Data Source View Using the Data Source View Wizard (Analysis Services)

Lesson 2: Building a Targeted Mailing Structure (Basic Data Mining Tutorial)

The Marketing department of Adventure Works Cycles wants to increase sales by targeting specific customers for a mailing campaign. The company's database, , contains a list of past customers and a list of potential new customers. By investigating the attributes of previous bike buyers, the company hopes to discover patterns that they can then apply to potential customers. They hope to use the discovered patterns to predict which potential customers are most likely to purchase a bike from Adventure Works Cycles.

In this lesson you will use the **Data Mining Wizard** to create the targeted mailing structure. After you complete the tasks in this lesson, you will have a mining structure with a single model. Because there are many steps and important concepts involved in creating a structure, we have separated this process into the following three tasks:

Creating a Targeted Mailing Mining Model Structure (Basic Data Mining Tutorial)

Specifying the Data Type and Content Type (Basic Data Mining Tutorial)

Specifying a Testing Data Set for the Structure (Basic Data Mining Tutorial)

First Task in Lesson

Creating a Targeted Mailing Mining Model Structure (Basic Data Mining Tutorial)

Previous Lesson

Lesson 1: Preparing the Analysis Services Database (Basic Data Mining Tutorial)

Next Lesson

Lesson 3: Adding and Processing Models (Basic Data Mining Tutorial)

See Also

Create the Data Mining Structure (Data Mining Wizard)

Creating a New Mining Structure

Creating a Targeted Mailing Mining Model Structure (Basic Data Mining Tutorial)

The first step in creating a targeted mailing scenario is to use the Data Mining Wizard in SQL Server Data Tools (SSDT) to create a new mining structure and decision tree mining model.

In this task you will set up a new mining structure, and add an initial mining model based on the Microsoft Decision Trees algorithm. To create the structure, you will first select tables and views and then identify which columns will be used for training and which for testing.

Procedures

To create a mining structure for the targeted mailing scenario

1. In Solution Explorer, right-click **Mining Structures** and select **New Mining Structure** to start the Data Mining Wizard.
2. On the **Welcome to the Data Mining Wizard** page, click **Next**.
3. On the **Select the Definition Method** page, verify that **From existing relational database or data warehouse** is selected, and then click **Next**.
4. On the **Create the Data Mining Structure** page, under **Which data mining technique do you want to use?**, select **Microsoft Decision Trees**.

Note

If you get a warning that no data mining algorithms can be found, the project properties might not be configured correctly. This warning occurs when the project attempts to retrieve a list of data mining algorithms from the Analysis Services server and cannot find the server. By default, SQL Server Data Tools will use **localhost** as the server. If you are using a different instance, or a named instance, you must change the project properties. For more information, see Creating an Analysis Services Project (Basic Data Mining Tutorial).

5. Click **Next**.
6. On the **Select Data Source View** page, in the **Available data source views** pane, select **Targeted Mailing**. You can click **Browse** to view the tables in the data source view and then click **Close** to return to the wizard.
7. Click **Next**.
8. On the **Specify Table Types** page, select the check box in the **Case** column for vTargetMail to use it as the case table, and then click **Next**. You will use the ProspectiveBuyer table later for testing; ignore it for now.
9. On the **Specify the Training Data** page, you will identify at least one predictable column, one key column, and one input column for your model. Select the check box in the **Predictable** column in the **BikeBuyer** row.

   Note

   Notice the warning at the bottom of the window. You will not be able to navigate to the next page until you select at least one **Input** and one **Predictable** column.
10. Click **Suggest** to open the **Suggest Related Columns** dialog box.

    The **Suggest** button is enabled whenever at least one predictable attribute has been selected. The **Suggest Related Columns** dialog box lists the columns that are most closely related to the predictable column, and orders the attributes by their correlation with the predictable attribute. Columns with a significant correlation (confidence greater than 95%) are automatically selected to be included in the model.

    Review the suggestions, and then click **Cancel** to ignore the suggestions.

    Note

    If you click **OK**, all listed suggestions will be marked as input columns in the wizard. If you agree with only some of the suggestions, you must change the values manually.
11. Verify that the check box in the **Key** column is selected in the **CustomerKey** row.

    Note

If the source table from the data source view indicates a key, the Data Mining Wizard automatically chooses that column as a key for the model.

12. Select the check boxes in the **Input** column in the following rows. You can check multiple columns by highlighting a range of cells and pressing CTRL while selecting a check box.
    - **Age**
    - **CommuteDistance**
    - **EnglishEducation**
    - **EnglishOccupation**
    - **Gender**
    - **GeographyKey**
    - **HouseOwnerFlag**
    - **MaritalStatus**
    - **NumberCarsOwned**
    - **NumberChildrenAtHome**
    - **Region**
    - **TotalChildren**
    - **YearlyIncome**
13. On the far left column of the page, select the check boxes in the following rows.
    - **AddressLine1**
    - **AddressLine2**
    - **DateFirstPurchase**
    - **EmailAddress**
    - **FirstName**
    - **LastName**

    Ensure that these rows have checks only in the left column. These columns will be added to your structure but will not be included in the model. However, after the model is built, they will be available for drillthrough and testing. For more information about drillthrough, see Using Drill through on Mining Models and Mining Structures (Analysis Services - Data Mining).
14. Click **Next**.

Next Task in Lesson

Specifying the Columns used in the Mining Structure (Basic Data Mining Tutorial)

See Also

Specify Table Types (Data Mining Wizard)

Data Mining Designer

Specifying the Data Type and Content Type (Basic Data Mining Tutorial)

Now that you have selected which columns to use for building your structure and training your models, make any necessary changes to the default data and content types that are set by the wizard.

Review and modify content type and data type for each column

1. On the **Specify Columns' Content and Data Type** page, click **Detect** to run an algorithm that determines the default data and content types for each column.
2. Review the entries in the **Content Type** and **Data Type** columns and change them if necessary, to make sure that the settings are the same as those listed in the following table.

   Typically, the wizard will detect numbers and assign an appropriate numeric data type, but there are many scenarios where you might want to handle a number as text instead. For example, the **GeographyKey** should be handled as text, because it would be inappropriate to perform mathematical operations on this identifier.

Column	Content Type	Data Type
Address Line1	**Discrete**	**Text**
Address Line2	**Discrete**	**Text**
Age	**Continuous**	**Long**
Bike Buyer	**Discrete**	**Long**
Commute Distance	**Discrete**	**Text**
CustomerKey	**Key**	**Long**
DateLastPurchase	**Continuous**	**Date**
Email Address	**Discrete**	**Text**
English Education	**Discrete**	**Text**
English Occupation	**Discrete**	**Text**
FirstName	**Discrete**	**Text**
Gender	**Discrete**	**Text**
Geography Key	**Discrete**	**Text**

House Owner Flag	**Discrete**	**Text**
Last Name	**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**

3. Click **Next**.

Next Task in Lesson

Specifying a Testing Data Set for the Structure (Basic Data Mining Tutorial)

Previous Task in Lesson

Creating a Targeted Mailing Mining Model Structure (Basic Data Mining Tutorial)

See Also

Content Types (Data Mining)

Data Types (Data Mining)

Specifying a Testing Data Set for the Structure (Basic Data Mining Tutorial)

In the final few screens of the Data Mining Wizard you will split your data into a testing set and a training set. You will then name your structure and enable drillthrough on the model.

Specifying a Testing Set

Separating data into training and testing sets when you create a mining structure makes it possible to easily assess the accuracy of the mining models that you create later. For more information on testing sets, see Partitioning Data into Training and Testing Sets (Analysis Services - Data Mining).

To specify the testing set

1. On the **Create Testing Set** page, for **Percentage of data for testing**, leave the default value of **30**.
2. For **Maximum number of cases in testing data set**, type **1000**.
3. Click **Next**.

Specifying Drillthrough

Drillthrough can be enabled on models and on structures. The checkbox in this dialog box enables drillthrough on the named model. After the model has been processed, you will be able to retrieve detailed information from the training data that were used to create the model.

If the underlying mining structure has also been configured to allow drillthrough, you can retrieve detailed information from both the model cases and the mining structure, including columns that were not included in the mining model. For more information, see Using Drillthrough on Mining Models and Mining Structures (Analysis Services - Data Mining).

To name the model and structure and specify drillthrough

1. On the **Completing the Wizard** page, in **Mining structure name**, type **Targeted Mailing**.
2. In **Mining model name**, type **TM_Decision_Tree**.
3. Select the **Allow drill through** check box.
4. Review the **Preview** pane. Notice that only those columns selected as **Key**, **Input** or **Predictable** are shown. The other columns you selected (e.g., AddressLine1) are not used for building the model but will be available in the underlying structure, and can be queried after the model is processed and deployed.
5. Click **Finish**.

Previous Task in Lesson

Specifying the Columns used in the Mining Structure (Basic Data Mining Tutorial)

Next Lesson

Lesson 3: Adding and Processing Models

See Also

How to: Enable Drillthrough for a Mining Model

Using Drillthrough on Mining Models and Mining Structures (Analysis Services - Data Mining)

Specify the Training Data (Data Mining Wizard)

Lesson 3: Adding and Processing Models

The mining structure that you created in the previous lesson contains a single mining model that is based on the Microsoft Decision Trees algorithm. You can use this model to identify customers for the targeted mailing campaign. However, to ensure that your analysis is thorough, it is a common practice to create related models using different algorithms and compare their results. That way you can get different insights as well. Therefore, you will create two additional models, then process and deploy the models.

In this lesson, you will create a set of mining models that will suggest the most likely customers from a list of potential customers.

To complete the tasks in this lesson, you will use the Microsoft Clustering Algorithm and the Microsoft Naive Bayes Algorithm.

This lesson contains the following tasks:

Adding New Models to the Targeted Mailing Structure (Basic Data Mining Tutorial)

Processing Models in the Targeted Mailing Structure (Baisc Data Mining Tutorial)

First Task in Lesson

Adding New Models to the Targeted Mailing Structure (Basic Data Mining Tutorial)

Previous Lesson

Lesson 2: Building a Targeted Mailing Scenario (Basic Data Mining Tutorial)

Next Lesson

Lesson 4: Exploring the Targeted Mailing Models (Basic Data Mining Tutorial)

See Also

Adding Mining Models to a Structure (Analysis Services - Data Mining)

Adding New Models to the Targeted Mailing Structure (Basic Data Mining Tutorial)

In this task, you will define two additional models by using the **Mining Models** tab of Data Mining Designer. You will use the Microsoft Clustering and Microsoft Naive Bayes algorithms to create the models. These two algorithms are selected because of their ability to predict a discrete value (i.e., bike purchase). For more information about these algorithms, see Microsoft Clustering Algorithm (Analysis Services- Data Mining) and Microsoft Naive Bayes Algorithm

To create a clustering mining model

1. Switch to the **Mining Models** tab in Data Mining Designer in SQL Server Data Tools (SSDT).

   Notice that the designer displays two columns, one for the mining structure and one for the **TM_Decision_Tree** mining model, which you created in the previous lesson.
2. Right-click the **Structure** column and select **New Mining Model**.
3. In the **New Mining Model** dialog box, in **Model name**, type **TM_Clustering**.
4. In **Algorithm name**, select **Microsoft Clustering**.
5. Click .

The new model now appears in the **Mining Models** tab of Data Mining Designer. This model, built with the Microsoft Clustering algorithm, groups customers with similar characteristics into clusters and predicts bike buying for each cluster. Although you can modify the column usage and properties for the new model, no changes to the **TM_Clustering** model are necessary for this tutorial.

To create a Naive Bayes mining model

1. In the **Mining Models** tab of Data Mining Designer, right-click the **Structure** column, and select **New Mining Model**.
2. In the **New Mining Model** dialog box, under **Model name**, type **TM_NaiveBayes**.
3. In **Algorithm name**, select **Microsoft Naive Bayes**, then click **OK**.

   A message appears stating that the Microsoft Naive Bayes algorithm does not support the **Age** and **Yearly Income** columns, which are continuous.
4. Click **Yes** to acknowledge the message and continue.

A new model appears in the **Mining Models** tab of Data Mining Designer. Although you can modify the column usage and properties for all the models in this tab, no changes to the **TM_NaiveBayes** model are necessary for this tutorial.

Next Task in Lesson

Processing Models in the Targeted Mailing Structure (Baisc Data Mining Tutorial)

See Also

Adding Mining Models to a Structure (Analysis Services - Data Mining)

Exploring the Targeted Mailing Models (Data Mining Tutorial)

Managing Mining Models in Data Mining Designer

Processing Models in the Targeted Mailing Structure (Basic Data Mining Tutorial)

Before you can browse or work with the mining models that you have created, you must deploy the Analysis Services project and process the mining structure and mining models. *Deploying* sends the project to a server and creates any objects in that project on the server. *Processing* is the step, or series of steps, that populates Analysis Services objects with data from relational data sources. Models cannot be used until they have been deployed and processed.

Ensuring Consistency with HoldoutSeed

When you deploy a project and process the structure and models, individual rows in your data structure are randomly assigned to the training and testing set based on a random number seed. Typically, the random number seed is computed based on attributes of the data structure. For the purposes of this tutorial, in order to ensure that your results are

the same as described here, we will arbitrarily assign a fixed *holdout seed* of **12**. The holdout seed is used to initialize random sampling and ensures that the data is partitioned in roughly the same way for all mining structures and their models.

This value does not affect the number of cases in the training set; instead, it ensures that the partition can be repeated.

For more information on holdout seed, see Partitioning Data into Training and Testing Sets (Analysis Services - Data Mining).

To set the Holdout Seed

1. Click on the **Mining Structure** tab or the **Mining Models** tab in Data Mining Designer in SQL Server Data Tools (SSDT).

   Targeted Mailing MiningStructure displays in the **Properties** pane.
2. Ensure that the **Properties** pane is open by pressing **F4**.
3. Ensure that **CacheMode** is set to **KeepTrainingCases**.
4. Enter **12** for **HoldoutSeed**.

Deploying and Processing the Models

In Data Mining Designer, you can process a mining structure, a specific mining model that is associated with a mining structure, or the structure and all the models that are associated with that structure. For this task, we will process the structure and all the models at the same time.

To deploy the project and process all the mining models

1. In the **Mining Model** menu, select **Process Mining Structure and All Models**.

   If you made changes to the structure, you will be prompted to build and deploy the project before processing the models. Click **Yes**.
2. Click **Run** in the **Processing Mining Structure - Targeted Mailing** dialog box.

   The **Process Progress** dialog box opens to display the details of model processing. Model processing might take some time, depending on your computer.
3. Click **Close** in the **Process Progress** dialog box after the models have completed processing.
4. Click **Close** in the **Processing Mining Structure - <structure>** dialog box.

There are multiple ways to process a model and structure. For more information, see the following topics:

- How to: Process a Mining Model
- How to: Process a mining structure

Previous Task in Lesson

Adding New Models to the Targeted Mailing Structure (Basic Data Mining Tutorial)

Next Lesson

Exploring the Targeted Mailing Models (Basic Data Mining Tutorial)

See Also

Processing Data Mining Objects

Lesson 4: Exploring the Targeted Mailing Models (Basic Data Mining Tutorial)

After the models in your project are processed, you can explore them to look for interesting trends. Because the results of mining models are complex and can be difficult to understand in a raw format, visually investigating the data is often the easiest way to understand the rules and relationships that the algorithms have discovered within the data. Exploring also helps you to understand the behavior of the model and discover which model performs best before you deploy it.

When you use SQL Server Data Tools (SSDT) to explore your models, each model you created is listed in the **Mining Model Viewer** tab in Data Mining Designer. You can use the viewers to explore the models. These viewers are also available in SQL Server Management Studio.

Each algorithm that you used to build a model in Analysis Services returns a different type of result. Therefore, Analysis Services provides a separate viewer for each algorithm. Analysis Services also provides a generic viewer that works for all model types. The Generic Content Tree Viewer displays detailed content from the mode. The model content varies depending on the algorithm that was used. For more information, see Viewing Model Details with the Microsoft Generic Content Tree Viewer.

In this lesson you will look at the same data using your three models. Each model type is based on a different algorithm and provides different insights into the data. The Decision Tree model tells you about factors that influence bike buying. The Clustering model groups your customers by attributes that include their bike buying behavior and other selected attributes. The Naive Bayes model enables you to explore the relationship between different attributes. Finally, the Generic Content Tree Viewer reveals the structure of the model and provides richer detail including formulas, patterns that were extracted, and a count of cases in a cluster or a particular tree.

Click on the following topics to explore the mining model viewers.

- Exploring the Decision Tree Model (Basic Data Mining Tutorial)
- Exploring the Clustering Model (Basic Data Mining Tutorial)
- Exploring the Naive Bayes Model (Basic Data Mining Tutorial)

First Task in Lesson

Exploring the Decision Tree Model (Basic Data Mining Tutorial)

Previous Lesson

Lesson 3: Adding and Processing Models (Basic Data Mining Tutorial)

Next Lesson

Lesson 5: Testing Models (Basic Data Mining Tutorial)

See Also

Mining Model Viewer Tab: How-to Topics

Viewing a Data Mining Model

Exploring the Decision Tree Model (Basic Data Mining Tutorial)

The Microsoft Decision Trees algorithm predicts which columns influence the decision to purchase a bike based upon the remaining columns in the training set.

The Microsoft Decision Tree Viewer provides the following tabs for use in exploring decision tree mining models:

Decision Tree

Dependency Network

The following sections describe how to select the appropriate viewer and explore the other mining models.

- Exploring the Clustering Model
- Exploring the Naive Bayes Model

Decision Tree Tab

On the **Decision Tree** tab, you can examine all the tree models that make up a mining model.

Because the targeted mailing model in this tutorial project contains only a single predictable attribute, Bike Buyer, there is only one tree to view. If there were more trees, you could use the **Tree** box to choose another tree.

Reviewing the **TM_Decision_Tree** model in the Decision Tree viewer reveals that age is the single most important factor in predicting bike buying. Interestingly, once you group the customers by age, the next branch of the tree is different for each age node. By exploring the Decision Tree tab we can conclude that purchasers age 34 to 40 with one or no cars are very likely to purchase a bike, and that single, younger customers who live in the Pacific region and have one or no cars are also very likely to purchase a bike.

To explore the model in the Decision Tree tab

1. Select the **Mining Model Viewer** tab in **Data Mining Designer**.

   By default, the designer opens to the first model that was added to the structure -- in this case, **TM_Decision_Tree**.
2. Use the magnifying glass buttons to adjust the size of the tree display.

By default, the Microsoft Tree Viewer shows only the first three levels of the tree. If the tree contains fewer than three levels, the viewer shows only the existing levels. You can view more levels by using the **Show Level** slider or the **Default Expansion** list.

3. Slide **Show Level** to the fourth bar.
4. Change the **Background** value to **1**.

   By changing the **Background** setting, you can quickly see the number of cases in each node that have the target value of **1** for [Bike Buyer]. Remember that in this particular scenario, each case represents a customer. The value **1** indicates that the customer previously purchased a bike; the value **0** indicates that the customer has not purchased a bike. The darker the shading of the node, the higher the percentage of cases in the node that have the target value.
5. Place your cursor over the node labeled **All**. An tooltip will display the following information:
   - Total number of cases
   - Number of non bike buyer cases
   - Number of bike buyer cases
   - Number of cases with missing values for [Bike Buyer]

   Alternately, place your cursor over any node in the tree to see the condition that is required to reach that node from the node that comes before it. You can also view this same information in the **Mining Legend**.
6. Click on the node for **Age >=34 and < 41**. The histogram is displayed as a thin horizontal bar across the node and represents the distribution of customers in this age range who previously did (pink) and did not (blue) purchase a bike. The Viewer shows us that customers between the ages of 34 and 40 with one or no cars are likely to purchase a bike. Taking it one step further, we find that the likelihood to purchase a bike increases if the customer is actually age 38 to 40.

Because you enabled drillthrough when you created the structure and model, you can retrieve detailed information from the model cases and mining structure, including those columns that were not included in the mining model (e.g., emailAddress, FirstName).

For more information, see Using Drillthrough on Mining Models and Mining Structures (Analysis Services - Data Mining).

To drill through to case data

1. Right-click a node, and select **Drill Through** then **Model Columns Only**.

   The details for each training case are displayed in spreadsheet format. These details come from the vTargetMail view that you selected as the case table when building the mining structure.
2. Right-click a node, and select **Drill Through** then **Model and Structure**

Columns.

The same spreadsheet displays with the structure columns appended to the end.

Back to Top

Dependency Network Tab

The **Dependency Network** tab displays the relationships between the attributes that contribute to the predictive ability of the mining model. The Dependency Network viewer reinforces our findings that Age and Region are important factors in predicting bike buying.

To explore the model in the Dependency Network tab

1. Click the **Bike Buyer** node to identify its dependencies.

   The center node for the dependency network, **Bike Buyer**, represents the predictable attribute in the mining model. The pink shading indicates that all of the attributes have an effect on bike buying.

2. Adjust the **All Links** slider to identify the most influential attribute.

   As you lower the slider, only the attributes that have the greatest effect on the [Bike Buyer] column remain. By adjusting the slider, you can discover that age and region are the greatest factors in predicting whether someone is a bike buyer.

Next Task in Lesson

Exploring the Clustering Model (Basic Data Mining Tutorial)

See Also

Mining Model Viewer Tab: How-to Topics

Decision Tree Tab (Mining Model Viewer View)

Dependency Network Tab (Mining Model Viewer View)

Testing the Accuracy of the Mining Models (Data Mining Tutorial)

Exploring the Clustering Model (Basic Data Mining Tutorial)

The Microsoft Clustering algorithm groups cases into clusters that contain similar characteristics. These groupings are useful for exploring data, identifying anomalies in the data, and creating predictions.

The Microsoft Cluster Viewer provides the following tabs for use in exploring clustering mining models:

Cluster Diagram

Cluster Profiles

Cluster Characteristics

Cluster Discrimination

The following sections describe how to select the appropriate viewer and explore the other mining models.

- Exploring the Decision Tree Model (Basic Data Mining Tutorial)
- Exploring the Naive Bayes Model (Basic Data Mining Tutorial)

Cluster Diagram Tab

The Cluster Diagram tab displays all the clusters that are in a mining model. The lines between the clusters represent "closeness" and are shaded based on how similar the clusters are. The actual color of each cluster represents the frequency of the variable and the state in the cluster.

To explore the model in the Cluster Diagram tab

1. Use the **Mining Model** list at the top of the **Mining Model Viewer** tab to switch to the **TM_Clustering** model.
2. In the **Viewer** list, select **Microsoft Cluster Viewer**.
3. In the **Shading Variable** box, select **Bike Buyer**.

   The default variable is **Population**, but you can change this to any attribute in the model, to discover which clusters contain members that have the attributes you want.
4. Select **1** in the **State** box to explore those cases where a bike was purchased.

   The **Density** legend describes the density of the attribute state pair selected in the Shading Variable and the State. In this example it tells us that the cluster with the darkest shading has the highest percentage of bike buyers.
5. Pause your mouse over the cluster with the darkest shading.

   A tooltip displays the percentage of cases that have the attribute, **Bike Buyer = 1**.
6. Select the cluster that has the highest density, right-click the cluster, select **Rename Cluster** and type **Bike Buyers High** for later identification. Click .
7. Find the cluster that has the lightest shading (and the lowest density). Right-click the cluster, select **Rename Cluster** and type **Bike Buyers Low**. Click .
8. Click the **Bike Buyers High** cluster and drag it to an area of the pane that will give you a clear view of its connections to the other clusters.

   When you select a cluster, the lines that connect this cluster to other clusters are highlighted, so that you can easily see all the relationships for this cluster. When the cluster is not selected, you can tell by the darkness of the lines how strong the relationships are amongst all the clusters in the diagram. If the shading is light or nonexistent, the clusters are not very similar.
9. Use the slider to the left of the network, to filter out the weaker links and find the clusters with the closest relationships. The Adventure Works Cycles marketing department might want to combine similar clusters together when determining

the best method for delivering the targeted mailing.

Back to Top

Cluster Profiles Tab

The **Cluster Profiles** tab provides an overall view of the **TM_Clustering** model. The **Cluster Profiles** tab contains a column for each cluster in the model. The first column lists the attributes that are associated with at least one cluster. The rest of the viewer contains the distribution of the states of an attribute for each cluster. The distribution of a discrete variable is shown as a colored bar with the maximum number of bars displayed in the **Histogram bars** list. Continuous attributes are displayed with a diamond chart, which represents the mean and standard deviation in each cluster.

To explore the model in the Cluster Profiles tab

1. Set **Histogram** bars to **5**.

   In our model, 5 is the maximum number of states for any one variable.
2. If the **Mining Legend** blocks the display of the **Attribute profiles**, move it out of the way.
3. Select the **Bike Buyers High** column and drag it to the right of the **Population** column.
4. Select the **Bike Buyers Low** column and drag it to the right of the **Bike Buyers High** column.
5. Click the **Bike Buyers High** column.

   The **Variables** column is sorted in order of importance for that cluster. Scroll through the column and review characteristics of the Bike Buyer High cluster. For example, they are more likely to have a short commute.
6. Double-click the **Age** cell in the **Bike Buyers High** column.

   The **Mining Legend** displays a more detailed view and you can see the age range of these customers as well as the mean age.
7. Right-click the **Bike Buyers Low** column and select **Hide Column**.

Back to Top

Cluster Characteristics Tab

With the **Cluster Characteristics** tab, you can examine in more detail the characteristics that make up a cluster. Instead of comparing the characteristics of all of the clusters (as in the Cluster Profiles tab), you can explore one cluster at a time. For example, if you select **Bike Buyers High** from the **Cluster** list, you can see the characteristics of the customers in this cluster. Though the display is different from the Cluster Profiles viewer, the findings are the same.

Note

Unless you set an initial value for **holdoutseed**, results will vary each time you process the model. For more information, see HoldoutSeed Element

Back to Top

Cluster Discrimination Tab

With the **Cluster Discrimination** tab, you can explore the characteristics that distinguish one cluster from another. After you select two clusters, one from the **Cluster 1** list, and one from the **Cluster 2** list, the viewer calculates the differences between the clusters and displays a list of the attributes that distinguish the clusters most.

To explore the model in the Cluster Discrimination tab

1. In the **Cluster 1** box, select **Bike Buyers High**.
2. In the **Cluster 2** box, select **Bike Buyers Low**.
3. Click **Variables** to sort alphabetically.

   Some of the more substantial differences among the customers in the **Bike Buyers Low** and **Bike Buyers High** clusters include age, car ownership, number of children, and region.

Next Task in Lesson

Exploring the Naive Bayes Model (Basic Data Mining Tutorial)

Previous Task in Lesson

Exploring the Decision Tree Model (Basic Data Mining Tutorial)

See Also

Viewing a Mining Model with the Microsoft Cluster Viewer

Cluster Discrimination Tab (Mining Model Viewer View)

Cluster Profiles Tab (Mining Model Viewer View)

Cluster Characteristics Tab (Mining Model Viewer View)

Cluster Diagram Tab (Mining Model Viewer View)

Exploring the Naive Bayes Model (Basic Data Mining Tutorial)

The Microsoft Naive Bayes algorithm provides several methods for displaying the interaction between bike buying and the input attributes.

The Microsoft Naive Bayes Viewer provides the following tabs for use in exploring Naive Bayes mining models:

Dependency Network

Attribute Profiles

Attribute Characteristics

Attribute Discrimination

The following sections describe how to explore the other mining models.

- Exploring the Decision Tree Model (Basic Data Mining Tutorial)
- Exploring the Clustering Model (Basic Data Mining Tutorial)

Dependency Network

The **Dependency Network** tab works in the same way as the **Dependency Network** tab for the Microsoft Tree Viewer. Each node in the viewer represents an attribute, and the lines between nodes represent relationships. In the viewer, you can see all the attributes that affect the state of the predictable attribute, Bike Buyer.

To explore the model in the Dependency Network tab

1. Use the **Mining Model** list at the top of the **Mining Model Viewer** tab to switch to the **TM_NaiveBayes** model.
2. Use the **Viewer** list to switch to **Microsoft Naive Bayes Viewer**.
3. Click the **Bike Buyer** node to identify its dependencies.

   The pink shading indicates that all of the attributes have an effect on bike buying.
4. Adjust the slider to identify the most influential attribute.

   As you lower the slider, only the attributes that have the greatest effect on the [Bike Buyer] column remain. By adjusting the slider, you can discover that a few of the most influential attributes are: number of cars owned, commute distance, and total number of children.

Back to Top

Attribute Profiles

The **Attribute Profiles** tab describes how different states of the input attributes affect the outcome of the predictable attribute.

To explore the model in the Attribute Profiles tab

1. In the **Predictable** box, verify that **Bike Buyer** is selected.
2. If the **Mining Legend** is blocking display of the **Attribute profiles**, move it out of the way.
3. In the **Histogram** bars box, select **5**.

   In our model, 5 is the maximum number of states for any one variable.

   The attributes that affect the state of this predictable attribute are listed together with the values of each state of the input attributes and their distributions in each state of the predictable attribute.
4. In the **Attributes** column, find **Number Cars Owned**. Notice the differences in the histograms for bike buyers (column labeled 1) and non-buyers (column labeled 0). A person with zero or one car is much more likely to buy a bike.
5. Double-click the **Number Cars Owned** cell in the bike buyer (column labeled 1) column.

The **Mining Legend** displays a more detailed view.

Back to Top

Attribute Characteristics

With the **Attribute Characteristics** tab, you can select an attribute and value to see how frequently values for other attributes appear in the selected value cases.

To explore the model in the Attribute Characteristics tab

1. In the **Attribute** list, verify that **Bike Buyer** is selected.
2. Set the **Value** to **1**.

   In the viewer, you will see that customers who have no children at home, short commutes, and live in the North America region are more likely to buy a bike.

Back to Top

Attribute Discrimination

With the **Attribute Discrimination** tab, you can investigate the relationship between two discrete values of bike buying and other attribute values. Because the **TM_NaiveBayes** model has only two states, 1 and 0, you do not have to make any changes to the viewer.

In the viewer, you can see that people who do not own cars tend to buy bicycles, and people who own two cars tend not to buy bicycles.

Next Lesson

Lesson 3: Testing Models (Basic Data Mining Tutorial)

Previous Task in Lesson

Exploring the Clustering Model (Basic Data Mining Tutorial)

See Also

Viewing a Mining Model with the Microsoft Naive Bayes Viewer
Attribute Discrimination Tab (Mining Model Viewer View)
Attribute Profiles Tab (Mining Model Viewer View)
Attribute Characteristics Tab (Mining Model Viewer View)
Dependency Network Tab (Mining Model Viewer View)

Lesson 5: Testing Models (Basic Data Mining Tutorial)

Now that you have processed the model by using the targeted mailing scenario training set, you will test your models against the testing set. Because the data in the testing set already contains known values for bike buying, it is easy to determine whether the model's predictions are correct. The model that performs the best will be used by the

Adventure Works Cycles marketing department to identify the customers for their targeted mailing campaign.

In this lesson you will first test your models by making predictions against the testing set. Next, you will test your models on a filtered subset of the data. Analysis Services provides a variety of methods to determine the accuracy of mining models. In this lesson we will take a look at a *lift chart*.

Validation is an important step in the data mining process. Knowing how well your targeted mailing mining models perform against real data is important before you deploy the models into a production environment. For more information about how model validation fits into the larger data mining process, see Data Mining Concepts (Analysis Services - Data Mining).

This lesson contains the following tasks:

Testing Accuracy with Lift Charts (Basic Data Mining Tutorial)

Testing a Filtered Model (Basic Data Mining Tutorial)

First Task in Lesson

Testing Accuracy with Lift Charts (Basic Data Mining Tutorial)

Previous Lesson

Lesson 4: Exploring the Models (Basic Data Mining Tutorial)

Next Lesson

Lesson 6: Creating and Working with Predictions (Basic Data Mining Tutorial)

See Also

Lift Chart Tab (Mining Accuracy Chart View)

Lift Chart (Analysis Services - Data Mining)

Validating Data Mining Models

Classification Matrix Tab (Mining Accuracy Chart View)

Classification Matrix (Analysis Services - Data Mining)

Testing Accuracy with Lift Charts (Basic Data Mining Tutorial)

On the **Mining Accuracy Chart** tab of Data Mining Designer, you can calculate how well each of your models makes predictions, and compare the results of each model directly against the results of the other models. This method of comparison is referred to as a *lift chart*. Typically, the predictive accuracy of a mining model is measured by either lift or classification accuracy. For this tutorial we will use the lift chart only. For more information about lift charts and other accuracy charts, see Tools for Charting Model Accuracy (Analysis Services - Data Mining).

In this topic, you will perform the following tasks:

- Choosing Input Data

- Selecting the Models, Predictable Columns, and Values

Choosing the Input Data

The first step in testing the accuracy of your mining models is to select the data source that you will use for testing. You will test how well the models perform against your testing data and then you will use them with external data.

To select the data set

1. Switch to the **Mining Accuracy Chart** tab in Data Mining Designer in SQL Server Data Tools (SSDT) and select the **Input Selection** tab.
2. In the **Select data set to be used for Accuracy Chart** group box, select **Use mining structure test cases** to test your models by using the testing data that you set aside when you created the mining structure.

   For more information on the other options, see Measuring Mining Model Accuracy.

Selecting the Models, Predictable Columns, and Values

The next step is to select the models that you want to include in the lift chart, the predictable column against which to compare the models, and the value to predict.

Note

The mining model columns in the **Predictable Column Name** list are restricted to columns that have the usage type set to **Predict** or **Predict Only** and have a content type of **Discrete** or **Discretized**.

To show the lift of the models

1. On the **Input Selection** tab of Data Mining Designer, under **Select predictable mining model columns to show in the lift chart**, select the checkbox for **Synchronize Prediction Columns and Values**.
2. In the **Predictable Column Name** column, verify that **Bike Buyer** is selected for each model.
3. In the **Show** column, select each of the models.

   By default, all the models in the mining structure are selected. You can decide not to include a model, but for this tutorial leave all the models selected.
4. In the **Predict Value** column, select **1**. The same value is automatically filled in for each model that has the same predictable column.
5. Select the **Lift Chart** tab to display the lift chart.

   When you click the tab, a prediction query runs against the server and database for the mining structure and the input table or test data. The results are plotted on the graph.

   When you enter a **Predict Value**, the lift chart plots a Random Guess Model as

well as an Ideal Model. The mining models you created will fall between these two extremes; between a random guess and a perfect prediction. Any improvement from the random guess is considered to be *lift*.

6. Use the legend to locate the colored lines representing the Ideal Model and the Random Guess Model.

   You'll notice that the **TM_Decision_Tree** model provides the greatest lift, outperforming both the Clustering and Naive Bayes models.

For an in-depth explanation of a lift chart similar to the one created in this lesson, see Lift Chart (Analysis Services - Data Mining).

Next Task in Lesson

Adding a Filter to a Model (Basic Data Mining Tutorial)

See Also

Creating Predictions (Data Mining Tutorial)

Lift Chart Tab (Mining Accuracy Chart View)

Testing a Filtered Model (Basic Data Mining Tutorial)

Now that you have determined that the **TM_Decision_Tree** model is the most accurate, you should evaluate the model in the context of the Adventure Works Cycles targeted mailing campaign. The ssSampleDBCoFull Marketing department wants to know if there is a difference in the characteristics of male bike buyers and female bike buyers. This information will help them decide which magazines to use for advertising and which products to feature in their mailings.

In this lesson, we will create a model that is filtered on gender. You can then easily make a copy of that model, and change just the filter condition to generate a new model based on a different gender.

For more information on filters, see Creating Filters for Mining Models (Analysis Services - Data Mining).

Using Filters

Filtering enables you to easily create models built on subsets of your data. The filter is applied only to the model and does not change the underlying data source. For information on applying filters to nested tables, see Intermediate Data Mining Tutorial (Analysis Services - Data Mining).

Filters on Case Tables

First you will make a copy of the **TM_Decision_Tree** model.

To copy the Decision Tree Model

1. In SQL Server Data Tools (SSDT), in Solution Explorer, select **BasicDataMining**.
2. Click the **Mining Models** tab.

3. Right click the **TM_Decision_Tree** model, and select **New Mining Model.**
4. In the **Model name** field, type **TM_Decision_Tree_Male**.
5. Click **OK**.

Next, create a filter to select customers for the model based on their gender.

To create a case filter on a mining model

1. Right-click the **TM_Decision_Tree_Male** mining model to open the shortcut menu.

   -- or --

   Select the model. On the **Mining Model** menu, select **Set Model Filter**.
2. In the **Model Filter** dialog box, click the top row in the grid, in the **Mining Structure Column** text box.

   The drop-down list displays only the names of the columns in that table.
3. In the Mining Structure Column text box, select **Gender**.

   The icon at the left side of the text box changes to indicate that the selected item is a table or a column.
4. Click the **Operator** text box and select the equal (=) operator from the list.
5. Click the **Value** text box, and type **M**.
6. Click the next row in the grid.
7. Click **OK** to close the **Model Filter** dialog box.

   The filter displays in the **Properties** window. Alternately, you can launch the **Model Filter** dialog from the **Properties** window.
8. Repeat the above steps, but this time name the model **TM_Decision_Tree_Female** and type **F** in the **Value** text box.

You now have two new models displayed in the **Mining Models** tab.

Process the Filtered Models

Models cannot be used until they have been deployed and processed. For more information on processing models, see Processing Models in the Targeted Mailing Structure (Basic Data Mining Tutorial).

To process the filtered model

1. Right-click the **TM_Decision_Tree_Male** model and select **Process Mining Structure and all Model**s
2. Click **Run** to process the new models.
3. After processing is complete, click **Close** on both processing windows..

Evaluate the Results

View the results and assess the accuracy of the filtered models in much the same way as you did for the previous three models. For more information, see:

Exploring the Decision Tree Model (Basic Data Mining Tutorial)

Testing Accuracy with Lift Charts (Basic Data Mining Tutorial)

To explore the filtered models

1. Select the **Mining Model Viewer** tab in **Data Mining Designer**.
2. In the Mining Model box, select **TM_Decision_Tree_Male**.
3. Slide **Show Level** to **3**.
4. Change the **Background** value to **1**.
5. Place your cursor over the node labeled **All** to see the number of bike buyers versus non-bike buyers.
6. Repeat steps 1 - 5 for **TM_Decision_Tree_Female**.
7. Explore the results for the **TM_Decision_Tree** and the models filtered for gender. Compared to all bike buyers, male and female bike buyers share some of the same characteristics as the unfiltered bike buyers but all three have interesting differences as well. This is useful information that Adventure Works Cycles can use to develop their marketing campaign.

To test the lift of the filtered models

1. Switch to the **Mining Accuracy Chart** tab in Data Mining Designer in SQL Server Data Tools (SSDT) and select the **Input Selection** tab.
2. In the **Select data set to be used for Accuracy Chart** group box, select **Use mining structure test cases**.
3. On the **Input Selection** tab of Data Mining Designer, under **Select predictable mining model columns to show in the lift chart**, select the checkbox for **Synchronize Prediction Columns and Values**.
4. In the **Predictable Column Name** column, verify that **Bike Buyer** is selected for each model.
5. In the **Show** column, select each of the models.
6. In the **Predict Value** column, select **1**.
7. Select the **Lift Chart** tab to display the lift chart.

   You will now notice that all three Decision Tree models provide significant lift compared to the random guess model, and also outperform the Clustering and Naive-Bayes models.

Previous Task in Lesson

Testing the Accuracy of the Mining Models (Basic Data Mining Tutorial)

Next Lesson

Lesson 5: Creating and Working with Predictions (Basic Data Mining Tutorial)

See Also

Intermediate Data MiningTutorial (Analysis Services - Data Mining)

Mining Models Tab: How-to Topics

How to: Delete a Filter from a Mining Model

Creating Filters for Mining Models (Analysis Services - Data Mining)

Lesson 6: Creating and Working with Predictions (Basic Data Mining Tutorial)

You have trained, tested, and explored the data mining models you created. Now you are ready to use the models to identify recipients for Adventure Works Cycles targeted mailing campaign. In this lesson you will create a query to predict which customers are most likely to purchase a bike. You will also retrieve the *probability* that the prediction is correct, so that you can decide whether to present the recommendation to the marketing department or not.

Once you have identified customers with a high probability of purchasing a bike, you will drill through to the details of the cases in the mining model to retrieve names and contact information for these customers.

This lesson contains the following topics:

Creating Predictions (Basic Data Mining Tutorial)

Using Drillthrough from a Model (Basic Data Mining Tutorial)

Next Lesson

Intermediate Data Mining Tutorial (Analysis Services - Data Mining)

Previous Lesson

Lesson 5: testing Models (Analysis Services - Data Mining)

Next Task in Lesson

Creating Predictions (Basic Data Mining Tutorial)

See Also

Mining Model Content for Decision Tree Models (Analysis Services - Data Mining)

How to: Create a Prediction Query

Creating Predictions (Basic Data Mining Tutorial)

After you have tested the accuracy of your mining models and decided that you are satisfied with them, you can then create prediction queries by using the Prediction Query Builder on the **Mining Model Prediction** tab in the Data Mining Designer. This interface

helps you build queries in DMX, or the Data Mining Extensions (DMX) language. DMX has syntax like that of T-SQL but is used for queries against data mining objects.

The Prediction Query Builder has three views. With the **Design** and **Query** views, you can build and examine your query. You can then run the query and view the results in the **Result** view.

For more information about how to use the Prediction Query Builder, see Using the Prediction Query Builder to Create DMX Prediction Queries.

Creating the Query

The first step in creating a prediction query is to select a mining model and input table.

To select a model and input table

1. On the **Mining Model Prediction** tab of Data Mining Designer, in the **Mining Model** box, click **Select Model**.
2. In the **Select Mining Model** dialog box, navigate through the tree to the **Targeted Mailing** structure, expand the structure, select **TM_Decision_Tree**, and then click **OK**.
3. In the **Select Input Table(s)** box, click **Select Case Table**.
4. In the **Select Table** dialog box, in the **Data Source** list, select Adventure Works DW Multidimensional 2012 .
5. In **Table/View Name**, select the **ProspectiveBuyer (dbo)** table, and then click **OK**.

   The **ProspectiveBuyer** table most closely resembles the **vTargetMail** case table.

Mapping the Columns

After you select the input table, Prediction Query Builder creates a default mapping between the mining model and the input table, based on the names of the columns. At least one column from the structure must match a column in the external data.

Important
The data that you use to determine the accuracy of the models must contain a column that can be mapped to the predictable column. If such a column does not exist, you can create one with empty values, but it must have the same data type as the predictable column.

To map the structure columns to the input table columns

1. Right-click the lines connecting the **Mining Model** window to the **Select Input Table** window, and select **Modify Connections**.

   Notice that not every column is mapped. We will add mappings for several **Table Columns**. We will also generate a new birth date column based on the current date column, so that the columns match better.

2. Under **Table Column**, click the **Bike Buyer** cell and select ProspectiveBuyer.Unknown from the dropdown.

   This maps the predictable column, [Bike Buyer], to an input table column.
3. Click **OK**.
4. In **Solution Explorer**, right-click the **Targeted Mailing** data source view and select **View Designer**.
5. Right-click the table, ProspectiveBuyer, and select **New Named Calculation**.
6. In the **Create Named Calculation** dialog box, for **Column name**, type **calcAge**.
7. For **Description**, type **Calculate age based on birthdate**.
8. In the **Expression** box, type **DATEDIFF(YYYY,[BirthDate],getdate())** and then click **OK**.

   Because the input table has no **Age** column corresponding to the one in the mode, you can use this expression to calculate customer age from the BirthDate column in the input table. Since **Age** was identified as the most influential column for predicting bike buying, it must exist in both the model and in the input table.
9. In Data Mining Designer, select the **Mining Model Prediction** tab and re-open the **Modify Connections** window.
10. Under **Table Column**, click the **Age** cell and select ProspectiveBuyer.calcAge from the dropdown.

    Warning
    If you do not see the column in the list, you might have to refresh the definition of the data source view that is loaded in the designer. To do this, from the **File** menu, select **Save all**, and then close and re-open the project in the designer.
11. Click **OK**.

Designing the Prediction Query

To design the prediction query

1. The first button on the toolbar of the **Mining Model Prediction** tab is the **Switch to design view / Switch to result view / Switch to query view** button. Click the down arrow on this button, and select **Design**.
2. In the grid on the **Mining Model Prediction** tab, click the cell in the first empty row in the **Source** column, and then select **Prediction Function**.
3. In the **Prediction Function** row, in the **Field** column, select **PredictProbability**.

   In the **Alias** column of the same row, type **Probability of result**.
4. From the **Mining Model** window above, select and drag [Bike Buyer] into the **Criteria/Argument** cell.

When you let go, [TM_Decision_Tree].[Bike Buyer] appears in the **Criteria/Argument** cell.

This specifies the target column for the **PredictProbability** function. For more information about functions, see Data Mining Extensions (DMX) Function Reference.

5. Click the next empty row in the **Source** column, and then select TM_Decision_Tree mining model**.**
6. In the **TM_Decision_Tree** row, in the **Field** column, select **Bike Buyer**.
7. In the **TM_Decision_Tree** row, in the **Criteria/Argument** column, type **=1**.
8. Click the next empty row in the **Source** column, and then select **ProspectiveBuyer table**.
9. In the **ProspectiveBuyer** row, in the **Field** column, select **ProspectiveBuyerKey**.

   This adds the unique identifier to the prediction query so that you can identify who is and who is not likely to buy a bicycle
10. Add five more rows to the grid. For each row, select **ProspectiveBuyer table** as the **Source** and then add the following columns in the **Field** cells:
    - calcAge
    - LastName
    - FirstName
    - AddressLine1
    - AddressLine2

Finally, run the query and browse the results.

To run the query and view results

1. In the **Mining Model Prediction** tab, select the **Result** button.
2. After the query runs and the results are displayed, you can review the results.

   The **Mining Model Prediction** tab displays contact information for potential customers who are likely to be bike buyers. The **Probability of result** column indicates the probability of the prediction being correct. You can use these results to determine which potential customers to target for the mailing.
3. At this point, you can save the results. You have three options:
   - Right-click a row of data in the results, and select **Copy** to save just that value (and the column heading) to the Clipboard.
   - Right-click any row in the results, and select **Copy All** to copy the entire result set, including column headings, to the Clipboard.
   - Click **Save query result** to save the results directly to a database as follows:

   a. In the **Save Data Mining Query Result** dialog box, select a data source, or

define a new data source.

b. Type a name for the table that will contain the query results.

c. Use the option, **Add to DSV**, to create the table and add it to an existing data source view. This is useful if you want to keep all related tables for a model—such as training data, prediction source data, and query results—in the same data source view.

d. Use the option, **Overwrite if exists**, to update an existing table with the latest results.

   You must use the option to overwrite the table if you have added any columns to the prediction query, changed the names or data types of any columns in the prediction query, or if you have run any ALTER statements on the destination table.

   Also, if multiple columns have the same name (for example, the default column name **Expression**) you must create an alias for the columns with duplicate names, or an error will be raised when the designer tries to save the results to SQL Server. The reason is that SQL Server does not allow multiple columns to have the same name.

   For more information, see Save Data Mining Query Result Dialog Box.

Next Task in Lesson

Using Drillthrough from a Model (Basic Data Mining Tutorial)

See Also

How to: Create a Prediction Query

Using the Prediction Query Builder to Create DMX Prediction Queries

Using Drillthrough on Structure Data (Basic Data Mining Tutorial)

As part of their advertising campaign, Adventure Works Cycles is sending a mailer to potential customers in the 34-40 age demographic. The marketing department has decided that they would also like to send the mailer to the customers who purchased bikes from Adventure Works Cycles more than five years ago. In this lesson you will identify customers with older bikes and retrieve their contact information. This information is not included in the model, but is included in the structure. To retrieve the contact information you will first ensure that drillthrough is enabled for the structure and then you will use drillthrough to reveal the names and addresses of the targeted customers.

For information on how to drill through to model cases, see Using Drillthrough from a Model (Basic Data Mining Tutorial).

To enable drillthrough on a mining model

1. In SQL Server Data Tools (SSDT), on the **Mining Models** tab of Data Mining Designer, right-click the **TM_Decision_Tree** model, and select **Properties**.
2. In the Properties windows, click **AllowDrillthrough**, and select **True**.
3. In the Mining Models tab, right-click the model, and select **Process Model**.

For more information, see Using Drillthrough on Mining Models and Mining Structures (Analysis Services - Data Mining)

To view drillthrough data from a mining model

1. In Data Mining Designer, click the **Mining Model Viewer** tab.
2. Select the **TM_Decision_Tree** model from the **Mining Model** list.
3. Change the **Background** value to **1**. By doing this, you show only the part of the model that is related to customer who bought bikes.
4. Select the Microsoft Tree viewer from the **Viewer** list. This will force the viewer to refresh with the new filter conditions. Then, locate the **Age >=34 and <41** node and right-click the node.
5. Select **Drill Through**, and then select **Model and Structure Columns** to open the **Drill Through** window.
6. Scroll to the **Structure.Date First Purchase** column to view the purchase dates for the older bikes.
7. To copy the data to the Clipboard, right-click any row in the table, and select **Copy All**.

Congratulations, you have completed the basic data mining tutorial. Now that you are comfortable using the data mining tools, we recommend that you also complete the intermediate data mining tutorial, which demonstrates how to create models for forecasting, market basket analysis, and sequence clustering.

Previous Task in Lesson

Creating Predictions (Basic Data Mining Tutorial)

See Also

How to: Create a Prediction Query

Using the Prediction Query Builder to Create DMX Prediction Queries

Intermediate Data Mining Tutorial (Analysis Services - Data Mining)

Microsoft Analysis Services provides an integrated environment for creating and working with data mining models. You can easily bind to data sources, create and test multiple models on the same data, and deploy models for use in predictive analysis.

In the Basic Data Mining Tutorial, you learned how to use SQL Server Data Tools (SSDT) to create a data mining solution, and you built three models to support a targeted mailing campaign for analyzing customer purchasing behavior and for targeting potential buyers.

This intermediate tutorial builds on that experience and introduces several new scenarios, including common business requirements such as forecasting and market basket analysis. You will learn how to create a time series model, an association model, and a sequence clustering model. Finally, you will learn how to use neural network to explore correlations in data and to use logistic regression for predictions.

The lessons are independent and can be completed separately.

To complete the following tutorials, you should to be familiar with the data mining tools and with the mining model viewers that were introduced in the Basic Data Mining Tutorial.

All scenarios use the data source, but you will create different data source views for different scenarios. You can do the lessons in any order as long as you create the data source first.

Lesson Scenarios

After your success with the targeted mailing campaign, you have been asked to apply your knowledge of data mining to develop several new models for use in business planning. These include the following tasks:

- **Forecasting:** You will create a *time series* model, to forecast the sales of products in different regions around the world. You will develop individual models for each region and learn how to use *cross-prediction*.
- **Market basket analysis:** You will create an *association model*, to analyze groupings of products that are purchased during visits to the Adventure Works Cycles e-commerce site. Based on this market basket model, you can recommend products to customers.
- **Sequence analysis:** You build a *sequence clustering model*, to analyze the order in which customers buy products. Based on this model, you can plan changes in Web site design or new product offerings.
- **Factor analysis:** You use a *neural network* model to explore the possible causes of poor service quality in call center data. Based on the insights from the preliminary

model, you will create a *logistic regression model* to predict strategies for improving customer experience.

What You Will Learn

This tutorial teaches you how to create and work with several types of data mining algorithms. This tutorial is divided into the following lessons:

Lesson 1: Modifying a Data Source (Intermediate Data Mining Tutorial)

In this lesson, you will create a new project based on the database, to support several new data sources views and many more mining models.

Lesson 2: Building the Forecasting Scenario

In this lesson, you will create a mining model that can be used as part of a forecasting scenario. You will also explore mining models that are built with the Microsoft Time Series algorithm.

You will build models for individual regions, and then build a general model that can be used for cross-prediction.

Lesson 3: Building the Market Basket Scenario

In this lesson, you will add a new data source view and learn how to work with nested tables and keys. Based on this data, you will create a mining model that can be used as part of a market basket scenario. You will also explore mining models that are built with the Microsoft Association algorithm.

Lesson 4: Building the Sequence Clustering Scenario

In this lesson, you will create a mining model that can be used as part of a sequence clustering scenario. You will also learn how to explore mining models that are built with the Microsoft Sequence Clustering algorithm.

Lesson 5 Neural Net and Logistic Regression

In this lesson, you will create several related mining models, using the Microsoft Neural Network and Microsoft Logistic Regression algorithms. You will also learn to work with data source views to explore data underlying the models.

Requirements

Make sure that the following are installed:

- Microsoft SQL Server 2012
- Microsoft SQL Server Analysis Services
- SQL Server with the database.

By default, the sample databases are not installed, to enhance security. To install the official databases for Microsoft SQL Server, visit the Microsoft SQL Sample Databases page and select the appropriate version of the sample database.

See Also

Basic Data Mining Tutorial

Bike Buyer DMX Tutorial

Market Basket DMX Tutorial

Lesson 1: Creating the Intermediate Data Mining Solution (Intermediate Data Mining Tutorial)

In the Basic Data Mining tutorial, you created an Analysis Services project that contains a simple data mining solution based on the new database.

For this tutorial, you will create a separate Analysis Services project by using SQL Server Data Tools (SSDT). You will create a Analysis Services data source that uses , and add several new data source views to that data source, to support the scenarios and model types.

This lesson consists of the following task:

- Creating a Solution and Data Source

Next Step

Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial)

All Lessons

Lesson 1: Creating the Intermediate Data Mining Solution

Lesson 2: Forecasting Scenario (Intermediate Data Mining Tutorial)

Lesson 3: Market Basket Scenario (Intermediate Data Mining Tutorial)

Lesson 4: Sequence Clustering Scenario (Intermediate Data Mining Tutorial)

Lesson 5: Neural Network and Logistic Regression Scenario (Intermediate Data Mining Tutorial)

See Also

Data Mining Tutorial

Creating and Querying Data Mining Models with DMX: Tutorials (Analysis Services - Data Mining)

Creating a Solution and Data Source (Intermediate Data Mining Tutorial)

To work with data mining, you must first create a project in SQL Server Data Tools (SSDT) using the template, **Analysis Services Multidimensional and Data Mining Project**. When you open the template, it loads into the designer all the schemas that you might

need for data mining: data sources, mining structures and mining models, and even cubes if your mining structure uses multidimensional data.

When you create the project, your solution is stored as a local file until the solution is deployed. When you deploy the solution, Analysis Services looks for the Analysis Services server specified in the project properties, and creates a new Analysis Services database with the same name as the project. By default, Analysis Services uses the **localhost** instance for new projects. If you are using a named instance, or if you specified a different name for the default instance, you must change the deployment database property of the project to the location where you want to create your data mining objects.

For more information about Analysis Services projects, see Defining an Analysis Services Project.

Procedures

To create a new Analysis Services project for this tutorial

1. Open SQL Server Data Tools (SSDT).
2. On the **File** menu, point to **New**, and then click **Project**.
3. Select **Analysis Services Multidimensional and Data Mining Project** from the **Installed Templates** pane.
4. In the **Name** box, name the new project **DM Intermediate**.
5. Click .

To change the instance where data mining objects are stored (optional)

1. In SQL Server Data Tools (SSDT), on the **Project** menu, click **Properties**.
2. In the left side of the **Property Pages** pane, click **Deployment**.
3. Verify that the **Server** name is **localhost**. If you are using a different instance, type the name of the instance. If you are using a named instance of Analysis Services, type the machine name and then the instance name. Click .

To change the deployment properties for a project (optional)

1. In Solution Explorer, right-click the project, and then select **Properties**.

   -- or --

   In SQL Server Data Tools (SSDT), on the **Project** menu, select **Properties**.
2. In the left side of the **Property Pages** pane, click **Deployment**.

   In the **Options** pane, select **Deployment Mode**, and set the options to **Deploy All** to overwrite, or to **Deploy Changes Only** to update objects or add objects.

Creating a Data Source

In the Basic Data Mining Tutorial, you created a *data source* that stores connection information for the database. Follow the same steps to create the data source in this solution.

To create a data source

- Creating a Data Source (Basic Data Mining Tutorial)

A single data source can support multiple data source views, and each data source view can have multiple tables. However, because the data source and data source view are deployed to your Microsoft SQL Server Analysis Services database together with the data mining models that you create, as a best practice you should include in each data source view only those tables that are required for each data mining model or group of models.

In the following lessons, you will add data source views to support each of the new scenarios. Only the market basket and sequence clustering lessons use the same data source view; otherwise, each scenario uses a different data source view, so the lessons are independent of each other and can be completed separately.

Scenario	Data included in the data source view
Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial)	Monthly sales reports for bicycle models in different regions, collected as a single view.
Lesson 3: Building a Market Basket Scenario (Intermediate Data Mining Tutorial)	A table containing a list of customer orders, and a nested table showing the individual purchases for each customer.
Lesson 4: Building a Sequence Clustering Scenario (Intermediate Data Mining Tutorial)	The same data that is used for the market basket analysis, with the addition of an identifier that shows the order in which items were purchased.
Lesson 5: Building a Neural Network Model (Intermediate Data Mining Tutorial)	A single table containing some preliminary performance tracking data from a call center.

Next Lesson

Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial)

See Also

Defining Data Sources (Analysis Services)

Designing Data Source Views (Analysis Services)

Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial)

As the sales analyst for Adventure Works Cycles, you have been asked to forecast the sales of products for the next year. In particular, you have been asked to compare forecasts for the different regions and product lines. Additionally, you have been asked to determine whether sales of different products vary depending on the time of the year.

To find the requested information, in this lesson you will summarize the company's sales data at the monthly level, and you will also summarize sales figures by three regions: Europe, North America, and the Pacific.

After you complete the tasks in this lesson, you will be able to answer the following questions:

- How do the sales of different bike models change over time?
- Are there differences between the patterns for sales in the three regions?
- Can we forecast sales peaks?

The lesson can be completed in two parts:

- Part One introduces the basics of how to create and use a time series model.
- Part Two walks you through creation of a general time series model, based on all regions, that can be used for *cross-prediction*.

To complete the tasks in this lesson, which are listed below, you will use the data source that you created in Lesson 1: Creating the Intermediate Data Mining Solution (Intermediate Data Mining Tutorial).

Warning
The dates in the Adventure Works Cycles sample database have been updated for this release. If you use an earlier version of Adventure Works Cycles, you can build the model following these steps, but you might see different results.

Creating a Simple Forecasting Model

- Adding a Data Source View for Forecasting (Intermediate Data Mining Tutorial)
- Creating a Forecasting Mining Model Structure (Data Mining Tutorial)
- Modifying the Forecasting Structure (Data Mining Tutorial)
- Customizing and Processing the Forecasting Model (Intermediate Data Mining Tutorial)
- Exploring the Forecasting Model (Data Mining Tutorial)
- Creating Time Series Predictions (Intermediate Data Mining Tutorial)

Creating a General Forecasting Model for Cross-Prediction

- Adding an Aggregated Forecasting Model (Intermediate Data Mining Tutorial)
- Understanding Trends in the Time Series Model (Intermediate Data Mining Tutorial)

- Predicting using the Averaged Forecasting Model (Intermediate Data Mining Tutorial)
- Comparing Predictions for Forecasting Models (Intermediate Data Mining Tutorial)

Next Task in Lesson

Adding a Data Source View for Forecasting (Intermediate Data Mining Tutorial)

All Lessons

Lesson 1: Creating the Intermediate Data Mining Solution

Lesson 2: Forecasting Scenario (Intermediate Data Mining Tutorial)

Lesson 3: Market Basket Scenario (Intermediate Data Mining Tutorial)

Lesson 4: Sequence Clustering Scenario (Intermediate Data Mining Tutorial)

Lesson 5: Neural Network and Logistic Regression Scenario (Intermediate Data Mining Tutorial)

See Also

Data Mining Tutorial

Intermediate Data Mining Tutorial (Analysis Services - Data Mining)

Microsoft Time Series Algorithm (Analysis Services - Data Mining)

Adding a Data Source View for Forecasting (Intermediate Data Mining Tutorial)

In this task, you add a data source view that will be used for the forecasting scenario. A forecasting model requires that the data contains a column that can be used to identify steps in a time series. If you plan to analyze multiple series of data, all series must end on the same date or time step.

Procedures

To add a data source view

1. In Solution Explorer, right-click **Data Source Views**, and then select **New Data Source View**.
2. On the **Welcome to the Data Source View Wizard** page, click **Next**.
3. On the **Select a Data Source** page, under **Relational data sources**, select the data source. Click **Next**.

   Note
   If you do not have this data source, you can find the steps to create the data source in the Basic Data Mining Tutorial.
4. On the **Select Tables and Views** page, select the table, vTimeSeries (dbo), and then click the right arrow to add it to the data source view.

5. Click **Next**.
6. On the **Completing the Wizard** page, by default the data source view is named Adventure Works DW Multidimensional 2012 . Change the name to **SalesByRegion**, and then click **Finish**.

   Data Source View Designer opens and the **SalesByRegion** data source view appears.

Working with the Data Source View

After you have created the data source view, you can explore the data in the following ways:

- Right-click the table vTimeSeries in the designer, and select **Explore Data** to open the selected table in a grid.
- Click **Sampling options** and then use the **Data Exploration Options** dialog box to change the sampling method. Click **Refresh** to load data in the table using the new option settings. For example, you could specify the number of rows to output in the sample, or choose the top rows.
- Right-click the table vTimeSeries and select **Properties** to assign a new name to the table. You can also select individual columns from the data source view, and the modify the column properties.
- Click anywhere in the data source view design area to create a new query and assign a name to it, to create relationships between tables, or to change the layout of the design area.
- Right-click a table and select **New Named Calculation** to create derived columns, including aggregations. You can also add new tables and views from the data source in this view.

In the next task, you will explore the time series data and determine the best column to use as the time series identifier. You will also learn how to handle gaps in time series data.

Next Task in Lesson

Understanding

See Also

Microsoft Time Series Algorithm (Analysis Services - Data Mining)

Understanding the Requirements for a Time Series Model (Intermediate Data Mining Tutorial)

When you are preparing data for use in a forecasting model, you must ensure that your data contains a column that can be used to identify the steps in the time series. That column will be designated as the **Key Time** column. Because it is a key, the column must contain unique numeric values.

Choosing the right unit for the **Key Time** column is an important part of analysis. For example, suppose your sales data is refreshed on a minute by minute basis. You would not necessarily use minutes as the unit for the time series; you might find it more meaningful to roll up sales data by the day, week, or even month. If you are unsure which unit of time to use, you can create a new data source view for each aggregation, and build related models, to see if different trends emerge at each level of aggregation.

For this tutorial, sales data is collected on a daily basis in the transactional sales database, but for data mining, the data has been pre-aggregated by the month, using a view.

Additionally, it is desirable for analysis that the data have as few gaps as possible. If you plan to analyze multiple series of data, all series should preferably start and end on the same date. If the data has gaps, but the gaps are not at the beginning or end of a series, you can use the MISSING_VALUE_SUBSTITUTION parameter to fill in the series. Analysis Services also provides several options for replacing missing data with values, such as using means or constants.

Warning
The PivotChart and PivotTable tools that were included in earlier versions of the data source view designer are no longer provided. We recommend that you identify gaps in time series data beforehand, by using tools such as the Data Profiler included in Integration Services.

Procedures

To identify the time key for the forecasting model

1. In the pane, **SalesByRegion.dsv [Design]**, right-click the table vTimeSeries, and then select **Explore Data**.

   A new tab opens, titled **Explore vTimeSeries Table**.

2. On the **Table** tab, review the data that is used in the TimeIndex and Reporting Date columns.

   Both are sequences with unique values and can both be used as the time series key; however, the data types of the columns are different. The Microsoft Time Series algorithm does not require a **datetime** data type, only that the values be distinct and ordered. Therefore, either column can be used as the time key for the forecasting model.

3. In the data source view design surface, select the column, Reporting Date and select **Properties**. Next, click the column TimeIndex and select **Properties**.

   The field TimeIndex has the data type System.Int32, whereas the field Reporting Date has the data type System.DateTime. Many data warehouses convert date/time values to integers and use the integer column as the key, to improve indexing performance. However, if you use this column, the Microsoft Time Series algorithm will make predictions using future values such as 201014, 201014, and

so forth. Because you want to represent your sales data forecast by using calendar dates, you will use the Reporting Date column as the unique series identifier.

To set the key in the data source view

1. In the pane **SalesByRegion.dsv**, select the vTimeSeries table.
2. Right-click the column, Reporting Date, and select **Set Logical Primary Key**.

Handling Missing Data (Optional)

If any series has missing data, you might get an error when you try to process the model. You have several ways to work around missing data:

- You can have Analysis Services fill in missing values, either by calculating a mean, or by using a previous value. You do this by setting the MISSING_VALUE_SUBSTITUTION parameter on the mining model. For more information about this parameter, see Microsoft Time Series Algorithm Technical Reference (Analysis Services - Data Mining). For information about how to change parameters on an existing mining model, see How to Parameters.
- You can alter the data source or filter the underlying view to eliminate ragged series or to replace values. You can do this in the relational data source, or you can modify the data source view by creating custom named queries or named calculations. For more information, see Designing Data Source Views (Analysis Services). A later task in this lesson provides an example of how to build both a named query and a custom calculation.

For this scenario, some data is missing at the beginning of one series: that is, there is no data for the T1000 product line until July 2007. Otherwise, all series end on the same date, and there are no missing values.

The requirement of the Microsoft Time Series algorithm is that any series that you include in a single model should have the same **ending** point. Because the T1000 bicycle model was introduced in 2007, the data for this series starts later than for other bicycle models, but the series ends on the same date; therefore the data is usable.

To close the data source view designer

- Right-click the tab, **Explore vTimeSeries Table**, and select **Close**.

Next Task in Lesson

Creating a Forecasting Structure and Model (Intermediate Data Mining Tutorial)

See Also

Microsoft Time Series Algorithm (Analysis Services - Data Mining)

Creating a Forecasting Structure and Model (Intermediate Data Mining Tutorial)

Next, you will use the Data Mining Wizard to create a new mining structure and mining model based on the data source view that you just created. In this task you will specify that the mining model should use the Microsoft Time Series algorithm.

Procedures

To create a forecasting mining structure

1. In Solution Explorer in SQL Server Data Tools (SSDT), right-click **Mining Structures** and select **New Mining Structure**.
2. On the **Welcome to the Data Mining Wizard** page, click **Next**.
3. On the **Select the Definition Method** page, verify that **From existing relational database or data warehouse** is selected, and then click **Next**.
4. On the **Create the Data Mining Structure** page, under **Which data mining technique do you want to use?**, select **Microsoft Time Series**, and then click **Next**.
5. On the **Select Data Source View** page, under **Available data source views**, select **SalesByRegion**.
6. Click **Next**.
7. On the **Specify Table Types** page, ensure that the check box in the **Case** column for the vTimeSeries table is selected, and then click **Next**.
8. On the **Specify the Training Data** page, select the check boxes in the **Key** column for the ModelRegion and ReportingDate columns.

   ReportingDate should be selected by default, because you specified this column as the logical primary key when you created the data source view. By adding ModelRegion as a second key, you are telling the algorithm to create a separate time series for each combination of model and region listed in this field.
9. Select the check boxes in the **Input** and **Predictable** columns for the Quantity, column, and then click **Next**.

   By selecting **Predictable**, you indicate that you want to create forecasts on the data in this column. However, because you want to base the forecasts on past data, you must also add the column as an input.
10. On the page **Specify Columns' Content and Data Type**, review the selections.

    The ModelRegion column is designated as a **Key** column and the ReportingDate column is automatically designated as a **Key Time** column. You can have only one of each type of key.
11. Click **Next**.
12. On the **Completing the Wizard** page, for **Mining structure name**, type

Forecasting.

Note
The option to enable drillthrough is not available for time series models.

13. In **Mining model name**, type **Forecasting**, and then click **Finish**.

    Data Mining Designer opens to display the **Forecasting** mining structure that you just created.

Next Task in Lesson

Modifying the Forecasting Structure (Data Mining Tutorial)

See Also

Data Mining Designer

Microsoft Time Series Algorithm

Modifying the Forecasting Structure (Intermediate Data Mining Tutorial)

The mining structure that you created in the previous task contains a single forecasting model. Before you process and explore the model, you must change its structure slightly and modify one of its properties.

Modifying the Mining Structure

You can change the mining structure by using the **Mining Structure** tab of Data Mining Designer. When you created the model with the Data Mining Wizard, you used three columns: ReportingDate, ModelRegion, and Quantity. However, the **Forecasting** table also contains an Amount column, which you can use to forecast the amount of sales. By using the **Mining Structure** tab, you can add this column from the data source view to the mining structure.

To add the Amount column to the Forecasting mining structure

1. On the **Mining Structure** tab of Data Mining Designer, in the **Data Source View** pane, select the Amount column in the vTimeSeries table.
2. Drag the Amount column from the **Data Source View** pane into the list of columns for the **Forecasting** structure.

   The Amount column is now included in the **Forecasting** mining structure.

Modifying the Columns in the Mining Model

Because you added a new column to the structure, you must define how the model will use the column. You can specify how the column will be used on the **Mining Models** tab of Data Mining Designer.

The **Mining Models** tab lists the columns that the mining structure contains in the **Structure** column of the grid, and lists the columns that the mining model contains in

the column that has the name of the model, in this case **Forecasting**. Click the names of the columns to make modifications. In the **Forecasting** mining model, the Amount column is used as an input column and is also used to forecast future sales. Therefore, you must set the properties of the column so that it can be used as both an input column and a predictable column.

Note
In the **Mining Models** tab, you can also create new models based on the same structure, and you can adjust the algorithm and column properties for each model. However, you must process the model before these changes take effect.

To define how the Amount column will be used

1. In the **Forecasting** column of the grid on the **Mining Models** tab, click the cell in the Amount row.
2. Select **Predict** from the list.

   The Amount column is now both an input column and a predictable column.

You can also change the properties of individual columns by selecting the column and opening the **Properties** window. To open the **Properties** window, right-click the column name, and then select **Properties**. If you change a property within the column for an individual model, you can change the properties only for that model. However, when you change a property within the **Structure** column, the change affects every model that is associated with the structure. Whenever you make changes to the model or structure, you must reprocess to see the effects.

Next Task in Lesson

Customizing the Forecasting Model (Intermediate Data Mining Tutorial)

See Also

Mining Structures (Analysis Services - Data Mining)

Mining Models (Analysis Services - Data Mining)

Customizing and Processing the Forecasting Model (Intermediate Data Mining Tutorial)

The Microsoft Time Series algorithm provides parameters that affect how a model is created, and how time data is analyzed. Changing these properties can significantly affect how the mining model makes predictions.

For this task in the tutorial, you will perform the following tasks to modify the model:

1. You will customize the way your model handles time periods by adding a new value for the PERIODICITY_HINT parameter.
2. You will learn about two other important parameters for the Microsoft Time Series algorithm: FORECAST_METHOD, which lets you control the method used for

forecasting, and PREDICTION_SMOOTHING, which lets you customize the blend of long-term and short-term predictions.

3. Optionally, you will tell the algorithm how you want missing values to be imputed.
4. After all the changes have been made, you will deploy and process the model.

Setting Time Series Parameters

Periodicity Hints

The PERIODICITY_HINT parameter provides the algorithm with information about additional time periods that you expect to see in the data. By default, time series models will try to automatically detect a pattern in the data. However, if you already know the expected time cycle, providing a periodicity hint can potentially improve the accuracy of the model. However, if you provide the wrong periodicity hint, it can decrease accuracy; therefore, if you are not sure what value should be used, it is best to use the default.

For example, the view used for this model aggregates sales data from Adventure Works DW Multidimensional 2012 on a monthly basis. Therefore each time slice used by the model represents one month, and all predictions will also be in terms of months. Since there are 12 months in a year and you expect that sales patterns more or less repeat on a yearly basis, you will set the PERIODICITY_HINT parameter to **12**, to indicate that 12 time slices (months) constitute one complete sales cycle.

Forecasting Method

The FORECAST_METHOD parameter controls whether the time series algorithm is optimized for short-term or long-term predictions. By default, the FORECAST_METHOD parameter is set to MIXED, which means that two different algorithms are blended and balanced to provide good results for both short-term and long-term prediction.

However, if you know that you want to use a particular algorithm, you can change the value to either ARIMA or ARTXP.

Weighting Long-Term vs. Short-Term Predictions

You can also customize the way that long-term and short-term predictions are combined by using the PREDICTION_SMOOTHING parameter. By default, this parameter is set to 0.5, which generally provides the best balance for overall accuracy.

To change the algorithm parameters

1. On the **Mining Models** tab, right-click **Forecasting**, and select **Set Algorithm Parameters**.
2. In the **PERIODICITY_HINT** row of the **Algorithm Parameters** dialog box, click the **Value** column, then type **{12}**, including the braces.

   By default, the algorithm will also add the value {1}.
3. In the **FORECAST_METHOD** row, verify that the **Value** text box is either blank or set to **MIXED**. If a different value has been entered, type **MIXED** to change the parameter back to the default value.

4. In the **PREDICTION_SMOOTHING** row, verify that the **Value** text box is either blank or set to 0.5. If a different value has been entered, click **Value** and type **0.5** to change the parameter back to the default value.

   Note
   The PREDICTION_SMOOTHING parameter is available only in SQL Server Enterprise. Therefore, you cannot view or change the value of the PREDICTION_SMOOTHING parameter in SQL Server Standard. However, the default behavior is to use both algorithms and weight them equally.

5. Click **OK**.

Handling Missing Data (Optional)

In many cases, your sales data might have gaps that are filled with nulls, or a store might have failed to meet the reporting deadline, leaving an empty cell at the end of the series. In such scenarios, Analysis Services raises the following error and will not process the model.

> "Error (Data mining): Time stamps not synchronized starting with series <series name>, of the mining model, <model name>. All time series must end at the same time mark and cannot have arbitrarily missing data points. Setting the MISSING_VALUE_SUBSTITUTION parameter to Previous or to a numeric constant will automatically patch missing data points where possible."

To avoid this error, you can specify that Analysis Services automatically provide new values to fill in the gaps by using any one of the following methods:

- Using an average value. The mean is calculated by using all valid values in the same data series.
- Using the previous value. You can substitute previous values for multiple missing cells, but you cannot fill starting values.
- Using a constant value that you supply.

To specify that gaps be filled by averaging values

1. On the **Mining Models** tab, right-click the **Forecasting** column, and select **Set Algorithm Parameters**.
2. In the **Algorithm Parameters** dialog box, in the **MISSING_VALUE_SUBSTITUTION** row, click the **Value** column, and type **Mean**.

Build the Model

To use the model, you must deploy it to a server, and process the model by running the training data through the algorithm.

To process the forecasting model

1. On the **Mining Model** menu of SQL Server Data Tools, select **Process Mining**

Structure and All Models.

2. At the warning asking whether you want to build and deploy the project, click **Yes**.
3. In the **Process Mining Structure - Forecasting** dialog box, click **Run**.

   The **Process Progress** dialog box opens to display information about model processing. Model processing may take some time.
4. After processing is complete, click **Close** to exit the **Process Progress** dialog box.
5. Click **Close** again to exit the **Process Mining Structure - Forecasting** dialog box.

Next Task in Lesson

Exploring the Forecasting Model (Data Mining Tutorial)

See Also

Microsoft Time Series Algorithm Technical Reference (Analysis Services - Data Mining)

Microsoft Time Series Algorithm

Processing Data Mining Objects

Exploring the Forecasting Model (Intermediate Data Mining Tutorial)

Now that you have built the forecasting mining model, you can explore the results by using the **Mining Model Viewer** tab of Data Mining Designer. The Microsoft Time Series Viewer contains two tabs: **Charts** and **Model**.

Additionally, you can use the Microsoft Generic Tree Viewer with all models. Each view presents a slightly different picture of the information in the time series model.

- Charts Tab
- Model Tab
- Microsoft Generic Content Viewer

Charts Tab

The **Charts** tab of the Microsoft Time Series Viewer graphically shows you each of the series, including historical data and predictions. Each line in the time series graph represents a unique combination of product, region, and predictable attribute.

The legend on the right side of the viewer lists the time series that available, based on the selections in the drop-down list. You can select and clear the check boxes in the legend to control which time series displays in the graph.

You can also change the display options, such as the colors used for each time series, or whether values are displayed at points in the chart.

To select a time series

1. Click the **Charts** tab of the **Mining Model Viewer** tab, if it is not visible.

2. Click the drop-down list to the right of the chart view, and select all the check boxes. Click .

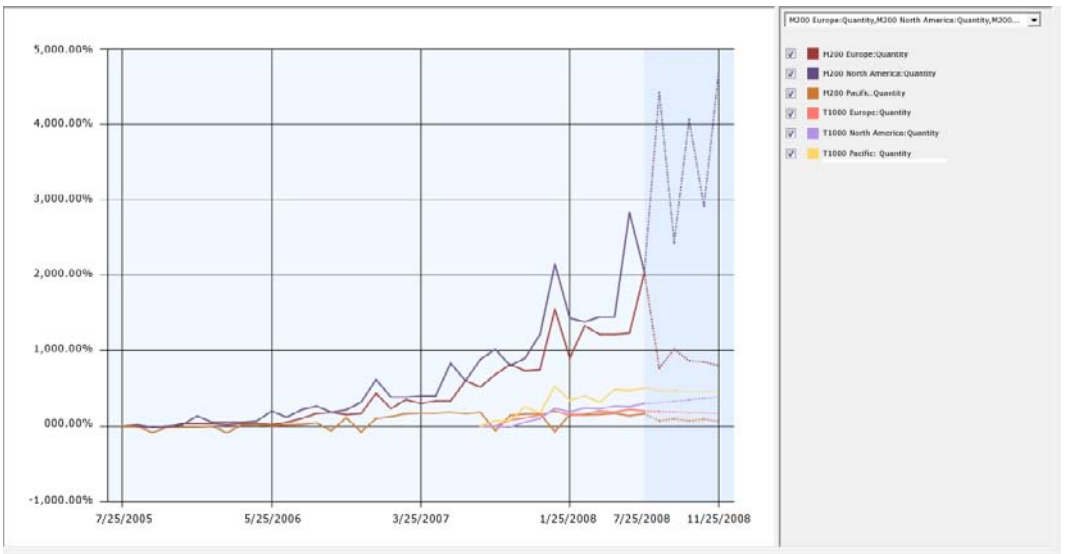
   The chart should now contain 24 different series lines.

3. In the check boxes to the right of the chart, clear the boxes to temporarily hide the lines for all series that are based on Amount.

   Now, clear the check boxes related to the R750 and R250 bicycles.

   The chart now contains just the following six series lines, so that can you more easily compare trends for the M200 and T1000 bicycles.

   - **M200 Europe: Quantity**
   - **M200 North America: Quantity**
   - **M200 Pacific: Quantity**
   - **T1000 Europe: Quantity**
   - **T1000 North America: Quantity**
   - **T1000 Pacific: Quantity**



The chart that is displayed in this viewer includes both historical and predicted data. Predicted data is shaded to differentiate it from historical data. To make it easier to compare different series, you can also change the colors associated with each line in the graph. For more information, see How to: Change the Colors used in the Data Mining Viewer.

From the trend lines, you can see that total sales for all regions are generally increasing, with a peak every 12 months in December. From the chart, you can also see that the data for the T1000 bicycle starts much later than the data for the other product series. That is

because it is a newer product, but because this series is based on much less data, the predictions might not be as accurate.

By default, five prediction steps are shown for each time series, displayed as dotted lines. You can change this value to view more or fewer predictions. You can also graphically view the standard deviation for the predictions by adding error bars to the chart.

To change prediction and display options in the Chart view

1. Try changing the value for **Prediction Steps** gradually, increasing it from **5** to **10**, and then back to **6**.

   When the historical data has large fluctuations, the fluctuations tend to be repeated or even amplified as you increase the number of predictions. You probably need to do some research at this point, to understand the cause of the big increase in the historical data and then decide whether to accept these results, seek some kind of correction in the source data, or apply some kind of smoothing in the model.
2. Select the **Show Deviations** check box.

   This option displays the estimated error for each predicted value.
3. Note the scale of the X-axis. The changes over both historical and predicted data are always expressed as a percentage, but the actual values are adjusted automatically to fit all values onto the graph. Therefore you need to be careful when comparing models to not rely on visuals alone. To get the exact value, or the percentage increase and value for predictions, pause the mouse over the dotted line or solid lines, or click the lines to view the values in the **Mining Legend**.

   Tip: If the **Mining Legend** is not visible, switch to **Model** view, right-click any node, and select **Show Legend**.

From looking at these trends, you are concerned about the lack of data for some of the series, and wonder if you might get more reliable predictions by averaging sales by model, or perhaps averaging sales by region. You will explore this approach in a later lesson in this tutorial.

Back to Top

Model Tab

The **Model** tab of the Microsoft Time Series Viewer in Data Mining Designer lets you view the forecasting model in the form of a tree graph.

First, notice that because your data describes two different measures (Amount and Quantity) for sales of multiple product lines (T1000, etc.) in three different regions (Europe, North America, and Pacific), the model that you built actually contains 24 different trees, each tree representing a model of the sales patterns for a different combination of region, product, and predictable attribute.

You can choose which combination of product line, region, and sales metric you want to view by selecting a series from the **Tree** dropdown list on the **Model** tab.

So what can you learn from viewing the model as a tree? As an example, let's compare two models, one that has several levels in the tree, and one that has a single node.

- When a tree graph contains a single node, it means the trend found in the model is mostly homogenous over time. You can use this single node, labeled **All**, to view the formula that describes the relationship between the input variables and the outcome.
- When a tree graph for a time series has multiple branches, it means the time series that was detected is too complex to be represented as a single equation. Instead, the tree graph might contain multiple branches, each branch labeled with the conditions that caused the tree to *split*. When the tree splits, each branch represents a different segment of time, inside which the trend can be described as a single equation.

  For example, if you look at the chart graph and see a sudden jump in sales volume starting sometime in September and continuing through a year-end holiday, you can switch to the **Model** view to see the exact date where the trend changed. The branches in the tree that represent "before September" and "after September" would contain different formulas: one formula that mathematically describes the sales trends up to the split, and another formula that describes sales trends for September through the year-end holiday.

To explore the decision tree for a time series model

1. In the **Tree** list on the **Model** tab of the viewer, select the **T1000 Europe: Amount** series.

   Click the node labeled **All**.

   For an **All** node, the ToolTip that appears includes information such as, the number of cases in the entire series, and time series equations derived from analysis of the data.
2. If the **Mining Legend** is not visible, right-click the node and select **Show Legend**.

   The **Mining Legend** provides much the same information that is in the Tooltip. If any of your independent variables are discrete, you will also see a histogram that shows the distribution of variables in the node.
3. Now select a different time series to view. Using the **Tree** list on the **Model** tab of the viewer, select the **M200 North America: Amount** series.

   The tree graph now contains an **All** node and two child nodes. By looking at the labels on the child nodes, you can understand at what point the trend line changed.

   For each child node, the description in the **Mining Legend** also includes the count of cases in each branch of the tree.

The following list describes some additional features in the tree viewer:

- You can change the variable that is represented in the chart by using the **Background** control. By default, nodes that are darker contain more cases, because the value of **Background** is set to **Population**. To see just how many cases there are in a node, pause the mouse over a node and view the ToolTip that appears, or click the node and view the numbers in the **Node Legend** window.
- The regression formula for the node can also be viewed in the ToolTip, or by clicking the node. If you have created a mixed model, you can see two formulas, one for ARTXP (in the leaf nodes) and one for ARIMA (in the root node of the tree).
- The little diamonds are used in nodes that represent continuous numbers. The range of the attributes is shown in the bar on which the diamond rests. The diamond is centered on the mean for the node, and the width of the diamond represents the variance of the attribute at that node.

Back to Top

(Optional) Generic Content Tree Viewer

In addition to the custom viewer for time series, Analysis Services provides the **Microsoft Generic Content Tree Viewer** for use with all data mining models. This viewer provides some advantages:

- **Microsoft Time Series Viewer**: This view merges the results of the two algorithms. Although you can view each series separately, you cannot determine how the results of each algorithm were combined. Also, in this view, the Tooltips and Mining Legend show only the most important statistics.
- **Generic Content Tree Viewer**: Lets you browse and view all of the data series that were used in the model at one time, and if you have created a mixed model, both the ARIMA and ARTXP trees are displayed in the same graph.

  You can use this viewer to get all the statistics from both algorithms, as well as distributions of the values.

  Recommended for expert users of data mining who want to know more about the ARIMA and ARTXP analyses.

To view details for a particular data series in the generic content viewer

1. In the **Mining Model Viewer** tab, select **Microsoft Generic Content Tree Viewer** from the **Viewer** drop-down list.
2. In the **Node Caption** pane, click the topmost (All) node.
3. In the **Node Details** pane, view the value for ATTRIBUTE_NAME.

   This value shows you which series, or combination of product and region, is contained in this node. In the AdventureWorks example, the topmost node is for the M200 Europe series.
4. In the **Node Caption** pane, locate the first node that has child nodes.

If a series node has children, the tree view that appears on the **Model** tab of the Microsoft Time Series Viewer will also have a branching structure.

5. Expand the node and click one of the child nodes.

   The NODE_DESCRIPTION column of the schema contains the condition that caused the tree to split.

6. In the **Node Caption** pane, click the topmost ARIMA node, and expand the node until all child nodes are visible.

7. In the **Node Details** pane, view the value for ATTRIBUTE_NAME.

   This value tells you which time series is contained in this node. The topmost node in the ARIMA section should match the topmost node in the (All) section. In the AdventureWorks example, this node contains the ARIMA analysis for the series, M200 Europe.

For more information, see Mining Model Content for Time Series Models (Analysis Services - Data Mining).

Back to Top

Next Task in Lesson

Creating Time Series Predictions (Intermediate Data Mining Tutorial)

See Also

Querying a Time Series Model (Analysis Services - Data Mining)

Microsoft Time Series Algorithm Technical Reference (Analysis Services - Data Mining)

Creating Time Series Predictions (Intermediate Data Mining Tutorial)

In the previous tasks in this lesson, you created a time series model and explored the results. By default, Analysis Services always creates a set of five (5) predictions for a time series model and displays the predicted values as part of the forecasting chart. However, you can also create forecasts by building Data Mining Extensions (DMX) prediction queries.

In this task, you will create a prediction query that generates the same predictions that you saw in the viewer. This task assumes that you have already completed the lessons in the Basic Data Mining Tutorial and are familiar with how to use Prediction Query Builder. You will now learn how to create queries specific to time series models.

Creating Time Series Predictions

Typically, the first step in creating a prediction query is to select a mining model and input table. However, a time series model does not require additional input for a regular prediction. Therefore, you do not need to specify a new source of data when making predictions, unless you are adding data to the model or replacing the data.

For this lesson, you must specify the number of prediction steps. You can specify the series name, to get a prediction for a particular combination of a product and a region.

To select a model and input table

1. On the **Mining Model Prediction** tab of the Data Mining Designer, in the **Mining Model** box, click **Select Model**.
2. In the **Select Mining Model** dialog box, expand the Forecasting structure, select the **Forecasting** model from the list, and then click **OK**.
3. Ignore the **Select Input Table(s)** box.

   Note
   For a time series model, you do not need to specify a separate input unless you are doing cross-prediction.
4. In the **Source** column, in the grid on the **Mining Model Prediction** tab, click the cell in the first empty row, and then select **Forecasting mining model**.
5. In the **Field** column, select **Model Region**.

   This action adds the series identifier to the prediction query to indicate the combination of model and region to which the prediction applies.
6. Click the next empty row in the **Source** column, and then select **Prediction Function**.
7. In the **Field** column, select **PredictTimeSeries**.

   Note
   You can also use the **Predict** function with time series models. However, by default, the Predict function creates only one prediction for each series. Therefore, to specify multiple prediction steps, you must use the **PredictTimeSeries** function.
8. In the **Mining Model** pane, select the mining model column, **Amount.** Drag Amount to the **Criteria/Arguments** box for the **PredictTimeSeries** function that you added earlier.
9. Click the **Criteria/Arguments** box, and type a comma, followed by **5**, after the field name.

   The text in the **Criteria/Arguments** box should now display the following:

   ```
   [Forecasting].[Amount],5
   ```
10. In the **Alias** column, type **PredictAmount**.
11. Click the next empty row in the **Source** column, and then select **Prediction Function** again.
12. In the **Field** column, select **PredictTimeSeries**.
13. In the **Mining Model** pane, select the column Quantity, and then drag it into the

Criteria/Arguments box for the second **PredictTimeSeries** function.

14. Click the **Criteria/Arguments** box, and type a comma, followed by **5**, after the field name.

    The text in the **Criteria/Arguments** box should now display the following:

    ```
    [Forecasting].[ Quantity],5
    ```

15. In the **Alias** column, type **PredictQuantity**.
16. Click **Switch to query result view**.

    The results of the query are displayed in tabular format.

Remember that you created three different types of results in the query builder, one that uses values from a column, and two that get predicted values from a prediction function. Therefore, the results of the query contain three separate columns. The first column contains the list of product and region combinations. The second and third columns each contain a nested table of prediction results. Each nested table contains the time step and predicted values, such as the following table:

Example results (amounts are truncated to two decimal places):

<table>
<tr><th>ModelRegion</th><th>PredictAmount</th><th>PredictQuantity</th></tr>
<tr><td>M200 Europe</td><td>
<table>
<tr><th>$TIME</th><th>Amount</th></tr>
<tr><td>7/25/2008</td><td>99978.00</td></tr>
<tr><td>8/25/2008</td><td>145575.07</td></tr>
<tr><td>9/25/2008</td><td>116835.19</td></tr>
<tr><td>10/25/2008</td><td>116537.38</td></tr>
<tr><td>11/25/2008</td><td>107760.55</td></tr>
</table>
</td><td>
<table>
<tr><th>$TIME</th><th>Quantity</th></tr>
<tr><td>7/25/2008</td><td>52</td></tr>
<tr><td>8/25/2008</td><td>67</td></tr>
<tr><td>9/25/2008</td><td>58</td></tr>
<tr><td>10/25/2008</td><td>57</td></tr>
<tr><td>11/25/2008</td><td>54</td></tr>
</table>
</td></tr>
<tr><td>M200 North America</td><td>
<table>
<tr><th>$TIME</th><th>Amount</th></tr>
<tr><td>7/25/2008</td><td>348533.93</td></tr>
<tr><td>8/25/2008</td><td>340097.98</td></tr>
<tr><td>9/25/2008</td><td>257986.19</td></tr>
<tr><td>10/25/2008</td><td>374658.24</td></tr>
</table>
</td><td>
<table>
<tr><th>$TIME</th><th>Quantity</th></tr>
<tr><td>7/25/2008</td><td>272</td></tr>
<tr><td>8/25/2008</td><td>152</td></tr>
<tr><td>9/25/2008</td><td>250</td></tr>
<tr><td>10/25/2008</td><td>181</td></tr>
</table>
</td></tr>
</table>

ModelRegion	PredictAmount		PredictQuantity	
	11/25/2008	379241.44	11/25/2008	290

Warning
The dates that are used in the sample database have changed for this release. If you are using an earlier version of the sample data, you might see different results.

Saving the Prediction Results

You have several different options for using the prediction results. You can flatten the results, copy the data from the Results view, and paste it into an Excel worksheet or other file.

To simplify the process of saving results, Data Mining Designer also provides the ability to save the data to a data source view. The functionality for saving results to a data source view is available only in SQL Server Data Tools (SSDT). The results can only be stored in a flattened format.

To flatten the results in the Results pane

1. In the Prediction Query Builder, click **Switch to query design view**.
   The view changes to allow manual editing of the DMX query text.
2. Type the **FLATTENED** keyword after the **SELECT** keyword. The complete query text should be as follows:

```
SELECT FLATTENED
  [Forecasting].[Model Region],
  (PredictTimeSeries([Forecasting].[Amount],5)) as
[PredictAmount],
  (PredictTimeSeries([Forecasting].[Quantity],5)) as
[PredictQuantity]
FROM
  [Forecasting]
```

3. Optionally, you can type a clause to restrict the results, such as the following example:

```
SELECT FLATTENED
  [Forecasting].[Model Region],
  (PredictTimeSeries([Forecasting].[Amount],5)) as
```

```
[PredictAmount],
  (PredictTimeSeries([Forecasting].[Quantity],5)) as
[PredictQuantity]
FROM
  [Forecasting]
WHERE [Forecasting].[Model Region] = 'M200 North America'
OR [Forecasting].[Model Region] = 'M200 Europe'
```

4. Click **Switch to query result view**.

To export prediction query results

1. Click **Save query results**.
2. In the **Save Data Mining Query Result** dialog box, for **Data Source**, select . You can also create a data source if you want to save the data to a different relational database.
3. In the **Table Name** column, type a new temporary table name, such as **Test Predictions**.
4. Click **Save**.

> **Note**
> To view the table that you created, create a connection to the database engine of the instance where you saved the data, and create a query.

Conclusion

You have learned how to build a basic time series model, interpret the forecasts, and create predictions.

The remaining tasks in this tutorial are optional, and describe advanced time series predictions. If you decide to go on, you will learn how to add new data to your model and create predictions on the extended series. You will also learn how to perform cross-prediction, by using the trend in the model but replacing the data with a new series of data.

Next Lesson

Adding an Averaged Forecasting Model (Intermediate Data Mining Tutorial)

See Also

Querying a Time Series Model (Analysis Services - Data Mining)

Advanced Time Series Predictions (Intermediate Data Mining Tutorial)

You saw from exploring the forecasting model that although sales in most of the regions follow a similar pattern, some regions and some models, such as the M200 model in the Pacific region, exhibit very different trends. This does not surprise you, as you know that differences among regions are common and can be caused by many factors, including marketing promotions, inaccurate reporting, or geopolitical events.

However, your users are asking for a model that can be applied worldwide. Therefore, to minimize the effect of individual factors on projections, you decide to build a model that is based on aggregated measures of worldwide sales. You can then use this model to make predictions for each individual region.

In this task, you will build all the data sources that you need to perform the advanced prediction tasks. You will create two data source views for use as inputs to the prediction query, and one data source view to use in building a new model.

Steps

1. Prepare the extended sales data (for prediction)
2. Prepare the aggregated data (for building the model)
3. Prepare the series data (for cross-prediction)
4. Predict using EXTEND
5. Create the cross-prediction model
6. Predict using REPLACE
7. Review the new predictions

Creating the New Extended Sales Data

To update your sales data, you will need to get the latest sales figures. Of particular interest are the data just in from the Pacific region, which launched a regional sales promotion to call attention to the new stores and raise awareness of their products.

For this scenario, we'll assume that the data has been imported from an Excel workbook that contains just three months of new data for a couple of regions. You'll create a table for the data using a Transact-SQL script, and then define a data source view to use for prediction.

Create the table with new sales data

1. In a Transact-SQL query window, execute the following statement to add the sales data to the AdventureWorksDW database (or any other database).

```
USE [database name];
GO
IF OBJECT_ID ([dbo].[NewSalesData]) IS NOT NULL
```

```
    DROP TABLE [dbo].[NewSalesData];
GO
CREATE TABLE [dbo].[NewSalesData](
[Series] [nvarchar](255) NULL,
[NewDate] [datetime] NULL,
[NewQty] [float] NULL,
[NewAmount] [money] NULL
) ON [PRIMARY]

GO
```

2. Insert the new values using the following script.

```
INSERT INTO [NewSalesData]
(Series,NewDate,NewQty,NewAmount)
VALUES('T1000 Pacific', '7/25/08', 55, '$130,170.22'),
('T1000 Pacific', '8/25/08', 50, '$114,435.36 '),
('T1000 Pacific', '9/25/08', 50, '$117,296.24 '),
('T1000 Europe', '7/25/08', 37, '$88,210.00 '),
('T1000 Europe', '8/25/08', 41, '$97,746.00 '),
('T1000 Europe', '9/25/08', 37, '$88,210.00 '),
('T1000 North America', '7/25/08', 69, '$164,500.00 '),
('T1000 North America', '8/25/08', 66, '$157,348.00 '),
('T1000 North America', '9/25/08', 58, '$138,276.00 '),
('M200 Pacific', '7/25/08', 65, '$149,824.35'),
('M200 Pacific', '8/25/08', 54,  '$124,619.46'),
('M200 Pacific', '9/25/08', 61, '$141,143.39'),
('M200 Europe', '7/25/08', 75, '$173,026.00'),
('M200 Europe', '8/25/08', 76, '$175,212.00'),
('M200 Europe', '9/25/08', 84, '$193,731.00'),
('M200 North America', '7/25/08', 94, '$216,916.00'),
('M200 North America', '8/25/08', 94, '$216,891.00'),
('M200 North America', '9/25/08', 91,'$209,943.00');
```

⚠ Warning

The quotation marks are used with the currency values to prevent problems with the comma separator and the currency symbol. You could also pass in the currency

values in this format: `130170.22`

Note that the dates used in the sample database have changed for this release. If you are using an earlier edition of AdventureWorks, you might need to adjust the inserted dates accordingly.

Create a data source view using the new sales data

1. In **Solution Explorer**, right-click **Data Source Views**, and then select **New Data Source View**.
2. In the Data Source View wizard, make the following selections:

   Data Source: Adventure Works DW Multidimensional 2012

   Select Tables and Views: Select the table that you just created, NewSalesData.
3. Click **Finish**.
4. In the Data Source View design surface, right-click NewSalesData, and then select **Explore Data** to verify the data.

Warning

You will use this data for prediction only, so it does not matter that the data is incomplete.

Creating the Data for the Cross-Prediction Model

The data that was used in the original forecasting model was already grouped somewhat by the view vTimeSeries, which collapsed several bike models into a smaller number of categories, and merged results from individual countries into regions. To create a model that can be used for world-wide projections, you will create some additional simple aggregations directly in the Data Source View Designer. The new data source view will contain just a sum and an average of the sales of all products for all regions.

After you have created the data source used for the model, you must create a new data source view to use for prediction. For example, if you want to predict sales for Europe using the new worldwide model, you must feed in data for the Europe region only. So you will set up a new data source view that filters the original data, and change the filter condition for each set of prediction queries.

To create the model data using a custom data source view

1. In **Solution Explorer**, right-click **Data Source Views**, and then select **New Data Source View**.
2. On the welcome page of the wizard, click **Next**.
3. On the **Select Data Source** page, select Adventure Works DW Multidimensional 2012 , and then click **Next**.
4. In the page, **Select Tables and Views**, do not add any tables—just click **Next**.
5. On the page, **Completing the Wizard**, type the name **AllRegions**, and then click

Finish.

6. Next, right-click the blank data source view design surface, and then select **New Named Query**.
7. In the **Create Named Query** dialog box, for **Name**, type **AllRegions**, and for **Description**, type **Sum and average of sales for all models and regions**.
8. In the SQL text pane, type the following statement and then click OK:

```
SELECT ReportingDate,
SUM([Quantity]) as SumQty, AVG([Quantity]) as AvgQty,
SUM([Amount]) AS SumAmt, AVG([Amount]) AS AvgAmt,
'All Regions' as [Region]
FROM dbo.vTimeSeries
GROUP BY ReportingDate
```

9. Right-click the **AllRegions** table, and then select **Explore Data**.

To create the series data for cross-prediction

1. In **Solution Explorer**, right-click **Data Source Views**, and then select **New Data Source View**.
2. In the Data Source View wizard, make the following selections:

   Data Source: Adventure Works DW Multidimensional 2012

   Select Tables and Views: Do not select any tables

   Name: **T1000 Pacific Region**
3. Click **Finish**.
4. Right-click the empty design surface for **T1000 Pacific Region.dsv**, and then select **New Named Query**.

   The **Create Named Query** dialog box appears. Retype the name, and then add the following description:

   Name: **T1000 Pacific Region**

   Description: **Filter** vTimeSeries **by region and model**
5. In the text pane, type the following query, and then click OK:

```
SELECT ReportingDate, ModelRegion, Quantity, Amount
FROM dbo.vTimeSeries
WHERE (ModelRegion = N'T1000 Pacific')
```

> **Note**
> Since you will need to create predictions for each series separately, you might want to copy the query text and save it to a text file so that you can re-use it for the other data series.

6. In the Data Source View design surface, right-click T1000 Pacific, and then select **Explore Data** to verify that the data is filtered correctly.

   You will use this data as the input to the model when creating cross-prediction queries.

Next Task in Lesson

Understanding Trends in the Time Series Model (Intermediate Data Mining Tutorial)

See Also

Microsoft Time Series Algorithm (Analysis Services - Data Mining)

Microsoft Time Series Algorithm Technical Reference (Analysis Services - Data Mining)

Designing Data Source Views (Analysis Services)

Time Series Predictions using Updated Data (Intermediate Data Mining Tutorial)

Creating Predictions using the Extended Sales Data

In this lesson, you will create a prediction query that adds the new sales data to the model. By extending the model with new data, you can get up-to-date predictions that include the newest data points.

Creating time series predictions that use new data is easy: you simply add the parameter EXTEND_MODEL_CASES to the PredictTimeSeries (DMX) function, specify the source of the new data, and specify how many predictions you want to get.

Warning

The parameter EXTEND_MODEL_CASES is optional; by default the model is extended any time that you create a time series prediction query by joining new data as inputs.

To build the prediction query and add new data

1. If the model is not already open, double-click the Forecasting structure, and in Data Mining Designer, click the **Mining Model Prediction** tab.
2. In the **Mining Model** pane, the model Forecasting should already be selected. If it is not selected, click **Select Model**, and then select the model, Forecasting.
3. In the **Select Input Table(s)** pane, click **Select Case Table**.
4. In the **Select Table** dialog box, select the data source, Adventure Works DW Multidimensional 2012 .

   From the list of data source views, select NewSalesData and then click **OK**.
5. Right-click the surface of the design area and select **Modify Connections**.
6. Using the **Modify Mapping** dialog box, map the columns in the model to the columns in the external data as follows:

- Map the ReportingDate column in the mining model to the NewDate column in the input data.
- Map the Amount column in the mining model to the NewAmount column in the input data.
- Map the Quantity column in the mining model to the NewQty column in the input data.
- Map the ModelRegion column in the mining model to the Series column in the input data.

7. Now you will build the prediction query.

   First, add a column to the prediction query to output the series the prediction applies to.

   a. In the grid, click the first empty row, under **Source**, and then select Forecasting.

   b. In the **Field** column, select Model Region and for **Alias**, type **Model Region**.

8. Next, add and edit the prediction function.

   a. Click an empty row, and under **Source**, select **Prediction Function**.

   b. For **Field**, select **PredictTimeSeries**.

   c. For **Alias**, type **Predicted Values**.

   d. Drag the field Quantity from the **Mining Model** pane into the **Criteria/Argument** column.

   e. In the **Criteria/Argument** column, after the field name, type the following text: **5,EXTEND_MODEL_CASES**

      The complete text of the **Criteria/Argument** text box should be as follows:

      ```
      [Forecasting].[Quantity],5,EXTEND_MODEL_CASES
      ```

9. Click **Results** and review the results.

   The predictions begin in July (the first time slice after the end of the original data) and end at November (the fifth time slice after the end of the original data).

You can see that to use this type of prediction query effectively, you need to know when the old data ends, as well as how many time slices there are in the new data.

For example, in this model, the original data series ended in June, and the data is for the months of July, August, and September.

Predictions that use EXTEND_MODEL_CASES always begin at the end of the original data series. Therefore, if you want to get only the predictions for the unknown months, you need to specify the starting point and the end point for prediction. Both values are specified as a number of time slices starting at the end of the old data.

The following procedure demonstrates how to do this.

Procedures

Change the start and end points of the predictions

1. In Prediction Query Builder, click **Query** to switch to DMX view.
2. Locate the DMX statement that contains the PredictTimeSeries function and change it as follows:

   ```
   PredictTimeSeries([Forecasting
   12].[Quantity],4,6,EXTEND_MODEL_CASES)
   ```

3. Click **Results** and review the results.

   Now the predictions begin at October (the fourth time slice, counting from the end of the original data) and end at December (the sixth time slice, counting from the end of the original data).

Next Task in Lesson

Predicting using the Averaged Forecasting Model (Intermediate Data Mining Tutorial)

See Also

Microsoft Time Series Algorithm Technical Reference (Analysis Services - Data Mining)

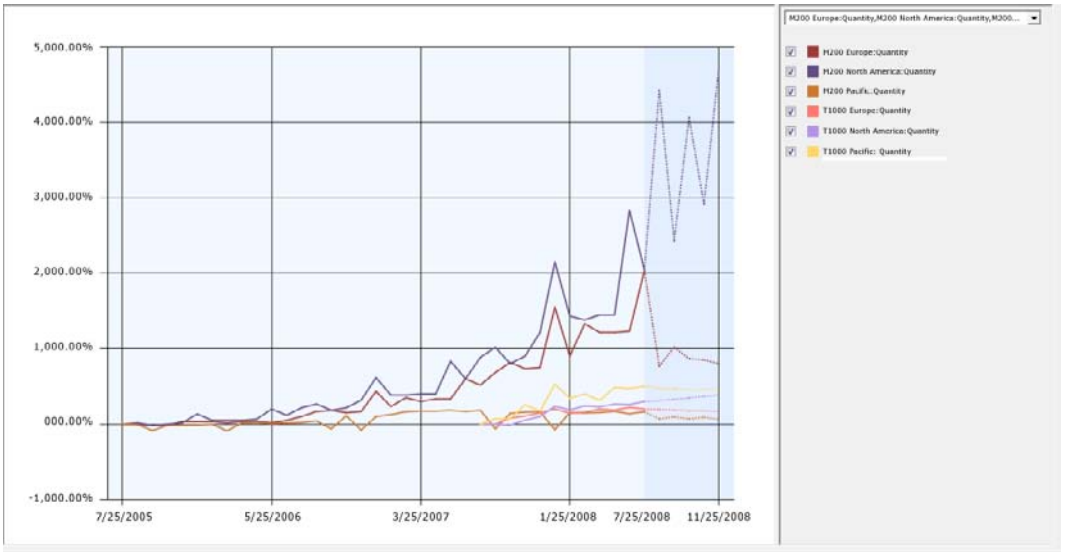
Mining Model Content for Time Series Models (Analysis Services - Data Mining)

Time Series Predictions using Replacement Data (Intermediate Data Mining Tutorial)

In this task, you will build a new model based on worldwide sales data. Then, you will create a prediction query that applies the worldwide sales model to one of the individual regions.

Building a General Model

Remember that your analysis of the results of the original mining model revealed big differences between regions and between product lines. For example, sales in North America were strong for the M200 model, while sales of the T1000 model did not do as well. However, the analysis is complicated by the fact that some series didn't have much data, or data started at a different point in time. Some data was also missing.



To address some of the data quality issues, you decide to merge the data from sales around the world, and use that set of general sales trends to build a model that can be applied to predict future sales in any region.

When you create predictions, you will use the pattern that is generated by training on worldwide sales data, but you will replace the historical data points with the sales data for each individual region. That way, the shape of the trend is preserved but the predicted values are aligned with the historical sales figures for each region and model.

Performing Cross-Prediction with a Time Series Model

The process of using data from one series to predict trends in another series is called cross-prediction. You can use cross-prediction in many scenarios: for example, you might decide that television sales are a good predictor of overall economic activity, and apply a model trained on television sales to general economic data.

In SQL Server Data Mining, you perform cross-prediction by using the parameter REPLACE_MODEL_CASES within the arguments to the function, PredictTimeSeries (DMX).

In the next task, you will learn how to use REPLACE_MODEL_CASES. You will use the merged world sales data to build a model, and then create a prediction query that maps the general model to the replacement data.

It is assumed that you are familiar with how to build data mining models by now, and so the instructions for building the model has been simplified.

To build a mining structure and mining model using the aggregated data

1. In **Solution Explorer**, right-click **Mining Structures**, and then select **New Mining Structure** to start the Data Mining Wizard.

2. In the Data Mining Wizard, make the following selections:
   - Algorithm: Microsoft Time Series
   - Use the data source that you built earlier in this advanced lesson as the source for the model. See Advanced Time Series Prediction.

     Data source view: **AllRegions**
   - Choose the following columns for the series key and time key:

     Key time: ReportingDate

     Key: Region
   - Choose the following columns for **Input** and **Predict**:

     SumQty

     SumAmt

     AvgAmt

     AvgQty
   - For **Mining structure name**, type: **All Regions**
   - For **Mining model name**, type: **All Regions**
3. Process the new structure and the new model.

To build the prediction query and map the replacement data

1. If the model is not already open, double-click the AllRegions structure, and in Data Mining Designer, click the **Mining Model Prediction** tab.
2. In the **Mining Model** pane, the model AllRegions should already be selected. If it is not selected, click **Select Model**, and then select the model, AllRegions.
3. In the **Select Input Table(s)** pane, click **Select Case Table**.
4. In the **Select Table** dialog box, change the data source to T1000 Pacific Region, and then click **OK**.
5. Right-click the join line between the mining model and the input data and select **Modify Connections**. Map the data in the data source view to the model as follows:
   a. Verify that the ReportingDate column in the mining model is mapped to the ReportingDate column in the input data.
   b. In the **Modify Mapping** dialog box, in the row for the model column AvgQty, click under **Table Column** and then select T1000 Pacific.Quantity. Click **OK**.

      This step maps the column you created in the model for predicting average quantity to the actual data from the T1000 series for sales quantity.
   c. Do not map the column Region in the model to any input column.

      Because the model aggregated the data across all series, there is no match for the series values such as T1000 Pacific, and an error is raised when the

prediction query runs.

6. Now you will build the prediction query.

   First, add a column to the results that outputs the AllRegions label from the model together with the predictions. This way you know that the results were based on the general model.

   a. In the grid, click the first empty row, under **Source**, and then select AllRegions mining model.
   b. For **Field**, select Region.
   c. For **Alias**, type **Model Used**.

7. Next, add another label to the results, so that you can see which series the prediction is for.

   a. Click an empty row, and under **Source**, select **Custom Expression**.
   b. In the **Alias** column, type **ModelRegion**.
   c. In the **Criteria/Argument** column, type **'T1000 Pacific'**.

8. Now you will set up the cross-prediction function.

   a. Click an empty row, and under **Source**, select **Prediction Function**.
   b. In the **Field** column, select **PredictTimeSeries**.
   c. For **Alias**, type **Predicted Values**.
   d. Drag the field AvgQty from the **Mining Model** pane into the **Criteria/Argument** column by using the drag and drop operation.
   e. In the **Criteria/Argument** column, after the field name, type the following text: **,5, REPLACE_MODEL_CASES**

      The complete text of the **Criteria/Argument** text box should be as follows:

      ```
      [AllRegions].[AvgQty],5,REPLACE_MODEL_CASES
      ```

9. Click **Results**.

Creating the Cross-Prediction Query in DMX

You might have noticed a problem with cross-prediction: namely, that to apply the general model to a different data series, such as the T1000 product model in the North America region, you must create a different query for each series, so that you can map the each set of inputs to the model.

However, rather than building the query in the designer, you can switch to DMX view and edit the DMX statement that you created. For example, the following DMX statement represents the query that you just built:

```
SELECT
      ([All Regions].[Region]) as [Model Used],
      ('T-1000 Pacific') as [ModelRegion],
```

```
      (PredictTimeSeries([All Regions].[Avg Qty],5,
REPLACE_MODEL_CASES)) as [Predicted Quantity]
     FROM [All Regions]
PREDICTION JOIN
    OPENQUERY([Adventure Works DW2003R2], 'SELECT [ReportingDate] FROM
      (
       SELECT  ReportingDate, ModelRegion, Quantity, Amount
       FROM dbo.vTimeSeries
       WHERE (ModelRegion = N''T1000 Pacific'')
       ) as [T1000 Pacific]    ')
    AS t
ON
[All Regions].[Reporting Date] = t.[ReportingDate]
AND
[All Regions].[Avg Qty] = t.[Quantity]
```

To apply this to a different model, you simply edit the query statement to replace the filter condition and to update the labels associated with each result.

For example, if you change the filter conditions and column labels by replacing 'Pacific' with 'North America', you will get predictions for the T1000 product in North America, based on the patterns in the general model.

Next Task in Lesson

Comparing Predictions for Forecasting Models (Intermediate Data Mining Tutorial)

See Also

Querying a Time Series Model (Analysis Services - Data Mining)

PredictTimeSeries (DMX)

Comparing Predictions for Forecasting Models (Intermediate Data Mining Tutorial)

In the previous steps of this tutorial, you created multiple time series models:

- Predictions for each combination of region and model, based only on data for the individual model and region.
- Predictions for each region, based on updated data.
- Predictions for all models on a worldwide basis, based on aggregated data.
- Predictions for the M200 model in the North America region, based on the aggregated model.

To summarize the features for time series predictions, you will review the changes to see how the use of the options to extend or replace data affected forecasting results.

EXTEND_MODEL_CASES

REPLACE_MODEL_CASES

Comparing the Original Results with Results after Adding Data

Let's look at the data for just the M200 product line in the Pacific region, to see how updating the model with new data affects the results. Remember that the original data series ended in June 2004, and we obtained new data for July, August, and September.

- The first column shows the new data that was added.
- The second column shows the forecast for July and later based on the original data series.
- The third column shows the forecast based on the extended data.

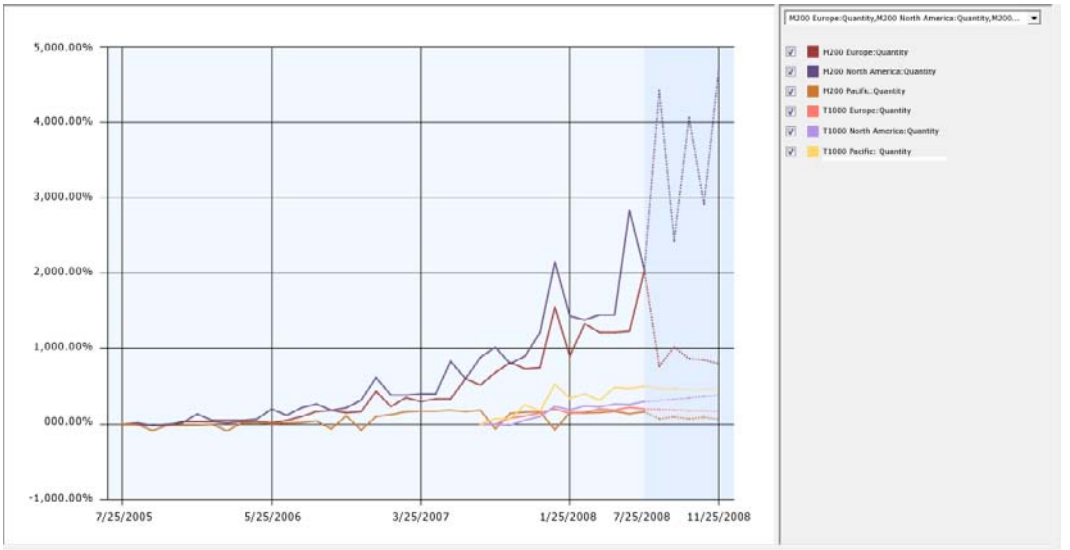
M200 Pacific	**Updated real sales data**	**Forecast before data was added**	**Extended prediction**
7-25-2008	**65**	32	**65**
8-25-2008	**54**	37	**54**
9-25-2008	**61**	32	**61**
10-25-2008	No data	36	32
11-25-2008	No data	31	41
12-25-2008	No data	34	32

You will note that the forecasts using the extended data (shown here in bold) repeat the real data points exactly. The repetition is by design. As long as there are real data points to use, the prediction query will return the actual values, and output new prediction values only after the new actual data points have been used up.

In general, the algorithm weights the changes in the new data more strongly than data from the beginning of the model data. However, in this case, the new sales figures represent an increase of only 20-30 percent over the previous period, so there only was a slight uptick in projected sales, after which the sales projections drop again, more in line with the trend in the months before the new data.

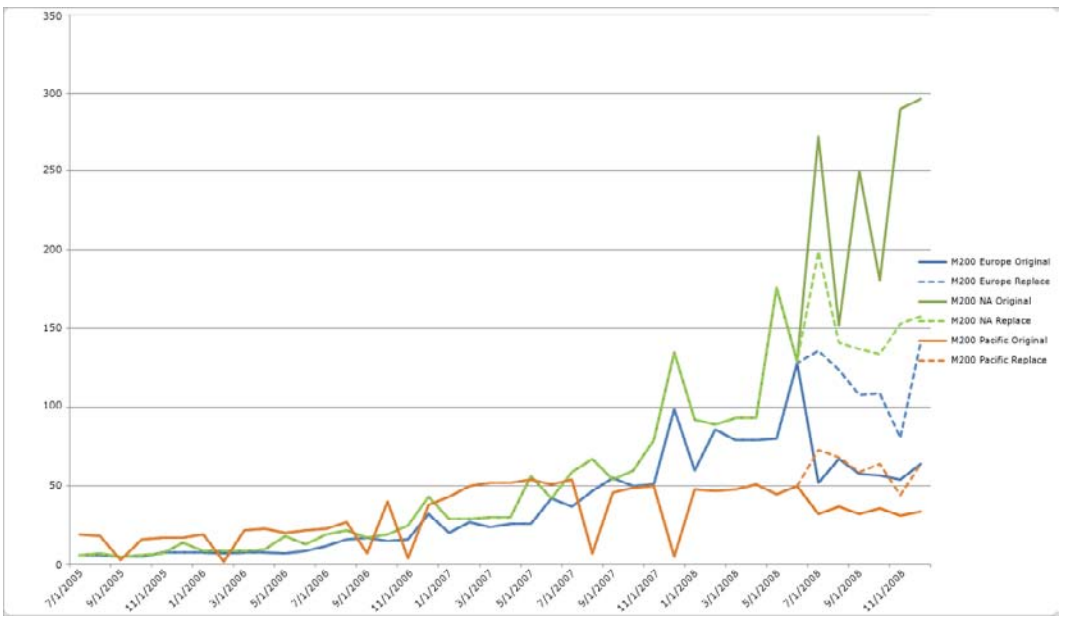
Comparing the Original and Cross-Prediction Results

Remember that the original mining model revealed big differences between regions and between product lines. For example, sales for the M200 model were very strong, while sales for the T1000 model were fairly low across all regions. Moreover, some series didn't have much data. Series were ragged, meaning they didn't have the same starting point.



So how did the predictions change when you made your projections based on the general model, which was based on world-wide sales, rather than the original data sets? To assure yourself that you have not lost any information or skewed the predictions, you can save the results to a table, join the table of predictions to the table of historical data, and then graph the two sets of historical data and predictions.

The following diagram is based on just one product line, the M200. The graph compares the predictions from the initial mining model against the predictions using the aggregated mining model.



From this diagram, you can see that the aggregated mining model preserves the overall range and trends in values while minimizing the fluctuations in the individual data series.

Conclusion

You have learned how to create and to customize a time series model that can be used for forecasting.

You have learned to update your time series models without having to reprocess them, by adding new data and creating predictions using the parameter, EXTEND_MODEL_CASES.

You have learned to create models that can be used for cross-prediction, by using the REPLACE_MODEL_CASES parameter and applying the model to a different data series.

See Also

Intermediate Data Mining Tutorial (Analysis Services - Data Mining)

Query

Lesson 3: Building a Market Basket Scenario (Intermediate Data Mining Tutorial)

The marketing department of Adventure Works Cycles wants to improve the company Web site to promote cross-selling. As part of the site update, they would like the ability to predict products that a customer might want to purchase, based on the other products that are already in the customer's online shopping basket. The marketing department also wants to understand customer purchasing behavior better, so that they can design the Web site so that the items that tend to be purchased together appear together. They have learned that data mining is especially useful for this kind of *market basket analysis* and have asked you to develop a data mining model.

After you complete the tasks in this lesson, you will have a mining model that shows groups of items from historical customer transactions. Additionally, you can use the mining model to predict additional items that a customer may want to purchase.

To complete the tasks in this lesson, you will use the solution and data source that you created in the first lesson of the Intermediate Data Mining Tutorial (Analysis Services - Data Mining). You will modify this solution by adding a data source view that contains tables about the customer, including a nested table of customer purchases. You will then build a mining model that uses the Microsoft Association Rules algorithm, which is suited to market basket scenarios.

This lesson contains the following topics:

- Adding a Data Source View with Nested Tables (Intermediate Data Mining Tutorial)
- Creating a Market Basket Structure and Model (Intermediate Data Mining Tutorial)
- Modifying the Market Basket Model (Intermediate Data Mining Tutorial)
- Exploring the Market Basket Models (Intermediate Data Mining Tutorial)

- Filtering a Nested Table in a Mining Model (Intermediate Data Mining Tutorial)
- Creating Recommendations and Predicting Associations

Next Task in Lesson

Adding a Data Source View with Nested Tables (Intermediate Data Mining Tutorial)

All Lessons

Lesson 1: Creating the Intermediate Data Mining Solution

Lesson 2: Forecasting Scenario (Intermediate Data Mining Tutorial)

Lesson 3: Market Basket Scenario (Intermediate Data Mining Tutorial)

Lesson 4: Sequence Clustering Scenario (Intermediate Data Mining Tutorial)

Lesson 5: Neural Network and Logistic Regression Scenario (Intermediate Data Mining Tutorial)

See Also

Data Mining Tutorial

Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial)

Lesson 4: Building a Sequence Clustering Scenario (Intermediate Data Mining Tutorial)

Adding a Data Source View with Nested Tables (Intermediate Data Mining Tutorial)

To create a market basket model, you must use a data source view that supports associative data. This data source view will also be used for the sequence clustering scenario.

This data source view is different from others that you may have worked with because it contains a *nested table*. A *nested table* is a table that contains multiple rows of information about a single row in the case table. For example, if your model analyzes the purchasing behavior of customers, you would typically use a table that has a unique row for each customer as the case table. However, each customer might make multiple purchases, and you might want to analyze the sequence of purchases, or products that are frequently purchased together. To logically represent these purchases in your model, you add another table to the data source view that lists the purchases for each customer.

This nested purchases table is related to the customer table by a many-to-one relationship. The nested table might contain many rows for each customer, each row containing a single product that was purchased, perhaps with additional information about the order that the purchases were made, the price at the time of the order, or any promotions that applied. You can use the information in the nested table as inputs to the model, or as the predictable attribute.

In this lesson, you do the following tasks:

- You add a data source view to the Adventure Works DW Multidimensional 2012 data source.
- You add the case and nested tables to this view.
- You specify the many-to-one relationship between the case and nested table.

  Note
  . It is important that you follow the described procedure exactly, to correctly specify the relationship between the case table and the nested table and to avoid errors when you process the model.
- You define how the columns of data are used in the model.

For more information about working with case and nested tables, and how to choose a nested table key, see Nested Tables (Analysis Services - Data Mining).

Procedures

To add a data source view

1. In Solution Explorer, right-click **Data Source Views**, and then select **New Data Source View**.

   The Data Source View Wizard opens.
2. On the **Welcome to the Data Source View Wizard** page, click **Next**.
3. On the **Select a Data Source** page, under **Relational data sources**, select the Adventure Works DW Multidimensional 2012 data source that you created in the Basic Data Mining Tutorial. Click **Next**.
4. On the **Select Tables and Views** page, select the following tables, and then click the right arrow to include them in the new data source view:
   - **vAssocSeqOrders**
   - **vAssocSeqLineItems**
5. Click **Next**.
6. On the **Completing the Wizard** page, by default the data source view is named Adventure Works DW Multidimensional 2012 . Change the name to **Orders**, and then click **Finish**.

   Data Source View Designer opens and the **Orders** data source view appears.

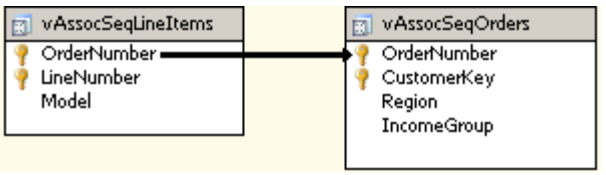
To create a relationship between tables

1. In Data Source View Designer, position the two tables so that the tables are aligned horizontally, with the vAssocSeqLineItems table on the left side and the vAssocSeqOrders table on the right side.
2. Select the **OrderNumber** column in the vAssocSeqLineItems table.
3. Drag the column to the vAssocSeqOrders table, and put it on the **OrderNumber**

column.

Important
Make sure to drag the **OrderNumber** column from the vAssocSeqLineItems nested table, which represents the many side of the join, to the vAssocSeqOrders case table, which represents the one side of the join.

A new *many-to-one relationship* now exists between the vAssocSeqLineItems and vAssocSeqOrders tables. If you have joined the tables correctly, the data source view should appear as follows:



Next Task in Lesson

Creating a Market Basket Structure and Model (Intermediate Data Mining Tutorial)

See Also

Intermediate Data Mining Tutorial (Analysis Services - Data Mining)

Mining Structures (Analysis Services - Data Mining)

Mining Models (Analysis Services - Data Mining)

Creating a Market Basket Structure and Model (Intermediate Data Mining Tutorial)

Now that you have created a data source view, you will use the Data Mining Wizard to create a new mining structure. In this task, you will create a mining structure and a mining model that is based on the Microsoft Association algorithm.

Note
If you encounter an error stating that vAssocSeqLineItems cannot be used as a nested table, return to the previous task in the lesson, and be sure to create the many-to-one join by dragging from the vAssocSeqLineItems table (the many side) to the vAssocSeqOrders table (the one side). You can also edit the relationship between the tables by right-clicking the join line.

Procedures

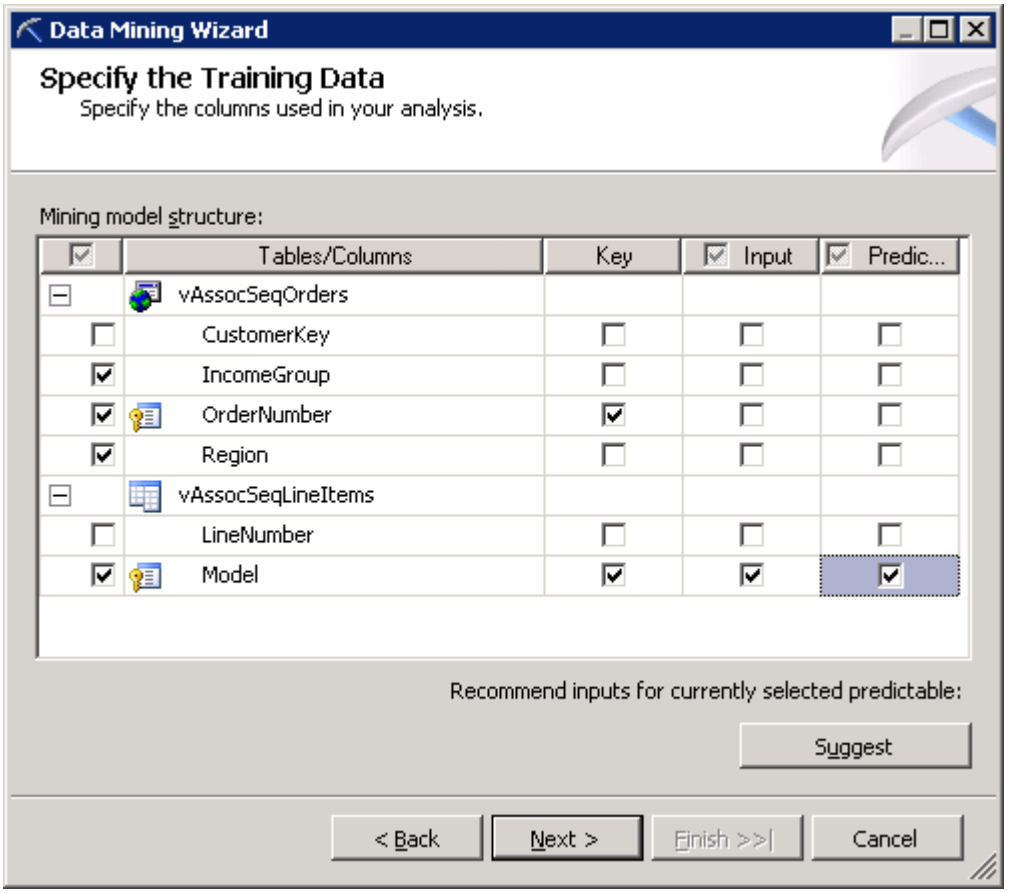
To create an association mining structure

1. In Solution Explorer in SQL Server Data Tools (SSDT), right-click **Mining Structures** and select **New Mining Structure** to open the Data Mining Wizard.

2. On the **Welcome to the Data Mining Wizard** page, click **Next**.
3. On the **Select the Definition Method** page, verify that **From existing relational database or data warehouse** is selected, and then click **Next**.
4. On the **Create the Data Mining Structure** page, under **Which data mining technique do you want to use?**, select **Microsoft Association Rules** from the list, and then click **Next**. The **Select Data Source View** page appears.
5. Select **Orders** under **Available data source views**, and then click **Next**.
6. On the **Specify Table Types** page, in the row for the vAssocSeqLineItems table, select the **Nested** check box, and in the row for the nested table vAssocSeqOrders, select the **Case** check box. Click **Next**.
7. On the **Specify the Training Data** page, clear any boxes that might be checked. Set the key for the case table, vAssocSeqOrders, by selecting the **Key** check box next to OrderNumber.

   Because the purpose of the market basket analysis is to determine which products are included in a single transaction, you do not have to use the **CustomerKey** field.
8. Set the key for the nested table, vAssocSeqLineItems, by selecting the **Key** check box next to Model. The **Input** check box is also automatically selected when you do this. Select the **Predictable** check box for **Model** as well.

   In a market basket model, you do not care about the sequence of products in the shopping basket, and therefore you should not include **LineNumber** as a key for the nested table. You would use **LineNumber** as a key only in a model where the sequence is important. You will create a model that uses the Microsoft Sequence Clustering algorithm in Lesson 4.
9. Select the check box to the left of IncomeGroup and Region,but do not make any other selections. Checking the leftmost column adds the columns to the structure for later reference, but the columns will not be used in the model. Your selections should look like the following:



10. Click **Next**.
11. On the **Specify Columns' Content and Data Type** page, review the selections, which should be as shown in the following table, and then click **Next**.

Columns	Content Type	Data Type
IncomeGroup	Discrete	Text
Order Number	Key	Text
Region	Discrete	Text
vAssocSeqLineItems		
Model	Key	Text

12. On the **Create testing set** page, the default value for the option **Percentage of data for testing** is 30 percent. Change this to **0**. Click **Next**.

Note
Analysis Services provides different charts for measuring model accuracy. However, some accuracy chart types, such as the lift chart and cross-validation report, are designed for classification and estimation. They are not supported for associative prediction.

13. On the **Completing the Wizard** page, in **Mining structure name**, type **Association**.
14. In **Mining model name**, type **Association**.
15. Select the option **Allow drill through**, and then click **Finish**.

    Data Mining Designer opens to display the **Association** mining structure that you just created.

Next Task in Lesson

Modifying the Market Basket Model (Data Mining Tutorial)

See Also

Microsoft Association Algorithm

Content Types (Data Mining)

Modifying and Processing the Market Basket Model (Intermediate Data Mining Tutorial)

Before you process the association mining model that you created, you must change the default values of two of the parameters: Support and Probability.

- Support defines the percentage of cases in which a rule must exist before it is considered valid. You will specify that a rule must be found in at least 1 percent of cases.
- Probability defines how likely an association must be before it is considered valid. You will consider any association with a probability of at least 10 percent.

For more information about the effects of increasing or decreasing support and probability, see Microsoft Association Algorithm Technical Reference (Analysis Services - Data Mining).

After you have defined the structure and parameters for the **Association** mining model, you will process the model.

Procedures

To adjust the parameters of the Association model

1. Open the **Mining Models** tab of Data Mining Designer.
2. Right-click the **Association** column in the grid in the designer and select **Set Algorithm Parameters to open the Algorithm Parameters** dialog box.

3. In the **Value** column of the **Algorithm Parameters** dialog box, set the following parameters:

   MINIMUM_PROBABILITY = 0.1

   MINIMUM_SUPPORT = 0.01

4. Click .

To process the mining model

1. On the **Mining Model** menu of SQL Server Data Tools (SSDT), select **Process Mining Structure and All Models.**
2. At the warning asking whether you want to build and deploy the project, click **Yes**.

   The **Process Mining Structure - Association** dialog box opens.
3. Click **Run**.

   The **Process Progress** dialog box opens to display information about model processing. Processing of the new structure and model might take some time.
4. After processing is complete, click **Close** to exit the **Process Progress** dialog box.
5. Click **Close** again to exit the **Process Mining Structure - Association** dialog box.

Next Task in Lesson

Exploring the Market Basket Models (Data Mining Tutorial)

See Also

Processing Data Mining Objects

Exploring the Market Basket Models (Intermediate Data Mining Tutorial)

Now that you have built the **Association** model, you can explore it by using the Microsoft Association Viewer in the **Mining Model Viewer** tab of Data Mining Designer. This tutorial walks you through using the viewer to explore relationships between items. The viewer helps you see at a glance which products tend to appear together, and get a general idea of the emerging patterns.

The Microsoft Association Viewer contains three tabs: **Rules**, **Itemsets**, and **Dependency Network**. Because each tab reveals a slightly different view of the data, when you are exploring a model, you will typically switch back and forth between the different panes several times as you pursue insights.

- Dependency Network tab
- Itemsets tab
- Rules tab
- Generic Content View

For this tutorial, you will start on the **Dependency Network** tab, and then use the **Rules** tab and **Itemsets** tab to deepen your understanding of the relationships revealed in the viewer. You will also use the **Microsoft Generic Content Tree Viewer** to retrieve detailed statistics for individual rules or itemsets.

Dependency Network Tab

With the **Dependency Network** tab, you can investigate the interaction of the different items in the model. Each node in the viewer represents an item, while the lines between them represent rules. By selecting a node, you can see which other nodes predict the selected item, or which items the current item predicts. In some cases, there is a two-way association between items, meaning that they often appear in the same transaction. You can refer to the color legend at the bottom of the tab to determine the direction of the association.

A line connecting two items means that these items are likely to appear in a transaction together. In other words, customers are likely to buy these items together. The slider is associated with the probability of the rule. Move the slider up or down to filter out weak associations, meaning rules with low probability.

The dependency network graph shows pairwise rules, which can be represented logically as A->B, meaning if Product A is purchased, then Product B is likely. The graph cannot show rules of the type AB->C. If you move the slider to show all rules but still do not see any lines in the graph, it means that there were no pairwise rules that met the criteria of the algorithm parameters.

You can also find nodes by name, by typing the first letters of the attribute name. For more information, see Find Node Dialog Box (Mining Model Viewer View).

To open the Association mode in the Microsoft Assocaition Rules Viewer

1. In **Solution Explorer**, double-click the Association structure.
2. In Data Mining Designer, click the **Mining Model Viewer** tab.
3. Select Association from the list of mining models in the **Mining Model** dropdown list.

To navigate the dependency graph and locate specific nodes

1. In the **Mining Model Viewer** tab, click the **Dependency Network** tab.
2. Click **Zoom In** several times, until you can easily view the labels for each node.

   By default, the graph displays with all nodes visible. In a complex model, there may be many nodes, making each node quite small.
3. Click the **+** sign in the lower right-hand corner of the viewer and hold down the mouse button to pan around the graph.
4. On the left side of the viewer, drag the slider down, moving it from **All Links** (the default) to the bottom of the slider control.

5. The viewer updates the graph to now show only the strongest association, between the Touring Tire and Touring Tire Tube items.
6. Click the node labeled **Touring Tire Tube = Existing**.

   The graph is updated to highlight only items that are strongly related to this item. Note the direction of the arrow between the two items.
7. On the left side of the viewer, drag the slider up again, moving it from the bottom to around the middle.

   Note the changes in the arrow that connects the two items.
8. Select **Show attribute name only** from the dropdown list at the top of the Dependency Network pane.

   The text labels in the graph are updated to show only the model name.

Back to Top

Itemsets Tab

Next, you will learn more about the rules and itemsets generated by the model for the Touring Tire and Touring Tire Tube products. The **Itemsets** tab displays three important pieces of information that relate to the itemsets that the Microsoft Association algorithm discovers:

- **Support:** The number of transactions in which the itemset occurs.
- **Size:** The number of items in the itemset.
- **Items:** A list of the items included in each itemset.

Depending on how the algorithm parameters are set, the algorithm might generate many itemsets. Each itemset that is returned in the viewer represents transactions in which the item was sold. By using the controls at the top of the **Itemsets** tab, you can filter the viewer to show only the itemsets that contain a specified minimum support and itemset size.

If you are working with a different mining model and no itemsets are listed, it is because no itemsets met the criteria of the algorithm parameters. In such a scenario, you can change the algorithm parameters to allow itemsets that have lower support.

To filter the itemsets that are shown in the viewer by name

1. Click the **Itemsets** tab of the viewer.
2. In the **Filter Itemset** box, type **Touring Tire**, and then click outside the box.

   The filter returns all items that contain this string.
3. In the **Show** list, select **Show attribute name only**.
4. Select the **Show long name** check box.

   The list of itemsets is updated to show only the itemsets that contain the string Touring Tire. The long name of the itemset includes the name of the table that contains the attribute and value for each item.

5. Clear the **Show long name** check box.

   The list of itemsets is updated to show only the short name.

The values in the **Support** column indicate the number of transactions for each itemset. A transaction for an itemset means a purchase that included all the items in the itemset.

By default, the viewer lists the itemsets in descending order by support. You can click on the column headers to sort by a different column, such as the itemset size or name. If you are interested in learning more about the individual transactions that are included in an itemset, you can drill through from the itemsets to the individual cases. The structure columns in the drillthrough results are the customer's income level and customer ID, which were not used in the model.

To view details for an itemset

1. In the list of itemsets, click the **Itemset** column heading to sort by name.
2. Locate the item, **Touring Tire** (with no second item).
3. Right-click the item, **Touring Tire**, select **Drill Through**, and then select **Model and Structure Columns**.

   The **Drill Through** dialog box displays the individual transactions used as support for this itemset.
4. Expand the nested table, vAssocSeqLineItems, to view the actual list of purchases in the transaction.

To filter itemsets by support or size

1. Clear any text that might be in the **Filter Itemset** box. You cannot use a text filter together with a numeric filter.
2. In the **Minimum support** box, type 100, and then click the background of the viewer.

   The list of itemsets is updated to show only itemsets with support of at least 100.

Back to Top

Rules Tab

The **Rules** tab displays the following information that is related to the rules that the algorithm finds.

- **Probability:** The *likelihood* of a rule, defined as the probability of the right-hand item given the left-hand side item.
- **Importance:** A measure of the usefulness of a rule. A greater value means a better rule.

  Importance is provided to help you gauge the usefulness of a rule, because probability alone can be misleading. For example, if every transaction contains a water bottle--perhaps the water bottle is added to each customer's cart

automatically as part of a promotion--the model would create a rule predicting that water bottle has a probability of 1. Based on probability alone, this rule is very accurate, but it does not provide useful information.

- **Rule:** The definition of the rule. For a market basket model, a rule describes a specific combination of items.

Each rule can be used to predict the presence of an item in a transaction based on the presence of other items. Just like in the **Itemsets** tab, you can filter the rules so that only the most interesting rules are shown. If you are working with a mining model that does not have any rules, you might want to change the algorithm parameters to lower the probability threshold for rules.

To see only rules that include the Mountain-200 bicycle

1. In the **Mining Model Viewer** tab, click the **Rules** tab.
2. In the **Filter Rule** box, enter **Mountain-200**.

   Clear the **Show long name** check box.
3. From the **Show** list, select **Show attribute name only**.

   The viewer will then display only the rules that contain the words "**Mountain-200**". The probability of the rule tells you how likely it is that when someone buys a **Mountain-200** bicycle, that person will also buy the other listed product.

The rules are ordered by probability in descending order, but you can click the column headings to change the sort order. If you are interested in finding out more details about a particular rule, you can use drillthrough to view the supporting cases.

To view cases that support a particular rule

1. In the **Rules** tab, right-click the rule that you want to view.
2. Select **Drill Through**, and then select **Model Columns Only**, or **Model and Structure Columns**.

   The **Drill Through** dialog box provides a summary of the rule at the top of the pane, and a list of all cases that were used as supporting data for the rule.

Back to Top

Generic Content Tree Viewer

This viewer can be used for all models, regardless of the algorithm or model type. The **Microsoft Generic Content Tree Viewer** is available from the **Viewer** drop-down list.

A content tree is a representation of a mining model as a series of nodes, where each node represents learned knowledge about some subset of the data. The node can contain a pattern, a set of rules, a cluster, or the definition of a range of dates that share some characteristics. The exact content of the node differs depending on the algorithm and the type of the predictable attribute, but the general representation of the content is

the same. You can expand each node to see increasing levels of detail, and copy the content of any node to the Clipboard.

To view details about the rule by using the content viewer

1. In the **Mining Model Viewer** tab, select **Microsoft Generic Content Tree Viewer** from the **Viewer** list.
2. In the Node Caption pane, scroll to the bottom of the list, and click the last node.

   The viewer shows itemsets first and rules next, but does not group them. The easiest way to find a specific node is to create a content query. For more information, see Querying an Association Model (Analysis Services - Data Mining).
3. In the Node Details pane, review the value for NODE_TYPE and NODE_DESCRIPTION.

   A node type of 8 is a rule, and a node type of 7 is an itemset. For a rule, the value of NODE_DESCRIPTION tells you the conditions that make up the rule. For an itemset, the value of NODE_DESCRIPTION tells you the items included in the itemset.

You can also create a content query to obtain detailed statistics about the rules. For more information about mining model content and how to interpret it, see Mining Model Content for Association Models (Analysis Services - Data Mining).

Back to Top

Next Task in Lesson

Filtering a Nested Table in a Mining Model (Intermediate Data Mining Tutorial)

See Also

Lesson 4: Building the Market Basket Scenario

Lesson 5: Building the Sequence Clustering Scenario

Microsoft Association Algorithm

Microsoft Association Algorithm Technical Reference (Analysis Services - Data Mining)

Filtering a Nested Table in a Mining Model (Intermediate Data Mining Tutorial)

After you have created and explored the model, you decide that you want to focus on a subset of the customer data. For example, you might want to analyze only the baskets that contain a specific item, or to analyze the demographics of customers who have not purchased anything in a certain period.

Analysis Services provides the ability to filter the data that is used in a mining model. This feature is useful because you do not need to set up a new data source view to use different data. In the Basic Data Mining Tutorial, you learned how to filter data from a flat

table by applying conditions to the case table. In this task, you create a filter that applies to a nested table.

Filters on Nested vs. Case Tables

If your data source view contains a case table and a nested table, like the data source view used in the Association model, you can filter on values from the case table, the presence or absence of a value in the nested table, or some combination of both.

In this task, you will first make a copy of the Association model and then add the IncomeGroup and Region attributes to the new related model, so that you can filter on those attributes in the case table.

To create and modify a copy of the Association model

1. In the **Mining Models** tab of SQL Server Data Tools (SSDT), right-click the **Association** model, and select **New Mining Model**.
2. For **Model Name**, type **Association Filtered**. For **Algorithm Name**, select **Microsoft Association Rules**. Click **OK**.
3. In the column for the Association Filtered model, click the IncomeGroup row and change the value from **Ignore** to **Input**.

Next, you will create a filter on the case table in the new association model. The filter will pass to the model only the customers in the target region or with the target income level. Then, you will add a second set of filter conditions to specify that the model uses only customers whose shopping baskets contained at least one item.

To add a filter to a mining model

1. In the **Mining Models** tab, right-click the model Association Filtered, and select **Set Model Filter**.
2. In the **Model Filter** dialog box, click the top row in the grid, in the **Mining Structure Column** text box.
3. In the **Mining Structure Column** text box, select IncomeGroup.

   The icon at the left side of the text box changes to indicate that the selected item is a column.
4. Click the **Operator** text box and select the **=** operator from the list.
5. Click the **Value** text box, and type **High** in the box.
6. Click the next row in the grid.
7. Click the **AND/OR** text box in the next row of the grid and select **OR**.
8. In the **Mining Structure Column** text box, select IncomeGroup. In the **Value** text box, type **Moderate**.

   The filter condition that you created is automatically added to the **Expression** text box, and should appears as follows:

```
[IncomeGroup] = 'High' OR [IncomeGroup] = 'Moderate'
```

9. Click the next row in the grid, leaving the operator as the default, **AND**.
10. For **Operator**, leave the default value, **Contains**. Click the **Value** text box.
11. In the **Filter** dialog box, in the first row under **Mining Structure Column**, select **Model**.
12. For **Operator**, select **IS NOT NULL**. Leave the **Value** text box blank. Click **OK**.

    The filter condition in the **Expression** text box of the **Model Filter** dialog box is automatically updated to include the new condition on the nested table. The completed expression is as follows:

    ```
    [IncomeGroup] = 'High' OR [IncomeGroup] = 'Moderate' AND EXISTS
    SELECT * FROM [vAssocSeqLineItems] WHERE [Model] <> NULL).
    ```

13. Click .

To enable drillthrough and to process the filtered model

1. In the **Mining Models** tab, right-click the **Association Filtered** model, and select **Properties**.
2. Change the **AllowDrillThrough** property to **True**.
3. Right-click the **Association Filtered** mining model, and select **Process Model**.
4. Click **Yes** in the error message to deploy the new model to the Analysis Services database.
5. In the **Process Mining Structure** dialog box, click **Run**.
6. When processing is complete click **Close** to exit the **Process Progress** dialog box, and click **Close** again to exit the **Process Mining Structure** dialog box.

You can verify by using the Microsoft Generic Content Tree viewer and looking at the value for NODE_SUPPORT that the filtered model contains fewer cases than the original model.

Remarks

The nested table filter that you just created checks only for the presence of at least one row in the nested table; however, you can also create filter conditions that check for the presence of specific products. For example, you could create the following filter:

```
 [IncomeGroup] = 'High' AND
  EXISTS (SELECT * FROM [<nested table name>] WHERE [Model] = 'Water
Bottle' )
```

This statement means that you are restricting the customers from the case table to only those who have purchased a water bottle. However, because the number of nested table attributes is potentially unlimited, Analysis Services does not supply a list of possible values from which to select. Instead, you must type the exact value.

You can click **Edit Query** to manually change the filter expression. However, if you change any part of a filter expression manually, the grid will be disabled and thereafter you must work with the filter expression in text edit mode only. To restore grid editing mode, you must clear the filter expression and start over.

Warning

You cannot use the LIKE operator in a nested table filter.

Next Task in Lesson

Creating Recommendations and Predicting Associations

See Also

Model Filter Syntax and Examples (Analysis Services - Data Mining)

Creating Filters for Mining Models (Analysis Services - Data Mining)

Predicting Associations (Intermediate Data Mining Tutorial)

After the models have been processed, you can use the information about associations stored in the model to create predictions. In the final task of this lesson, you learn how to build prediction queries against the association models that you created. This lesson assumes that you are familiar with how to use the Prediction Query Builder and want to learn how to build prediction queries against association models. For more information how to use Prediction Query Builder, see Using the Prediction Query Builder to Create DMX Prediction Queries.

Creating a Singleton Prediction Query

Prediction queries on an association model can be very useful:

- Recommend items to a customer, based on prior or related purchases
- Find related events.
- Identify relationships in or across sets of transactions.

To build a prediction query, you first select the association model you want to use, and then you specify the input data. Inputs can come from an external data source, such as a list of values, or you can build a singleton query and provide values as you go.

For this scenario, you will first create some singleton prediction queries, to get an idea of how prediction works. Then you will create a query for batch predictions that you could use for making recommendations based on a customer's current purchases.

To create a prediction query on an association model

1. Click the **Mining Model Prediction** tab of Data Mining Designer.
2. In the **Mining Model** pane, click **Select Model**. (You can skip this step and the next step if the correct model is already selected.)
3. In the **Select Mining Model** dialog box, expand the node that represents the mining structure **Association**, and select the model **Association**. Click **OK**.

For now, you can ignore the input pane.

4. In the grid, click the empty cell under **Source** and select **Prediction Function.** In the cell under **Field**, select **PredictAssociation**.

   You can also use the **Predict** function to predict associations. If you do, be sure to choose the version of the **Predict** function that takes a table column as argument.

5. In the **Mining Model** pane, select the nested table `vAssocSeqLineItems`, and drag it into the grid, to the **Criteria/Argument** box for the **PredictAssociation** function.

   Dragging and dropping table and column names lets you build complex statements without syntax errors. However, it replaces the current contents of the cell, which include other optional arguments for the **PredictAssociation** function. To view the other arguments, you can temporarily add a second instance of the function to the grid for reference.

6. Click the **Criteria/Argument** box and type the following text after the table name: **,3**

   The complete text in the **Criteria/Argument** box should be as follows:

   ```
   [Association].[v Assoc Seq Line Items],3
   ```

7. Click the **Results** button in the upper corner of the Prediction Query Builder.

The expected results contain a single column with the heading **Expression**. The **Expression** column contains a nested table with a single column and the following three rows. Because you did not specify an input value, these predictions represent the most likely product associations for the model as a whole.

Model
Women's Mountain Shorts
Water Bottle
Touring-3000

Next, you will use the **Singleton Query Input** pane to specify a product as input to the query, and view the products that are most likely associated with that item.

To create a singleton prediction query with nested table inputs

1. Click the **Design** button in the corner of the Prediction Query Builder to switch back to the query building grid.
2. On the **Mining Model** menu, select **Singleton Query**.
3. In the **Mining Model** dialog box, select the **Association** model.

4. In the grid, click the empty cell under **Source** and select **Prediction Function.** In the cell under **Field**, select **PredictAssociation**.
5. In the **Mining Model** pane, select the nested table `vAssocSeqLineItems`, and drag it into the grid, to the **Criteria/Argument** box for the **PredictAssociation** function. Type **,3** after the nested table name just as in the previous procedure.
6. In the **Singleton Query Input** dialog box, click the **Value** box next to **vAssoc Seq Line Items**, and then click the **(...)** button.
7. In the **Nested Table Input** dialog box, select **Touring Tire** in the **Key column** pane, and then click **Add**.
8. Click the **Results** button.

The results now show the predictions for products that are most likely associated with the Touring Tire.

Model
Touring Tire Tube
Sport-100
Water Bottle

However, you already know from exploring the model that the Touring Tire Tube is frequently purchased with the Touring Tire; you are more interested in knowing what products you can recommend to customers who purchase these items together. You will change the query so that it predicts related products based on two items in the basket. You will also modify the query to add the probability for each predicted product.

To add inputs and probabilities to the singleton prediction query

1. Click the **Design** button in the corner of the Prediction Query Builder to switch back to the query building grid.
2. In the **Singleton Query Input** dialog box, click the **Value** box next to **vAssoc Seq Line Items**, and then click the **(...)** button.
3. In the **Key column** pane, select **Touring Tire**, and then click **Add**.
4. In the grid, click the empty cell under **Source** and select **Prediction Function.** In the cell under **Field**, select **PredictAssociation**.
5. In the **Mining Model** pane, select the nested table `vAssocSeqLineItems`, and drag it into the grid, to the **Criteria/Argument** box for the **PredictAssociation** function. Type **,3** after the nested table name just as in the previous procedure.
6. In the **Nested Table Input** dialog box, select **Touring Tire Tube** in the **Key column** pane, and then click **Add**.

7. In the grid, in the row for the **PredictAssociation** function, click the **Criteria/Argument** box, and change the arguments to add the argument, INCLUDE_STATISTICS.

   The complete text in the **Criteria/Argument** box should be as follows:

   ```
   [Association].[v Assoc Seq Line Items], INCLUDE_STATISTICS, 3
   ```

8. Click the **Results** button.

The results in the nested table now change to show the predictions, together with support and probability. For more information about how to interpret these values, see Mining Model Content for Association Models (Analysis Services - Data Mining).

Model	$SUPPORT	$PROBABILITY	$ADJUSTEDPROBABILITY
Sport-100	4334	0.291…	0.252…
Water Bottle	2866	0.192…	0.175…
Patch Kit	2113	0.142…	0.132

Working with Results

When there are many nested tables in the results, you might want to flatten the results for easier viewing. To do this, you can manually modify the query and add the **FLATTENED** keyword.

To flatten nested rowsets in a prediction query

1. Click the **SQL** button in the corner of the Prediction Query Builder.

   The grid changes to an open pane where you can view and modify the DMX statement that was created by the Prediction Query Builder.

2. After the **SELECT** keyword, type **FLATTENED**.

   The complete text of the query should be as follows:

```
SELECT FLATTENED
  PredictAssociation([Association].[v Assoc Seq Line
Items],INCLUDE_STATISTICS,3)
FROM
  [Association]
NATURAL PREDICTION JOIN
(SELECT (SELECT 'Touring Tire' AS [Model]
  UNION SELECT 'Touring Tire Tube' AS [Model]) AS [v Assoc
Seq Line Items]) AS t
```

3. Click the **Results** button in the upper corner of the Prediction Query Builder.

Note that after you have manually edited a query, you will not be able to switch back to Design view without losing the changes. If you wish to save the query, you can copy the DMX statement that you created manually to a text file. When you change back to Design view, the query is reverted to the last version that was valid in Design view.

Creating Multiple Predictions

Suppose you want to know the best predictions for individual customers, based on past purchases. You can use external data as input to the prediction query, such as tables containing the customer ID and the most recent product purchases. The requirements are that the data tables be already defined as an Analysis Services data source view; moreover, the input data must contain case and nested tables like those used in the model. They need not have the same names, but the structure must be similar. For the purpose of this tutorial, you will use the original tables on which the model was trained.

To change the input method for the prediction query

1. In the **Mining Model** menu, select **Singleton Query** again, to clear the check mark.
2. An error message appears warning that your singleton query will be lost. Click **Yes**.

   The name of the input dialog box changes to **Select Input Table(s)**.

Because you are interested in creating a prediction query that provides Customer ID and a list of products as input, you will add the customer table as the case table, and the purchases table as the nested table. Then you will add prediction functions to create recommendations.

To create a prediction query using nested table inputs

1. In the Mining Model pane, select the Association Filtered model.
2. In the **Select Input Table(s)** dialog box, click **Select Case Table**.
3. In the **Select Table** dialog box, for **Data Source**, select AdventureWorksDW2008. In the **Table/View Name** list, select vAssocSeqOrders, and then click **OK**.

   The table vAssocSeqOrders is added to the pane.
4. In the **Select Input Table(s)** dialog box, click **Select Nested Table**.
5. In the **Select Table** dialog box, for **Data Source**, select AdventureWorksDW2008. In the **Table/View name** list, select vAssocSeqLineItems, and then click **OK**.

   The table vAssocSeqLineItems is added to the pane.
6. In the **Specify Nested Join** dialog box, drag the OrderNumber field from the case table and drop it onto the OrderNumber field in the nested table.

   You can also click **Add Relationship** and create the relationship by selecting

columns from a list.

7. In the **Specify Relationship** dialog box, verify that the OrderNumber fields are mapped correctly, and then click **OK**.
8. Click **OK** to close the **Specify Nested Join** dialog box.

   The case and nested tables are updated in the design pane to show the joins connecting the external data columns to the columns in the model. If the relationships are wrong, you can right-click the join line and select **Modify Connections** to edit the column mapping, or you can right-click the join line and select **Delete** to remove the relationship completely.
9. Add a new row to the grid. For **Source**, select **vAssocSeqOrders table**. For **Field**, select CustomerKey.
10. Add a new row to the grid. For **Source**, select **vAssocSeqOrders table**. For **Field**, select Region.
11. Add a new row to the grid. For **Source**, select **Prediction Function**, and for **Field**, select **PredictAssociation**.
12. Drag vAssocSeqLineItems, into the **Criteria/Argument** box of the **PredictAssociation** row. Click at the end of the **Criteria/Argument** box and then type the following text: **INCLUDE_STATISTICS,3**

    The complete text in the **Criteria/Argument** box should be: `[Association].[v Assoc Seq Line Items], INCLUDE_STATISTICS, 3`
13. Click the **Result** button to view the predictions for each customer.

You might try creating a similar prediction query on the multiple models, to see whether filtering changes the prediction results. For more information about creating predictions and other types of queries, see Querying an Association Model (Analysis Services - Data Mining).

See Also

Mining Model Content for Association Models (Analysis Services - Data Mining)

PredictAssociation (DMX)

How to: Create a Prediction Query

Lesson 4: Building a Sequence Clustering Scenario (Intermediate Data Mining Tutorial)

The marketing department of Adventure Works Cycles wants to understand how customers move through the Adventure Works Cycles Web site. The company suspects that there is a pattern to the order in which customers put products into their shopping baskets. They want to analyze the order of purchase sequences to learn how customers add related items to their baskets. They can then use this information to streamline the flow of the Web site so that it leads customers to purchase additional products.

After you complete the tasks in this lesson, you will have created a mining model that uses the Microsoft Sequence Clustering algorithm to predict the next item that customers will put into their shopping baskets. You will experiment with two versions of the model: one that analyzes only the order of products in the basket, and one that contains some additional customer demographics for clustering. Finally, you will use the models to create predictions that you can use to recommend products to customers.

To complete the tasks in the lesson, you will use the market basket mining structure that you created in Lesson 3: Building a Market Basket Scenario (Intermediate Data Mining Tutorial). This lesson contains the following tasks:

- Creating a Sequence Clustering Mining Model Structure
- Processing the Sequence Clustering Model
- Exploring the Sequence Clustering Models
- Creating and Modifying a Related Sequence Clustering Model (Intermediate Data Mining Tutorial)
- Creating Predictions on a Sequence Clustering Model (Intermediate Data Mining Tutorial)

Next Task in Lesson

Creating a Sequence Clustering Mining Model Structure

All Lessons

Lesson 1: Creating the Intermediate Data Mining Solution

Lesson 2: Forecasting Scenario (Intermediate Data Mining Tutorial)

Lesson 3: Market Basket Scenario (Intermediate Data Mining Tutorial)

Lesson 4: Sequence Clustering Scenario (Intermediate Data Mining Tutorial)

Lesson 5: Neural Network and Logistic Regression Scenario (Intermediate Data Mining Tutorial)

See Also

Data Mining Tutorial

Intermediate Data Mining Tutorial (Analysis Services - Data Mining)

Creating a Sequence Clustering Mining Model Structure (Intermediate Data Mining Tutorial)

The first step in creating a sequence clustering mining model is to use the Data Mining Wizard to create a new mining structure and a mining model based on the Microsoft Sequence Clustering algorithm.

You will use the same data source view that you used for the market basket analysis, but you will add a column that contains the **sequence** identifier. In this scenario, the sequence means the order in which the customer added items to the shopping basket.

You will also add some columns that are used in one of the models to group customers by demographics.

Procedures

To create a sequence clustering structure and model

1. In Solution Explorer in SQL Server Data Tools (SSDT), right-click **Mining Structures** and select **New Mining Structure**.
2. On the **Welcome to the Data Mining Wizard** page, click **Next**.
3. On the **Select the Definition Method** page, verify that **From existing relational database or data warehouse** is selected, and then click **Next**.
4. On the **Create the Data Mining Structure** page, verify that the option **Create mining structure with a mining model** is selected. Next, click the dropdown list for the option, **Which data mining technique do you want to use?**, and select **Microsoft Sequence Clustering**. Click **Next**.

   The **Select Data Source View** page appears. Under **Available data source views**, select **Orders**.

   Orders is the same data source view that you used for the market basket scenario. If you have not created this data source view, see Adding a Data Source View with Nested Tables (Intermediate Data Mining Tutorial).
5. Click **Next**.
6. On the **Specify Table Types** page, select the **Case** check box next to the **vAssocSeqOrders** table, and select the **Nested** check box next to the **vAssocSeqLineItems** table. Click **Next**.

   nNote

   If an error occurs when you select the **Case** or **Nested** check boxes, it may be that the join in the data source view is not correct. The nested table, **vAssocSeqLineItems**, must be connected to the case table, **vAssocSeqOrders,** by a many-to-one join. You can edit the relationship by right-clicking on the join line and then reversing the direction of the join. For more information, see Create/Edit Relationship Dialog Box (Analysis Services - Multidimensional Data).
7. On the **Specify the Training Data** page, choose the columns for use in the model by selecting a check box as follows:
   - **IncomeGroup** Select the **Input** check box.

     This column contains interesting information about the customers that you can use for clustering. You will use it in the first model and then ignore it in the second model.
   - **OrderNumber** Select the **Key** check box.

     This field will be used as the identifier for the case table, or **Key**. In general,

you should never use the key field of the case table as an input, because the key contains unique values that are not useful for clustering.

- **Region** Select the **Input** check box.

  This column contains interesting information about the customers that you can use for clustering. You will use it in the first model and then ignore it in the second model.

- **LineNumber** Select the **Key** and **Input** check boxes.

  The **LineNumber** field will be used as the identifier for the nested table, or **Sequence Key**. The key for a nested table must always be used for input.

- **Model** Select the **Input** and **Predictable** check boxes.

Verify that the selections are correct, and then click **Next**.

8. On the **Specify Columns' Content and Data Type** page, verify that the grid contains the columns, content types, and data types shown in the following table, and then click **Next**.

Tables/Columns	Content Type	Data Type
IncomeGroup	Discrete	Text
OrderNumber	Key	Text
Region	Discrete	Text
vAssocSeqLineItems		
Line Number	Key Sequence	Long
Model	Discrete	Text

9. On the **Create Testing Set** page, change the **Percentage of data for testing** to 20, and then click **Next**.
10. On the **Completing the Wizard** page, for the **Mining structure name**, type **Sequence Clustering with Region**.
11. For the **Mining model name**, type **Sequence Clustering with Region**.
12. Check the **Allow drill through** box, and then click **Finish**.

Next Task in Lesson

Processing the Sequence Clustering Model

See Also

Data Mining Designer

Microsoft Sequence Clustering Algorithm

Processing the Sequence Clustering Model

After you create a new mining structure, you must deploy the changes that you made to the data mining solution, and then process the structure. After processing of both the new structure and the mining model is complete, you can browse the mining model.

Processing is always required when you create a new data mining structure. However, if you add a new mining model to an existing structure, you can process just the mining model. In this scenario, because you have created a new mining structure and a new mining model, you must process both.

Procedures

To process the mining structure and model

1. On the **Mining Model** menu of SQL Server Data Tools (SSDT), select **Process Mining Structure and All Models**.
2. At the warning asking whether you want to build and deploy the project, click **Yes**.
3. In the **Process Mining Structure - Sequence Clustering with Region** dialog box, click **Run**.

   The **Process Progress** dialog box opens to display information about model processing. Processing of the new structure and model might take some time.
4. After processing is complete, click **Close** to exit the **Process Progress** dialog box.
5. Click **Close** again to exit the **Process Mining Structure - Sequence Clustering with Region** dialog box.

Next Task in Lesson

Exploring the Sequence Clustering Models (Data Mining Tutorial)

See Also

Data Mining Designer

Microsoft Sequence Clustering Algorithm

Processing Data Mining Objects

Exploring the Sequence Clustering Model (Intermediate Data Mining Tutorial)

Now that you have built the **Sequence Clustering with Region** model, you can explore it by using the Microsoft Sequence Clustering Viewer in the **Mining Model Viewer** tab of Data Mining Designer. The Microsoft Sequence Cluster Viewer contains five tabs: **Cluster Diagram**, **Cluster Profiles**, **Cluster Characteristics**, **Cluster Discrimination**, and **State Transitions**. For more information about how to use this viewer, see Viewing a Mining Model with the Microsoft Sequence Cluster Viewer.

- Cluster Diagram tab
- Cluster Profiles tab
- Cluster Characteristics tab
- Cluster Discrimination tab
- State Transitions tab
- Generic Content View

Cluster Diagram Tab

The **Cluster Diagram** tab graphically displays the clusters that the algorithm discovered in the database. The layout in the diagram represents the relationships of the clusters, with similar clusters grouped close together. By default, the shade of each node represents the density of all cases in the cluster: the darker the shade of the node, the more cases it contains. You can change the meaning of the shading of the nodes so that it represents support, within each cluster, for an attribute and a state.

You can also rename the clusters, to make it easier to identify and work with target clusters. For this tutorial, you will rename the cluster that has the highest percentage of customers from the Pacific region, and the cluster that has the most cases overall.

Note

The cases that are assigned to specific clusters might change when you reprocess the model, depending on the data and the model parameters. Also, if you rename clusters, the names will be lost when you reprocess the mining model.

To change the attribute used for highlighting clusters

1. In the **Shading Variable** list, select **Model**.
2. Select **Cycling Cap** in the **State** list.

   The diagram updates to show the concentration of the selected product in each of the clusters. The cluster that has the darkest shading contains the highest density of cycling caps. You can change the shading variable to use any any state of any input column.
3. In the **Shading Variable** list, select **Population**.

   When you change the shading variable to population, the diagram updates to compare the clusters by size. The cluster that has the darkest shading contains more cases than the other clusters.

To rename nodes in the model

1. Change **Shading Variable** to **Region**, and set **State** to **Pacific**.
2. Highlight the darkest node in the graph.
3. Right-click this cluster and select **Rename Cluster.**
4. Type the name **Pacific Cluster.**

5. Change the value of **Shading Variable** to **Population**.
6. In the updated graph, locate the darkest cluster, which should be the largest cluster. If you cannot tell by the shading which cluster is largest, pause the mouse over each cluster and view the ToolTip, and then choose the cluster that contains the most cases.
7. Right-click this cluster and select **Rename Cluster**. Type the new name, **Largest Cluster**.

You can drill through from the node that represents the cluster to view details of the cases that are in each cluster. This can be useful if you want to take action on the results of your analysis, such as sending e-mail to a customer. You can also browse the other attributes of the cases that you included in the structure but did not use in the model, such as Region and IncomeGroup. For more information about drilling through from mining models to the underlying cases, see Using Drillthrough on Mining Models and Mining Structures (Analysis Services - Data Mining).

To drill through to details from the Cluster diagram

1. Right-click **Pacific Cluster**, select **Drill Through**, and then select **Model and Structure columns**.

   The **Drill Through** dialog box opens. Columns that are not used in the model but that are available for querying are prefixed with **Structure**.

   You can see that this cluster contains mostly customers from the Pacific region, with only a few customers from other regions.
2. Click the plus sign in the nested column v Assoc Seq Line Items to view the sequence of items in a particular customer order.
3. Close the **Drill Through** dialog box.

   Note
   The **Play** button enables you to requery the data; however, requerying does not change the data that is displayed, unless the model has been dynamically updated in the background by some other process.

Back to Top

Cluster Profiles Tab

The **Cluster Profiles** tab displays the sequences that are in each cluster. The clusters are listed in individual columns to the right of the **States** column.

In the viewer, the **Model** row describes the overall distribution of items in a cluster, and the **Model.samples** row contains sequences of the items. Each line of the color sequences in each cell of the **Model.samples** row represents the behavior of a randomly selected user in the cluster.

Each color in an individual sequence histogram represents a product model. The Mining Legend shows you the sequences of products by using both color-coding and the

product model names. If you have added other columns to the model for clustering, such as Region or Income Group, the viewer will contain an additional row for each column that shows the distribution of these values within each cluster.

To view the sequences that are most common in a cluster

1. Right-click the **Model** row in the column for the cluster **Largest Cluster**, and select **Show Legend**.

   The **Color** column contains a shaded bar that indicates the frequency of items found in sequences. Each item is represented by a different color. The **Meaning** column lists the product model names for each color. The **Distribution** column tells you the percentage of cases that contained this item in a sequence.
2. Close the **Mining Legend**.
3. Right-click the **Model.samples** row in the column with the heading, **Population,** and select **Show Legend**.
4. Scan the list of sequences in the overall model**.**

   The Mining Legend lists the most common sequences first, so you can see that Mountain Tire Tube is the first item in many sequences. This means that a customer is very likely to put the Mountain Tire Tube in the shopping basket first.

To drill through to cases from the cluster viewer

1. Scroll down in the Attribute pane until you find the row for the **Region** attribute.

   The row contains a histogram for each cluster in the model, plus one additional histogram for **Population**, meaning the entire set of cases used in the model. A histogram is a bar with different colors in it, where each color represents an attribute, and the size of the colored section for that attribute represents the percentage of cases with that attribute.
2. Compare the histograms for the clusters that you renamed **Pacific Cluster** and **Largest Cluster**. Each cluster appears in a different column.

   Both look like solid colors, but the colors are different.
3. In the **Region** row, pause the mouse over the colored histogram for **Largest Cluster**.

   The ToolTip displays values that show the actual percentages of cases from each region.
4. Right-click the colored histogram in the **Region** row for **Pacific Cluster**, select **Drill Through**, and then select **Model Columns Only**.
5. Move the scroll bar to review all of the customers in this cluster.

   Again, from drilling through to the details you can see that the cluster contains mostly orders from the Pacific region but also a few from the North America and Europe regions.

6. Close the **Drill Through** dialog box.

Back to Top

Cluster Characteristics Tab

The **Cluster Characteristics** tab summarizes the transitions between states in a cluster by displaying bars that visually represent the importance of the attribute value for the selected cluster. The **Variables** column tells you what the model found to be important for the selected cluster or population: either a particular value or the relationship between values, known as *transition*. The **Values** column provides more detail about the value or transition, and the **Probability** column visually represents the weight of this attribute or transition.

To view the important attributes for a cluster

1. In the **Cluster** dropdown list, select **Pacific Cluster**.

   The list updates to show the characteristics of the cluster that you renamed **Pacific Cluster**. In this cluster, the most important characteristic is **Region**.
2. Pause the mouse over the shaded bar in the row for **Region**.

   The probability of the value being Pacific is very high. For more information about how to interpret these values, see Microsoft Sequence Clustering Algorithm Technical Reference (Analysis Services - Data Mining).
3. Look through the list of characteristics for the cluster until you find the first transition row.
4. A transition row contains the text Transition in the **Variables** column, and some combination of sequential attribute values in the **Value** column. The sequence can also contain starting points and missing values.

   For example, suppose the transition has the value, [Start] -> Road Tire Tube. This means that customers in this cluster frequently put the Road Tire Tube in their shopping basket first. This might signify that the product is a popular item that customers seek out first, or it might only indicate that the product is easy to find on the purchasing site.
5. Scroll through the list until you find the first transition that does not have **[Start]** or **missing** in it.

   For example, suppose you find the transition, **Touring Tire, Touring Tire Tube**. This means that customers in this cluster frequently purchased these items together, in exactly this order.
6. Pause the mouse over the shaded bar for this transition.

   The probability of this transition is displayed as a percentage.
7. In the **Cluster** dropdown list, select **Population (All)**.

   The list of attributes updates to show the characteristics of all orders used to create the model. In this mining model, the most important characteristic for

distinguishing between clusters is **Region**, with a value of **North America**.

After reviewing these tasks, you realize two things. The first is that you need a lot of data to obtain a meaningful number of combinations. For example, the sequences with the highest probabilities are likely to include a **[Start]** or **Missing** state.

The second is that there is a strong clustering effect on attributes for **Region**, which makes it more difficult to see the groups of sequences. Therefore, you decide to create another model that uses sequences only, and does not include the columns for region or income.

Back to Top

Cluster Discrimination Tab

The **Cluster Discrimination** tab helps you compare two clusters, to determine which attributes distinguish a particular cluster from another cluster. The tab contains four columns: **Variables**, **Values**, **Cluster 1**, and **Cluster 2**. You can choose any cluster to use as **Cluster 1** and **Cluster 2**.

The **Variables** column tells you the name of the attribute, which can either be a column name or combination of column name and the word **transition**. The **Values** column shows the exact value of the attribute or the transition. The shaded bars in the columns for **Cluster 1** and **Cluster 2** indicate the strength of the attribute in the clusters that you are comparing. The longer the bar, the more the cluster is likely to include cases with that attribute.

To compare two clusters by using the Cluster Discrimination tab

1. In the **Cluster Discrimination** tab, for **Cluster 1**, select **Pacific Cluster**.

   By default, the selection for **Cluster 2** changes to **Complement of Pacific Cluster**.

   The top attribute that distinguishes **Pacific Cluster** from all other cases is the region. Region is such a strong attribute for clustering that it obscures other attributes. To avoid this effect, try comparing several of the smaller clusters to each other. When you do so, the list of attributes changes and might include more transitions between models.

2. Locate a transition row, and pause the mouse over the shaded bar.

   The items in the **Values** column can include both states and transitions. The shading for each item indicates the discrimination score. To learn more about the meaning of different scores, see Mining Model Content for Sequence Clustering Models (Analysis Services - Data Mining).

Back to Top

State Transitions Tab

On the **State Transitions** tab, you can select a cluster and browse through its state transitions. If you select **Population (All)** from the cluster drop-down list, the diagram shows the distribution of states for the whole mining model.

Each node in the graph represents a state, or possible value, of the sequences that you are trying to analyze. The background color of the nodes represents the frequency of that state. Lines connect some states, indicating a transition between states. You can move the slider up or down to change the probability threshold for the transitions. Numbers are associated with some nodes, indicating the probability of the state.

To explore the relationships in the State Transition tab

1. In the **State Transitions** tab of the Mining Model viewer, select **Pacific Cluster** from the list of clusters. Ensure that the **Show Edge Labels** option is selected.

   The graph updates to show the transitions that are most common in this cluster.
2. Click any node that is connected by a line to another node.

   The graph is updated and highlights the related nodes. The numeric value next to the line indicates the probability of the transition.
3. Raise the slider up to **All Links**, to increase the number of transitions included in the graph.
4. Select **Population (All)** from **Cluster**.

   Note that when you load a different cluster, the graph resets to the default display settings, so the slider control is reset to the middle position.
5. Click the darkest node in the graph, which should be **Sport-100**.

   Note that there are no lines connecting this product to other products.
6. Raise the slider up one step, to increase the number of transitions included in the graph. Do not go all the way to **All Links** yet.

   The graph is updated by adding several more transitions to the graph, but none that include the Sport-100 model.
7. Move the slider control all the way to **All Links**. Click the Sport-100 node if it is not already selected.

   The graph updates to show many transitions that include the Sport-100 product. The direction of the arrow on the connecting line tells you whether the Sport-100 item was selected as the first item or the second item in the pair.
8. Clicking the node for Touring Tire and move the slider control back down to the middle position.

   At first, there are many transition lines connecting Touring Tire to other products, but when you raise the probability threshold, the less likely transitions are eliminated from the graph, leaving just the transition, Touring Tire > Touring Tire Tube. This transition means that if a customer puts a Touring Tire into the shopping basket, there is a strong probability that the customer will next put a

Touring Tire Tube into the basket.

Back to Top

Generic Content Tree Viewer

This viewer can be used for all models, regardless of the algorithm or model type. The **Microsoft Generic Content Tree Viewer** is available from the **Viewer** drop-down list.

A content tree is a representation of any mining model as a series of nodes, wherein each node represents learned knowledge about the training data. The node can contain a pattern, a set of rules, a cluster, or the definition of a range of dates that share some attributes. The exact content of the node differs depending on the algorithm and the predictable attribute, but the general representation of the content is the same.

You can expand each node to see increasing levels of detail, and copy the content of any node to the Clipboard. For more information, see Viewing Model Details with the Microsoft Generic Content Tree Viewer.

To view details for a sequence clustering model by using the Generic Content Tree Viewer

1. In the **Mining Model Viewer** tab, click the **Viewer** list, and select **Microsoft Generic Content Tree viewer**.
2. In the **Node Caption** pane, click **Pacific Cluster (1)**.

   The name for this node contains both the friendly name that you assigned to the cluster and the underlying node ID. You can use the node IDs to drill down into additional detail in the model.
3. Expand the first child node, named **Sequence level for cluster 1**.

   The sequence level node for a cluster contains details about the states and transitions that are included in that cluster. You can use these details, available in the NODE_DISTRIBUTION column, to explore the sequences and the states for each cluster or for the model as a while.
4. Continue to expand nodes and view the details in the HTML viewer pane.

For more information about the mining model content, and how to use the details in the viewer, see Mining Model Content for Sequence Clustering Models (Analysis Services - Data Mining).

Back to Top

Next Task in Lesson

Creating a Related Sequence Clustering Model (Intermediate Data Mining Tutorial)

See Also

Microsoft Sequence Clustering Algorithm

Querying a Sequence Clustering Model (Analysis Services - Data Mining)

Creating a Related Sequence Clustering Model (Intermediate Data Mining Tutorial)

Through your exploration of the sequence clustering model, you learned that other attributes such as Region or Income have a strong effect on the models; therefore, to understand the sequences better, you will create a related sequence clustering model and remove the attributes related to customer demographics.

In this task, you will create a copy of the regional sequence clustering model, and then remove from the model any columns that are not directly related to the sequences.

The new model will contain all the same columns as the mining model on which it is based. However, you do not need to remove the columns from the mining structure, only specify that the new mining model ignore the columns.

Procedures

To make a copy of the sequence clustering model

1. In SQL Server Data Tools (SSDT), in the Data Mining Designer, click the **Mining Models** tab.
2. Right-click the model you want to copy, and select **New Mining Model**.
3. In the **New Mining Model** dialog box, type a model name, and select Microsoft **Sequence Clustering**.

   For this tutorial, type the name **Sequence Clustering**.
4. Click **OK**.

To remove columns from the mining model

1. In the **Mining Model** tab, in the column for the new model named Sequence Clustering, click the row for the **Income Group** attribute, and select **Ignore**.
2. Repeat this step for the attribute **Region**.
3. Click the plus sign next to the table name, **v Assoc Seq Line Items**, to expand the table and view the columns from the nested table.

   The new model should have only the following columns:

   Order Number Key

   Line Number Key

   Model Predict

To process the new sequence clustering model

1. In the **Mining Model** tab, right-click the new model named **Sequence Clustering**, and select **Process Model**.

   Because the new simplified mining model is based on a structure that has already

been processed, you do not need to reprocess the structure. You can process just the new mining model.

2. Click **Yes** to deploy the updated data mining project to the server.
3. In the **Process Mining Model** dialog box, click **Run**.
4. Click **Close** to close the **Process Progress** dialog box, and then click **Close** again in the **Process Mining Model** dialog box.

Next Task in Lesson

Creating Predictions on a Sequence Clustering Model (Intermediate Data Mining Tutorial)

See Also

Processing Data Mining Objects

Creating Predictions on a Sequence Clustering Model (Intermediate Data Mining Tutorial)

After you understand the sequence clustering model better by browsing it in the viewer, you can create prediction queries by using Prediction Query Builder on the **Mining Model Prediction** tab in Data Mining Designer. To create a prediction, you first select the sequence clustering model, and then select the input data. For inputs, you can use either an external data source, or you can build a singleton query and provide values in a dialog box.

This lesson assumes that you are already familiar with how to use the prediction query builder and want to learn how to build queries that are specific to a sequence clustering model. For general information about how to use Prediction Query Builder, see Using the Prediction Query Builder to Create DMX Prediction Queries or the section of the Basic Data Mining tutorial, Creating Predictions (Basic Data Mining Tutorial).

Creating Predictions on the Regional Model

For this scenario, you will first create some singleton prediction queries, to get an idea of how predictions might be different by region.

To create a singleton query on a sequence clustering model

1. Click the **Mining Model Prediction** tab of Data Mining Designer.
2. In the **Mining Model** column menu, select **Singleton Query**.

   The **Mining Model** pane and **Singleton Query Input** pane appear.
3. In the **Mining Model** pane, click **Select Model**. (You can skip this step if the sequence clustering mode is already selected.)

   The **Select Mining Model** dialog box opens.
4. Expand the node that represents the mining structure **Sequence Clustering with Region**, and select the model **Sequence Clustering with Region**. Click **OK**. For

now, ignore the input pane; you will specify the inputs after you have set up the prediction functions.

5. In the grid, click the empty cell under **Source** and select **Prediction Function.** In the cell under **Field**, select **PredictSequence**.

   Note
   You can also use the **Predict** function. If you do, be sure to choose the version of the **Predict** function that takes a table column as argument..

6. In the **Mining Model** pane, select the nested table `v Assoc Seq Line Items`, and drag it into the grid, to the **Criteria/Argument** box for the **PredictSequence** function.

   Dragging and dropping table and column names enables you to build complex statements without syntax errors. However, it replaces the current contents of the cell, which include other optional arguments for the **PredictSequence** function. To view the other arguments, you can temporarily add a second instance of the function to the grid for reference.

7. Click the **Result** button in the upper corner of the Prediction Query Builder.

The expected results contain a single column with the heading **Expression**. The **Expression** column contains a nested table with three columns as follows:

$SEQUENCE	Line Number	Model
1		Mountain-200

What do these results mean? Remember that you did not specify any inputs. Therefore, the prediction is made against the entire population of cases, and Analysis Services returns the most likely prediction overall.

Adding Inputs to a Singleton Prediction Query

Until now, you have not specified any inputs. In the next task, you will use the **Singleton Query Input** pane to specify some inputs to the query. First, you will use [Region] as an input to the regional sequence clustering model, to determine whether the predicted sequences are the same for all regions. You will then learn how to modify the query to add the probability for each prediction, and flatten the results to make them easier to view.

To generate predictions for a specific customer group

1. Click the **Design** button in the upper left-hand corner of the Prediction Query Builder to switch back to the query building grid.
2. In the **Singleton Query Input** dialog box, click the **Value** box for **Region**, and

select **Europe**.

3. Click the **Result** button to view predictions for customers in Europe.
4. Click the **Design** button in the upper left-hand corner of the Prediction Query Builder to switch back to the query building grid.
5. In the **Singleton Query Input** dialog box, click the **Value** box for **Region**, and select **North America**.
6. Click the **Result** button to view predictions for customers in North America.

Adding Probabilities by Using a Custom Expression

To output the probability for each prediction is slightly more complicated, because the probability is an attribute of the prediction and is output as a nested table. If you are familiar with Data Mining Extensions (DMX), you can easily alter the query to add a sub-select statement on the nested table. However, you can also create a sub-select statement in the Prediction Query Builder by adding a custom expression.

To output probabilities for a predicted sequence by using a custom expression

1. Click the **Design** button in the upper left-hand corner of the Prediction Query Builder to switch back to the query building grid.
2. In the grid, under **Source**, click a new row, and select **Custom Expression**.
3. Leave the box under **Field** blank.
4. For **Alias**, type **t**.
5. In the **Criteria/Argument** box, type the complete sub-select statement as shown in the following code sample. Be sure to include the starting and ending parentheses.

```
(SELECT PredictProbability([Model]) FROM
PredictSequence([Sequence Clustering with Region].[v Assoc
Seq Line Items]))
```

6. Click the **Result** button to view predictions for customers in Europe.

The results now contain two nested tables, one with the prediction, and one with the probability for the prediction. If the query does not work, you can switch to query design view and review the complete query statement, which should be as follows:

```
SELECT
  PredictSequence([Sequence Clustering with Region].[v Assoc Seq Line
Items]),
  ( (SELECT PredictProbability([Model]) FROM PredictSequence([Sequence
Clustering with Region].[v Assoc Seq Line Items]))) as [t]
FROM
  [Sequence Clustering with Region]
```

```
NATURAL PREDICTION JOIN
(SELECT 'Europe' AS [Region]) AS t
```

Working with Results

When there are many nested tables in the results, you might want to flatten the results for easier viewing. To do this, you can manually modify the query and add the **FLATTENED** keyword.

To flatten nested rowsets in a prediction query

1. Click the **Query** button in the corner of the Prediction Query Builder.
   The grid changes to an open pane where you can view and modify the DMX statement that was created by the Prediction Query Builder.
2. After the **SELECT** keyword, type **FLATTENED**.
   The complete text of the query should be similar to the following:

```
SELECT FLATTENED
  PredictSequence([Sequence Clustering with Region].[v Assoc
Seq Line Items]),
  ( (SELECT PredictProbability([Model]) FROM
PredictSequence([Sequence Clustering with Region].[v Assoc
Seq Line Items]))) as [t]
FROM
  [Sequence Clustering with Region]
NATURAL PREDICTION JOIN
(SELECT 'Europe' AS [Region]) AS t
```

3. Click the **Results** button in the upper corner of the Prediction Query Builder.

After you have manually edited a query, you will not be able to switch back to Design view without losing the changes. You can, however, save the DMX statement that you created manually to a text file, and then change back to Design view. When you do so, the query is reverted to the last version that was valid in Design view.

Creating Predictions on the Related Model

The previous examples used a case table column, Region, as the input to the singleton prediction query, because you were interested in knowing whether the model had found any differences between regions. However, after exploring the model, you decided that the differences are not strong enough to justify customizing product recommendations by region. What you are really interested in predicting is the items that customers select. Therefore, in the queries that follow, you will use the sequence clustering model that does not include Region, to generate recommendations for all customers.

Using Nested Table Columns as Input

First you will create a singleton prediction query that takes a single item as input and returns the next most likely item. To get a prediction of this kind, you have to use a nested table column as the input value. This is because the attribute that you are predicting, Model, is part of a nested table. Analysis Services provides the **Nested Table Input** dialog box to help you easily create prediction queries on nested table attributes, by using the Prediction Query Builder.

To use a nested table as input to a prediction

1. Click the **Design** button in the upper left-hand corner of the Prediction Query Builder to switch back to the query building grid.
2. In the **Singleton Query Input** dialog box, click the **Value** box for **Region**, and select the empty row to clear the input for this field.
3. In the **Singleton Query Input** dialog box, click the **Value** box for **vAssocSeqLineItems**, and then click the (...) button.
4. In the **Nested Table Input** dialog box, click **Add**.
5. In the new row, click the box under **Model**, and select Touring Tire from the list. Click **OK**.
6. Click the **Result** button to view the predictions.

The model recommends the following next items for all customers who choose the Touring Tire as the first item. You already know from exploring the model that customers frequently purchase the products Touring Tire and Touring Tire Tube together, so these recommendations look good.

$SEQUENCE	Line Number	Model
1		Touring Tire Tube
2		Sport-100
3		Long-Sleeve Logo Jersey

Creating a Bulk Prediction Query using Nested Table Inputs

Now that you are satisfied that the model creates the kind of predictions that you can use in making recommendations, you will create a prediction query that is mapped to an external data source. That data source will provide values representing current products. Because you are interested in creating a prediction query that provides Customer ID and a list of products as input, you will add the customer table as the case table, and the purchases table as the nested table. Then you will add prediction functions as you did previously to create recommendations.

This is the same procedure that you use to create predictions for the market basket scenario in Lesson 3; however, in a sequence clustering model predictions also need the order as input.

To create a prediction query using nested table inputs

1. In the **Mining Model** pane, select the Sequence Clustering model, if it is not already selected.
2. In the **Select Input Table(s)** dialog box, click **Select Case Table**.
3. In the **Select Table** dialog box, for Data Source, select Orders. In the **Table/View Name** list, select vAssocSeqOrders, and then click **OK**.
4. In the **Select Input Table(s)** dialog box, click **Select Nested Table**.
5. In the **Select Table** dialog box, for **Data Source**, select Orders. In the **Table/View name** list, select vAssocSeqLineItems, and then click **OK**.

   Analysis Services will try to detect relationships and create them automatically if the data types match and the column names are similar. If the relationships that it creates are wrong, you can right-click the join line and select **Modify Connections** to edit the column mapping, or you can right-click the join line and select **Delete** to remove the relationship completely. In this case, because the tables were already joined in the data source view, those relationships are automatically added to the design pane.
6. Add a new row to the grid. For **Source**, select vAssocSeqOrders, and for **Field**, select CustomerKey.
7. Add a new row to the grid. For **Source**, select **Prediction Function**, and for **Field**, select **PredictSequence**.
8. Drag vAssocSeqLineItems, into the **Criteria/Argument** box. Click at the end of the **Criteria/Argument** box and then type the following arguments: **2**.

   The complete text in the **Criteria/Argument** box should be: `[Sequence Clustering].[v Assoc Seq Line Items],2`
9. Click the **Result** button to view the predictions for each customer.

You have completed the tutorial on sequence clustering models.

Next Steps

If you have finished all the sections in the Intermediate Data Mining Tutorial (Analysis Services - Data Mining), the next step might be to learn to use Data Mining Extensions (DMX) statements to build models and generate predictions. For more information, see Creating and Querying Data Mining Models with DMX: Tutorials (Analysis Services - Data Mining).

If you are familiar with programming concepts, you can also use Analysis Management Objects (AMO) to programmatically work with data mining objects. For more information, see AMO Data Mining Classes.

See Also

Querying a Sequence Clustering Model (Analysis Services - Data Mining)

Mining Model Content for Sequence Clustering Models (Analysis Services - Data Mining)

Lesson 5: Building Neural Network and Logistic Regression Models (Intermediate Data Mining Tutorial)

The Operations department of Adventure Works is engaged in a project to improve customer satisfaction with their call center. They hired a vendor to manage the call center and to report metrics on call center effectiveness, and have asked you to analyze some preliminary data provided by the vendor. They want to know if there are any interesting findings. In particular, they would like to know if the data suggests any staffing problems with staffing or ways to improve customer satisfaction.

The data set is small and covers only a 30-day period in the operation of the call center. The data tracks the number of new and experienced operators in each shift, the number of incoming calls, the number of orders as well as issues that must be resolved, and the average time a customer waits for someone to respond to a call. The data also includes a service quality metric based on *abandon rate*, which is an indicator of customer frustration.

Because you do not have any prior expectations about what the data will show, you decide to use a neural network model to explore possible correlations. Neural network models are often used for exploration because they can analyze complex relationships between many inputs and outputs.

What You Will Learn

In this lesson, you will use the neural network algorithm to build a model that you and the Operations team can use to understand the trends in the data. As part of this lesson, you will try to answer the following questions:

- What factors affect customer satisfaction?
- What can the call center do to improve service quality?

Based on the results, you will then build a logistic regression model that you can use for predictions. The predictions will be used by the Operations team as an aid in planning call center operation.

This lesson contains the following topics:

- Adding a Data Source View for Call Center Data (Intermediate Data Mining Tutorial)
- Creating a Neural Network Structure and Model (Intermediate Data Mining Tutorial)
- Exploring the Call Center Model (Intermediate Data Mining Tutorial)

- Adding a Logistic Regression Model to the Call Center Structure (Intermediate Data Mining Tutorial)
- Creating Predictions for the Call Center Models (Intermediate Data Mining Tutorial)

Next Task in Lesson

Adding a Data Source View with Call Center Data

All Lessons

Lesson 1: Creating the Intermediate Data Mining Solution

Lesson 2: Forecasting Scenario (Intermediate Data Mining Tutorial)

Lesson 3: Market Basket Scenario (Intermediate Data Mining Tutorial)

Lesson 4: Sequence Clustering Scenario (Intermediate Data Mining Tutorial)

Lesson 5: Neural Network and Logistic Regression Scenario (Intermediate Data Mining Tutorial)

See Also

Data Mining Tutorial

Intermediate Data Mining Tutorials (Analysis Services - Data Mining)

Adding a Data Source View for Call Center Data (Intermediate Data Mining Tutorial)

In this task, you add a data source view that will be used to access the call center data. The same data will be used to build both the initial neural network model for exploration, and the logistic regression model that you will use to make recommendations.

You will also use the Data Source View Designer to add a column for the day of the week. That is because, although the source data tracks call center data by dates, your experience tells you that there are recurring patterns both in terms of call volume and service quality, depending on whether the day is a weekend or a weekday.

Procedures

To add a data source view

1. In **Solution Explorer**, right-click **Data Source Views**, and select **New Data Source View**.

   The Data Source View Wizard opens.
2. On the **Welcome to the Data Source View Wizard** page, click **Next**.
3. On the **Select a Data Source** page, under **Relational data sources**, select the Adventure Works DW Multidimensional 2012 data source. If you do not have this data source, see Basic Data Mining Tutorial. Click **Next**.

4. On the **Select Tables and Views** page, select the following table and then click the right arrow to add it to the data source view:
   - **FactCallCenter (dbo)**
   - **DimDate**
5. Click **Next**.
6. On the **Completing the Wizard** page, by default the data source view is named Adventure Works DW Multidimensional 2012 . Change the name to **CallCenter**, and then click **Finish**.

   Data Source View Designer opens to display the **CallCenter** data source view.
7. Right-click inside the Data Source View pane, and select **Add/Remove Tables**. Select the table, **DimDate** and click **OK**.

   A relationship should be automatically added between the **DateKey** columns in each table. You will use this relationship to get the column, **EnglishDayNameOfWeek**, from the **DimDate** table and use it in your model.
8. In the Data Source View designer, right-click the table, **FactCallCenter**, and select **New Named Calculation**.

   In the **Create Named Calculation** dialog box, type the following values:

Column name	DayOfWeek
Description	Get day of week from DimDate table
Expression	`(SELECT EnglishDayNameOfWeek AS DayOfWeek FROM DimDate where FactCallCenter.DateKey = DimDate.DateKey)`

   To verify that the expression creates the data you need, right-click the table **FactCallCenter**, and then select **Explore Data**.
9. Take a minute to review the data that is available, so that you can understand how it is used in data mining:

Column name	Contains
FactCallCenterID	An arbitrary key created when the data was imported to the data warehouse. This column identifies unique records and should be used as the case key for the data mining model.

DateKey	The date of the call center operation, expressed as an integer. Integer date keys are often used in data warehouses, but you might want to obtain the date in date/time format if you were going to group by date values. Note that dates are not unique because the vendor provides a separate report for each shift in each day of operation.
WageType	Indicates whether the day was a weekday, a weekend, or a holiday. It is possible that there is a difference in quality of customer service on weekends vs. weekdays so you will use this column as an input.
Shift	Indicates the shift for which calls are recorded. This call center divides the working day into four shifts: AM, PM1, PM2, and Midnight. It is possible that the shift influences the quality of customer service so you will use this as an input.
LevelOneOperators	Indicates the number of Level 1 operators on duty. Call center employees start at Level 1 so these employees are less experienced.
LevelTwoOperators	Indicates the number of Level 2 operators on duty. An employee must log a certain number of service hours to qualify as a Level 2 operator.
TotalOperators	The total number of operators present during the shift.
Calls	Number of calls received during the shift.
AutomaticResponses	The number of calls that were handled entirely by automated call processing (Interactive Voice Response, or IVR).

Orders	The number of orders that resulted from calls.
IssuesRaised	The number of issues requiring follow-up that were generated by calls.
AverageTimePerIssue	The average time required to respond to an incoming call.
ServiceGrade	A metric that indicates the general quality of service, measured as the *abandon rate* for the entire shift. The higher the abandon rate, the more likely it is that customers are dissatisfied and that potential orders are being lost.

Note that the data includes four different columns that are based on a single date column: **WageType**, **DayOfWeek**, **Shift**, and **DateKey**. Ordinarily in data mining it is not a good idea to use multiple columns that are derived from the same data, as the values correlate with each other too strongly and can obscure other patterns.

However, we will not use **DateKey** in the model because it contains too many unique values. There is no direct relationship between **Shift** and **DayOfWeek**, and **WageType** and **DayOfWeek** are only partly related. If you were worried about collinearity, you could create the structure using all of the available columns, and then ignore different columns in each model and test the effect.

Next Task in Lesson

Creating a Neural Network Structure and Model

See Also

Designing Data Source Views (Analysis Services)

Creating a Neural Network Structure and Model (Intermediate Data Mining Tutorial)

To create a data mining model, you must first use the Data Mining Wizard to create a new mining structure based on the new data source view. In this task you will use the wizard to create a mining structure, and at the same time create an associated mining model that is based on the Microsoft Neural Network algorithm.

Because neural networks are extremely flexible and can analyze many combinations of inputs and outputs, you should experiment with several ways of processing the data to get the best results. For example, you might want to customize the way that the

numerical target for service quality is *binned*, or grouped, to target specific business requirements. To do this, you will add a new column to the mining structure that groups numerical data in a different way, and then create a model that uses the new column. You will use these mining models to do some exploration.

Finally, when you have learned from the neural network model which factors have the greatest impact for your business question, you will build a separate model for prediction and scoring. You will use the Microsoft Logistic Regression algorithm, which is based on the neural networks model but is optimized for finding a solution based on specific inputs.

Steps

1. Create the basic mining structure, using defaults
2. Create a copy of the predictable column and modify it by binning the values
3. Add a new model and use the new column as the output for that model
4. Create an alias for the modified predictable attribute
5. Assign a seed so that models are processed the same way; process both models

Creating the Default Call Center Structure

To create the default neural network mining structure and model

1. In Solution Explorer in SQL Server Data Tools (SSDT), right-click **Mining Structures** and select **New Mining Structure**.
2. On the **Welcome to the Data Mining Wizard** page, click **Next**.
3. On the **Select the Definition Method** page, verify that **From existing relational database or data warehouse** is selected, and then click **Next**.
4. On the **Create the Data Mining Structure** page, verify that the option **Create mining structure with a mining model** is selected.
5. Click the dropdown list for the option **Which data mining technique do you want to use?**, then select **Microsoft Neural Networks**.

   Because the logistic regression models are based on the neural networks, you can reuse the same structure and add a new mining model.
6. Click **Next**.

   The **Select Data Source View** page appears.
7. Under **Available data source views**, select **Call Center**, and click **Next**.
8. On the **Specify Table Types** page, select the **Case** check box next to the **FactCallCenter** table. Do not select anything for **DimDate**. Click **Next**.
9. On the **Specify the Training Data** page, select **Key** next to the column **FactCallCenterID.**
10. Select the **Predict** and **Input** check boxes.
11. Select the **Key**, **Input**, and **Predict** check boxes as shown in the following table:

Tables/Columns	Key/Input/Predict
AutomaticResponses	Input
AverageTimePerIssue	Input/Predict
Calls	Input
DateKey	Do not use
DayOfWeek	Input
FactCallCenterID	Key
IssuesRaised	Input
LevelOneOperators	Input/Predict
LevelTwoOperators	Input
Orders	Input/Predict
ServiceGrade	Input/Predict
Shift	Input
TotalOperators	Do not use
WageType	Input

Note that multiple predictable columns have been selected. One of the strengths of the neural network algorithm is that it can analyze all possible combinations of input and output attributes. You wouldn't want to do this for a large data set, as it could exponentially increase processing time..

12. On the **Specify Columns' Content and Data Type** page, verify that the grid contains the columns, content types, and data types as shown in the following table, and then click **Next**.

Columns	Content Type	Data Types
AutomaticResponses	Continuous	Long
AverageTimePerIssue	Continuous	Long
Calls	Continuous	Long
DayOfWeek	Discrete	Text
FactCallCenterID	Key	Long

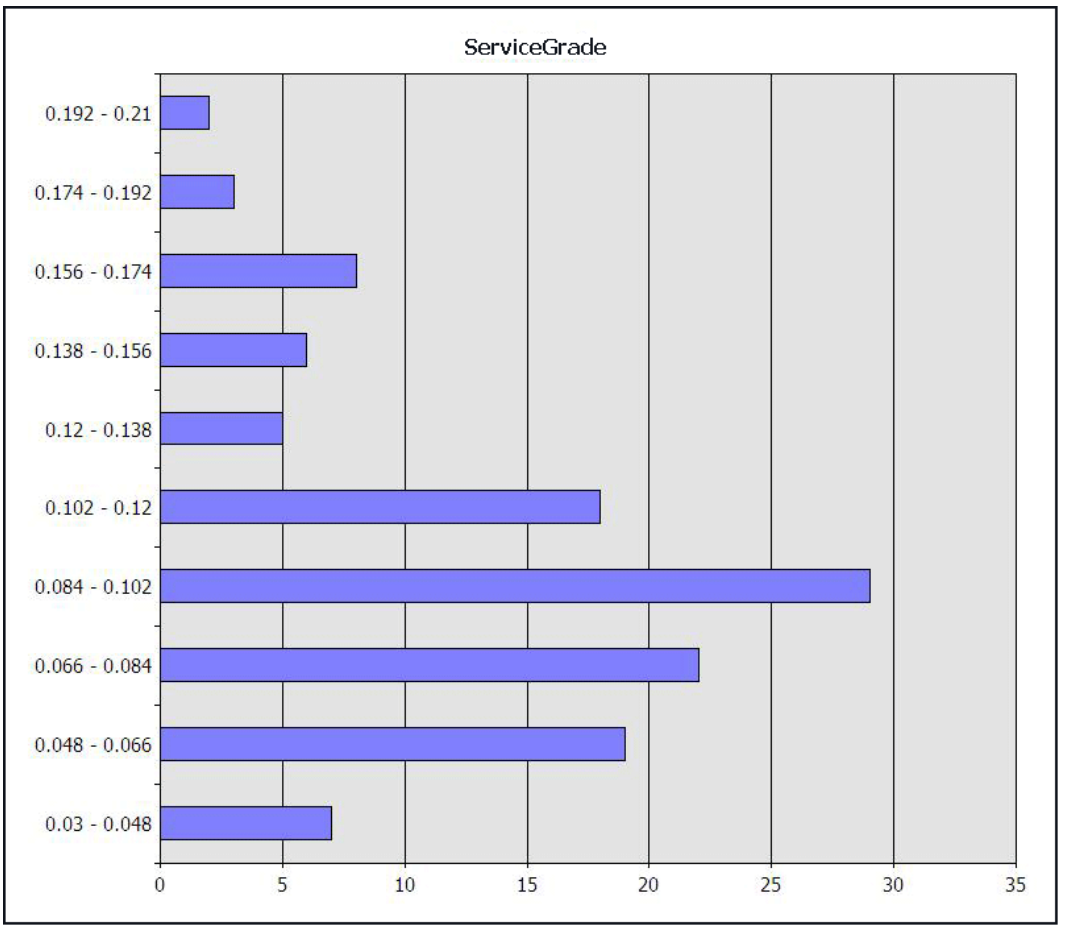
IssuesRaised	Continuous	Long
LevelOneOperators	Continuous	Long
LevelTwoOperators	Continuous	Long
Orders	Continuous	Long
ServiceGrade	Continuous	Double
Shift	Discrete	Text
WageType	Discrete	Text

13. On the **Create testing set** page, clear the text box for the option, **Percentage of data for testing**. Click **Next**.
14. On the **Completing the Wizard** page, for the **Mining structure name**, type **Call Center**.
15. For the **Mining model name**, type **Call Center Default NN**, and then click **Finish**.

    The **Allow drill through** box is disabled because you cannot drill through to data with neural network models.
16. In Solution Explorer, right-click the name of the data mining structure that you just created, and select **Process**.

Understanding Discretization

By default, when you create a neural network model that has a numeric predictable attribute, the Microsoft Neural Network algorithm treats the attribute as a continuous number. For example, the ServiceGrade attribute is a number that theoretically ranges from 0.00 (all calls are answered) to 1.00 (all callers hang up). In this data set, the values have the following distribution:



As a result, when you process the model the outputs might be grouped differently than you expect. For example, if you use clustering to identify the best groups of values, the algorithm divides the values in ServiceGrade into ranges such as this one: 0.0748051948 - 0.09716216215. Although this grouping is mathematically accurate, such ranges might not be as meaningful to business users. To group the numerical values differently, you can create a copy or multiple copies of the numerical data column and specify how the data mining algorithm should process the values. For example, you might specify that the algorithm divide the values into no more than five bins.

Analysis Services provides a variety of methods for binning or processing numerical data. The following table illustrates the differences between the results when the output attribute ServiceGrade has been processed three different ways:

- Treating it as a continuous number.
- Having the algorithm use clustering to identify the best arrangement of values.
- Specifying that the numbers be binned by the Equal Areas method.

Default model (continuous)		Binned by clustering		Binned by equal areas	
VALUE	**SUPPORT**	**VALUE**	**SUPPORT**	**VALUE**	**SUPPORT**
Missing	0	< 0.0748051948	34	< 0.07	26
0.09875	120	0.0748051948 - 0.09716216215	27	0.07 - 0.00	22
		0.09716216215 - 0.13297297295	39	0.09 - 0.11	36
		0.13297297295 - 0.167499999975	10	>= 0.12	36
		>= 0.167499999975	10		

Note
You can obtain these statistics from the marginal statistics node of the model, after all the data has been processed. For more information about the marginal statistics node, see Mining Model Content for Neural Network Models (Analysis Services - Data Mining).

In this table, the VALUE column shows you how the number for ServiceGrade has been handled. The SUPPORT column shows you how many cases had that value, or that fell in that range.

1. **Use continuous numbers (default)**

   If you used the default method, the algorithm would compute outcomes for 120 distinct values, the mean value of which is 0.09875. You can also see the number of missing values.

2. **Bin by clustering**

   When you let the Microsoft Clustering algorithm determine the optional grouping of values, the algorithm would group the values for ServiceGrade into five (5) ranges. The number of cases in each range is not evenly distributed, as you can see from the support column.

3. **Bin by equal areas**

   When you choose this method, the algorithm forces the values into buckets of equal size, which in turn changes the upper and lower bounds of each range. You can

specify the number of buckets, but you want to avoid having two few values in any bucket.

For more information about binning options, see Discretization Methods (Data Mining).

Alternatively, rather than using the numeric values, you could add a separate derived column that classifies the service grades into predefined target ranges, such as **Best** (ServiceGrade <= 0.05), **Acceptable** (0.10 > ServiceGrade > 0.05), and **Poor** (ServiceGrade >= 0.10).

Creating a Copy of a Column and Changing the Discretization Method

In Analysis Services data mining, you can easily change the way that numerical data is binned within a mining structure by adding a copy of the column containing the target data and changing the discretization method.

The following procedure describes how to make a copy of the mining column that contains the target attribute, ServiceGrade. You can create multiple copies of any column in a mining structure, including the predictable attribute.

You will then customize the grouping of the numeric values in the copied column, to reduce the complexity of the groupings. For this tutorial, you will use the Equal Areas method of discretization, and specify four buckets. The groupings that result from this method are fairly close to the target values of interest to your business users.

Note

During initial exploration of data, you can also experiment with various discretization methods, or try clustering the data first.

To create a customized copy of a column in the mining structure

1. In Solution Explorer, double-click the mining structure that you just created.
2. In the Mining Structure tab, click **Add a mining structure column**.
3. In the **Select column** dialog box, select ServiceGrade from the list in **Source column**, then click **OK**.

   A new column is added to the list of mining structure columns. By default, the new mining column has the same name as the existing column, with a numerical postfix: for example, ServiceGrade 1. You can change the name of this column to be more descriptive.

   You will also specify the discretization method.
4. Right-click ServiceGrade 1 and select **Properties**.
5. In the **Properties** window, locate the **Name** property, and change the name to **Service Grade Binned** .
6. A dialog box appears asking whether you want to make the same change to the name of all related mining model columns. Click **No**.
7. In the **Properties** window, locate the section **Data Type** and expand it if

necessary.

8. Change the value of the property **Content** from **Continuous** to **Discretized**.

   The following properties are now available. Change the values of the properties as shown in the following table:

Property	Default value	New value
DiscretizationMethod	**Continuous**	**EqualAreas**
DiscretizationBucketCount	No value	4

   Note
   The default value of **P:Microsoft.AnalysisServices.ScalarMiningStructureColumn.DiscretizationBucketCount** is actually 0, which means that the algorithm automatically determines the optimum number of buckets. Therefore, if you want to reset the value of this property to its default, type 0.

9. In Data Mining Designer, click the **Mining Models** tab.

   Notice that when you add a copy of a mining structure column, the usage flag for the copy is automatically set to **Ignore**. Usually, when you add a copy of a column to a mining structure, you would not use the copy for analysis together with the original column, or the algorithm will find a strong correlation between the two columns that might obscure other relationships.

Adding a New Mining Model to the Mining Structure

Now that you have created a new grouping for the target attribute, you need to add a new mining model that uses the discretized column. When you are done, the CallCenter mining structure will have two mining models:

- The mining model, Call Center Default NN, handles the ServiceGrade values as a continuous range.
- You will create a new mining model, Call Center Binned NN, that uses as its target outcomes the values of the ServiceGrade column, distributed into four buckets of equal size.

To add a mining model based on the new discretized column

1. In Solution Explorer, right-click the mining structure that you just created, and select **Open**.
2. Click the **Mining Models** tab.
3. Click **Create a related mining model**.
4. In the **New Mining Model** dialog box, for **Model name**, type **Call Center**

Binned NN. In the **Algorithm name** dropdown list, select **Microsoft Neural Network**.

5. In the list of columns contained in the new mining model, locate ServiceGrade, and change the usage from **Predict** to **Ignore**.
6. Similarly, locate ServiceGrade Binned, and change the usage from **Ignore** to **Predict**.

Ordinarily you cannot compare mining models that use different predictable attributes. However, you can create an alias for a mining model column. That is, you can rename the column, ServiceGrade Binned, within the mining model so that it has the same name as the original column. You can then directly compare these two models in an accuracy chart, even though the data is discretized differently.

To add an alias for a mining structure column in a mining model

1. In the **Mining Models** tab, under **Structure**, select ServiceGrade Binned.

   Note that the **Properties** window displays the properties of the object, ScalarMiningStructure column.
2. Under the column for the mining model, ServiceGrade Binned NN, click the cell corresponding to the column ServiceGrade Binned.

   Note that now the **Properties** window displays the properties for the object, MiningModelColumn.
3. Locate the **Name** property, and change the value to **ServiceGrade**.
4. Locate the **Description** property and type **Temporary column alias**.

   The **Properties** window should contain the following information:

Property	Value
Description	Temporary column alias
ID	ServiceGrade Binned
Modeling Flags	
Name	Service Grade
SourceColumn ID	Service Grade 1
Usage	Predict

5. Click anywhere in the **Mining Model** tab.

   The grid is updated to show the new temporary column alias, **ServiceGrade**, beside the column usage. The grid containing the mining structure and two mining models should look like the following:

Structure	Call Center Default NN	Call Center Binned NN
	Microsoft Neural Network	Microsoft Neural Network
AutomaticResponses	Input	Input
AverageTimePerIssue	Predict	Predict
Calls	Input	Input
DayOfWeek	Input	Input
FactCallCenterID	Key	Key
IssuesRaised	Input	Input
LevelOneOperators	Input	Input
LevelTwoOperators	Input	Input
Orders	Input	Input
ServceGrade Binned	Ignore	Predict (ServiceGrade)
ServiceGrade	Predict	Ignore
Shift	Input	Input
Total Operators	Input	Input
WageType	Input	Input

Processing the Model

Finally, to ensure that the models you have created can be easily compared, you will set the seed parameter for both the default and binned models. Setting a seed value guarantees that each model starts processing the data from the same point.

Note

If you do not specify a numeric value for the seed parameter, SQL Server Analysis Services will generate a seed based on the name of the model. Because the models always have different names, you must set a seed value to ensure that they process data in the same order.

To specify the seed and process the models

1. In the **Mining Model** tab, right-click the column for the model named Call Center - LR, and select **Set Algorithm Parameters**.
2. In the row for the HOLDOUT_SEED parameter, click the empty cell under **Value**, and type **1**. Click **OK**. Repeat this step for each model associated with the

structure.

> **Note**
> The value that you choose as the seed does not matter, as long as you use the same seed for all related models.

3. In the **Mining Models** menu, select **Process Mining Structure and All Models**. Click **Yes** to deploy the updated data mining project to the server.
4. In the **Process Mining Model** dialog box, click **Run**.
5. Click **Close** to close the **Process Progress** dialog box, and then click **Close** again in the **Process Mining Model** dialog box.

Now that you have created the two related mining models, you will explore the data to discover relationships in the data.

Next Task in Lesson

Exploring the Call Center Model (Intermediate Data Mining Tutorial)

See Also

Mining Structures (Analysis Services - Data Mining)

Exploring the Call Center Model (Intermediate Data Mining Tutorial)

Now that you have built the exploratory model, you can use it to learn more about your data by using the following tools provided in SQL Server Data Tools (SSDT).

- Microsoft Neural Network Viewer**:** This viewer is available in the **Mining Model Viewer** tab of Data Mining Designer, and is designed to help you experiment with interactions in the data.
- Microsoft Generic Content Tree Viewer**:** This standard viewer provides in-depth detail about the patterns and statistics discovered by the algorithm when it generated the model.

Microsoft Neural Network Viewer

The viewer has three panes — **Input**, **Output**, and **Variables**.

By using the **Output** pane, you can select different values for the predictable attribute, or dependent variable. If your model contains multiple predictable attributes, you can select the attribute from the **Output Attribute** list.

The **Variables** pane compares the two outcomes that you chose in terms of contributing attributes, or variables. The colored bars visually represent how strongly the variable affects the target outcomes. You can also view lift scores for the variables. A lift score is calculated differently depending on which mining model type you are using, but generally tells you the improvement in the model when using this attribute for prediction.

The **Input** pane lets you add influencers to the model to try out various what-if scenarios.

Using the Output Pane

In this initial model, you are interested in seeing how various factors affect the grade of service. To do this, you can select Service Grade from the list of output attributes, and then compare different levels of service by selecting ranges from the dropdown lists for **Value 1** and **Value 2**.

To compare lowest and highest service grades

1. For **Value 1**, select the range with the lowest values. For example, the range 0-0-0.7 represents the lowest abandon rates, and therefore the best level of service.

   Note
   The exact values in this range may vary depending on how you configured the model.

2. For **Value 2**, select the range with the highest values. For example, the range with the value >=0.12 represents the highest abandon rates, and therefore the worst service grade. In other words, 12% of the customers who phoned during this shift hung up before speaking to a representative.

   The contents of the **Variables** pane are updated to compare attributes that contribute to the outcome values. Therefore, the left column shows you the attributes that are associated with the best grade of service, and the right column shows you the attributes associated with the worst grade of service.

Using the Variables Pane

In this model, it appears that **Average Time Per Issue** is an important factor. This variable indicates the average time that it takes for a call to be answered, regardless of call type.

To view and copy probability and lift scores for an attribute

1. In the **Variables** pane, pause the mouse over the colored bar in the first row.

   This colored bar shows you how strongly **Average Time Per Issue** contributes toward the service grade. The tooltip shows an overall score, probabilities, and lift scores for each combination of a variable and a target outcome.

2. In the **Variables** pane, right-click any colored bar and select **Copy**.
3. In an Excel worksheet, right-click any cell and select **Paste**.

   The report is pasted as an HTML table, and shows only the scores for each bar.

4. In a different Excel worksheet, right-click any cell and select **Paste Special**.

   The report is pasted as text format, and includes the related statistics described in the next section.

Using the Input Pane

Suppose that you are interested in looking at the effect of a particular factor, such as the shift, or number of operators. You can select a particular variable by using the **Input** pane, and the **Variables** pane is automatically updated to compare the two previously selected groups given the specified variable.

To review the effect on service grade by changing input attributes

1. In the **Input** pane, for **attribute**, select Shift.
2. For **Value**, select **AM**.

   The **Variables** pane updates to show the impact on the model when the shift is **AM**. All other selections remain the same — you are still comparing the lowest and highest service grades.
3. For **Value**, select **PM1**.

   The **Variables** pane updates to show the impact on the model when the shift changes.
4. In the **Input** pane, click the next blank row under **Attribute**, and select Calls. For **Value**, select the range that indicates the greatest number of calls.

   A new input condition is added to the list. The **Variables** pane updates to show the impact on the model for a particular shift when the call volume is highest.
5. Continue to change the values for Shift and Calls to find any interesting correlations between shift, call volume, and service grade.

   Note
   To clear the **Input** pane so that you can use different attributes, click **Refresh viewer content**.

Interpreting the Statistics Provided in the Viewer

Longer waiting times are a strong predictor of a high abandon rate, meaning a poor service grade. This may seem an obvious conclusion; however, the mining model provides you with some additional statistical data to help you interpret these trends.

- **Score**: Value that indicates the overall importance of this variable for discriminating between outcomes. The higher the score, the stronger the effect the variable has on the outcome.
- **Probability of value 1**: Percentage that represents the probability of this value for this outcome.
- **Probability of value 2**: Percentage that represents the probability of this value for this outcome.
- **Lift for Value 1** and **Lift for Value 2**: Scores that represents the impact of using this particular variable for predicting the Value 1 and Value 2 outcomes. The higher the score, the better the variable is at predicting the outcomes.

The following table contains some example values for the top influencers. For example, the **Probability of value 1** is 60.6% and **Probability of value 2** is 8.30%, meaning that when the Average Time Per Issue was in the range of 44-70 minutes, 60.6% of cases were in the shift with the highest service grades (Value 1), and 8.30% of cases were in the shift with the worse service grades (Value 2).

From this information, you can draw some conclusions. Shorter call response time (the range of 44-70) strongly influences better service grade (the range 0.00-0.07). The score (92.35) tells you that this variable is very important.

However, as you look down the list of contributing factors, you see some other factors with effects that are more subtle and more difficult to interpret. For example, shift appears to influence service, but the lift scores and the relative probabilities indicate that shift is not a major factor.

Attribute	Value	Favors < 0.07	Favors >= 0.12
Average Time Per Issue	89.087 - 120.000		Score: 100 Probability of Value1: 4.45 % Probability of Value2: 51.94 % Lift for Value1: 0.19 Lift for Value2: 1.94
Average Time Per Issue	44.000 - 70.597	Score: 92.35 Probability of Value1: 60.06 % Probability of Value2: 8.30 % Lift for Value1: 2.61 Lift for Value2: 0.31	

Attribute	Value	Favors < 0.07	Favors >= 0.12

Back to Top

Microsoft Generic Content Tree Viewer

This viewer can be used to view even more detailed information created by the algorithm when the model is processed. The **Microsoft Generic Content Tree Viewer** represents the mining model as a series of nodes, wherein each node represents learned knowledge about the training data. This viewer can be used with all models, but the contents of the nodes are different depending in the model type.

For neural network models or logistic regression models, you might find the **marginal statistics node** particularly useful. This node contains derived statistics about the distribution of values in your data. This information can be useful if you want to get a summary of the data without having to write many T-SQL queries. The chart of binning values in the previous topic was derived from the marginal statistics node.

To obtain a summary of data values from the mining model

1. In Data Mining Designer, in the **Mining Model Viewer** tab, select <mining model name>.
2. From the **Viewer** list, select **Microsoft Generic Content Tree Viewer**.

   The view of the mining model refreshes to show a node hierarchy in the left-hand pane and an HTML table in the right-hand pane.
3. In the **Node Caption** pane, click the node that has the name 10000000000000000.

   The topmost node in any model is always the model root node. In a neural network or logistic regression model, the node immediately under that is the marginal statistics node.
4. In the **Node Details** pane, scroll down until you find the row, NODE_DISTRIBUTION.
5. Scroll down through the NODE_DISTRIBUTION table to view the distribution of values as calculated by the neural network algorithm.

To use this data in a report, you could select and then copy the information for specific rows, or you can use the following Data Mining Extensions (DMX) query to extract the complete contents of the node.

```
SELECT *
FROM [Call Center EQ4].CONTENT
```

```
WHERE NODE_NAME = '10000000000000000'
```

You can also use the node hierarchy and the details in the NODE_DISTRIBUTION table to traverse individual paths in the neural network and view statistics from the hidden layer. For more information, see Querying a Neural Network Model (Analysis Services- Data Mining).

Back to Top

Next Task in Lesson

Adding a Logistic Regression Model to the Call Center Structure (Intermediate Data Mining Tutorial)

See Also

Mining Model Content for Neural Network Models (Analysis Services - Data Mining)

Querying a Neural Network Model (Analysis Services- Data Mining)

Microsoft Neural Network Algorithm Technical Reference (Analysis Services - Data Mining)

How to: Change the Discretization of a Column in a Mining Model

Adding a Logistic Regression Model to the Call Center Structure (Intermediate Data Mining Tutorial)

In addition to analyzing the factors that might affect call center operations, you were also asked to provide some specific recommendations on how the staff can improve service quality. In this task, you will use the same mining structure that you used to build the exploratory model and add a mining model that will be used for creating predictions.

In Analysis Services, a logistic regression model is based on the neural networks algorithm, and therefore provides the same flexibility and power as a neural network model. However, logistic regression is particularly well-suited for predicting binary outcomes.

For this scenario, you will use the same mining structure that you used for the neural network model. However, you will customize the new model to target your business questions. You are interested in improving service quality and determining how many experienced operators you need, so you will set up your model to predict those values.

To ensure that all the models based on the call center data are as similar as possible, you will use the same seed value as before. Setting the seed parameter ensures that the model processes the data from the same starting point, and minimizes variations caused by artifacts in the data.

Procedures

To add a new mining model to the call center mining structure

1. In SQL Server Data Tools (SSDT), in Solution Explorer, right-click the mining

structure, **Call Center Binned**, and select **Open Designer**.

2. In Data Mining Designer, click the **Mining Models** tab.
3. Click **Create a related mining model**.
4. In the **New Mining Model** dialog box, for **Model name**, type **Call Center - LR**. For **Algorithm name**, select **Microsoft Logistic Regression**.
5. Click **OK**.

   The new mining model is displayed in the **Mining Models** tab.

To customize the logistic regression model

1. In the column for the new mining model, **Call Center - LR**, leave Fact CallCenter ID as the key.
2. Change the value of ServiceGrade and Level Two Operators to **Predict**.

   These columns will be used both as input and for prediction. In essence, you are creating two separate models on the same data: one that predicts the number of operators, and one that predicts the service grade.
3. Change all other columns to **Input**.

To specify the seed and process the models

1. In the **Mining Model** tab, right-click the column for the model named Call Center - LR, and select **Set Algorithm Parameters**.
2. In the row for the HOLDOUT_SEED parameter, click the empty cell under **Value**, and type **1**. Click **OK**.

   Note
   The value that you choose as the seed does not matter, as long as you use the same seed for all related models.
3. In the **Mining Models** menu, select **Process Mining Structure and All Models**. Click **Yes** to deploy the updated data mining project to the server.
4. In the **Process Mining Model** dialog box, click **Run**.
5. Click **Close** to close the **Process Progress** dialog box, and then click **Close** again in the **Process Mining Model** dialog box.

Next Task in Lesson

Creating Predictions for the Call Center Models (Intermediate Data Mining Tutorial)

See Also

Processing Data Mining Objects

Creating Predictions for the Call Center Models (Intermediate Data Mining Tutorial)

Now that you have learned something about the interactions between shifts, the number of operators, calls, and service grade, you are ready to create some prediction queries that can be used in business analysis and planning. You will first create some predictions on the exploratory model to test some assumptions. Next, you will create bulk predictions by using the logistic regression model.

This lesson assumes that you are already familiar with the concept of prediction queries.

Creating Predictions using the Neural Network Model

The following example demonstrates how to make a singleton prediction using the neural network model that was created for exploration. Singleton predictions are a good way to try out different values to see the effect in the model. In this scenario, you will predict the service grade for the midnight shift (no day of the week specified) if six experienced operators are on duty.

To create a singleton query by using the neural network model

1. In SQL Server Data Tools (SSDT), open the solution that contains the model that you want to use.
2. In Data Mining Designer, click the **Mining Model Prediction** tab.
3. In the **Mining Model** pane, click **Select Model**.
4. The **Select Mining Model** dialog box shows a list of mining structures. Expand the mining structure to view a list of mining models associated with that structure.
5. Expand the mining structure Call Center Default, and select the neural network model, Call Center - LR.
6. From the **Mining Model** menu, select **Singleton Query**.

   The **Singleton Query Input** dialog box appears, with columns mapped to the columns in the mining model.
7. In the **Singleton Query Input** dialog box, click the row for Shift, and then select midnight.
8. Click the row for Lvl 2 Operators, and type **6**.
9. In the bottom half of the **Mining Model Prediction** tab, click the first row in the grid.
10. In the **Source** column, click the down arrow, and select **Prediction function**. In the **Field** column, select **PredictHistogram**.

    A list of arguments that you can use with this prediction function automatically appears in the **Criteria/Arguments** box.
11. Drag the ServiceGrade column from the list of columns in the **Mining Model**

pane to the **Criteria/Arguments** box.

The name of the column is automatically inserted as the argument. You can choose any predictable attribute column to drag into this text box.

12. Click the button **Switch to query results view**, in the upper corner of the Prediction Query Builder.

The expected results contain the possible predicted values for each service grade given these inputs, together with support and probability values for each prediction. You can return to design view at any time and change the inputs, or add more inputs.

Creating Predictions by using a Logistic Regression Model

If you already know the attributes that are relevant to the business problem, you can use a logistic regression model to predict the effect of making changes in some attributes. Logistic regression is a statistical method that is commonly used to make predictions based on changes in independent variables: for example, it is used in financial scoring, to predict customer behavior based on customer demographics.

In this task, you will learn how to create a data source that will be used for predictions, and then make predictions to help answer several business questions.

Generating Data used for Bulk Prediction

There are many ways to provide input data: for example, you might import staffing levels from a spreadsheet, and run that data through the model to predict service quality for the next month.

In this lesson, you will use the Data Source View designer to create a named query. This named query is a custom Transact-SQL statement that for each shift on the schedule calculates the maximum number of operators on staff, the minimum calls received, and the average number of issues that are generated. You will then join that data to a mining model to make predictions about a series of upcoming dates.

To generate input data for a bulk prediction query

1. In Solution Explorer, right-click **Data Source Views**, and then select **New Data Source View**.
2. In the Data Source View wizard, select Adventure Works DW Multidimensional 2012 as the data source, and then click **Next**.
3. On the **Select Tables and Views** page, click **Next** without selecting any tables.
4. On the **Completing the Wizard** page, type the name, **Shifts**.

   This name will appear in Solution Explorer as the name of the data source view.
5. Right-click the empty design pane, then select **New Named Query**.
6. In the **Create Named Query** dialog box, for **Name**, type **Shifts for Call Center**.

   This name will appear in Data Source View designer only as the name of the named query.

7. Paste the following query statement into the SQL text pane in the lower half of the dialog box.

```
SELECT DISTINCT WageType, Shift,
AVG(Orders) as AvgOrders, MIN(Orders) as MinOrders,
MAX(Orders) as MaxOrders,
AVG(Calls) as AvgCalls, MIN(Calls) as MinCalls, MAX(Calls) as
MaxCalls,
AVG(LevelTwoOperators) as AvgOperators,
MIN(LevelTwoOperators) as MinOperators,
MAX(LevelTwoOperators) as MaxOperators,
AVG(IssuesRaised) as AvgIssues, MIN(IssuesRaised) as
MinIssues, MAX(IssuesRaised) as MaxIssues
FROM dbo.FactCallCenter
GROUP BY Shift, WageType
```

8. In the design pane, right-click the table, Shifts for Call Center, and select **Explore Data** to preview the data as returned by the T-SQL query.
9. Right-click the tab, **Shifts.dsv (Design),** and then click **Save** to save the new data source view definition.

Predicting Service Metrics for Each Shift

Now that you have generated some values for each shift, you will use those values as input to the logistic regression model that you built, to generate some predictions that can be used in business planning.

To use the new DSV as input to a prediction query

1. In Data Mining Designer, click the **Mining Model Prediction** tab.
2. In the **Mining Model** pane, click **Select Model**, and choose Call Center - LR from the list of available models.
3. From the **Mining Model** menu, clear the option, **Singleton Query**. A warning tells you that the singleton query inputs will be lost. Click **OK**.

   The **Singleton Query Input** dialog box is replaced with the **Select Input Table(s)** dialog box.
4. Click **Select Case Table**.
5. In the **Select Table** dialog box, select Shifts from the list of data sources. In the **Table/View name** list, select Shifts for Call Center (it might be automatically selected), and then click **OK.**

   The **Mining Model Prediction** design surface is updated to show mappings that are created based on the names and data types of columns in the input data and

in the model.

6. Right-click one of the join lines, and then select **Modify Connections**.

   In this dialog box, you can see exactly which columns are mapped and which are not. The mining model contains columns for Calls, Orders, IssuesRaised, and LvlTwoOperators, which you can map to any of the aggregates that you created based on these columns in the source data. In this scenario, you will map to the averages.

7. Click the empty cell next to LevelTwoOperators, and select **Shifts for Call Center.AvgOperators**.
8. Click the empty cell next to Calls, select **Shifts for Call Center.AvgCalls**. and then click **OK**.

To create the predictions for each shift

1. In the grid at the bottom half of the **Prediction Query Builder**, click the empty cell under **Source,** and then select Shifts for Call Center.
2. In the empty cell under **Field**, select Shift.
3. Click the next empty line in the grid and repeat the procedure described above to add another row for WageType.
4. Click the next empty line in the grid. In the **Source** column, select **Prediction Function**. In the **Field** column, select **Predict**.
5. Drag the column ServiceGrade from the **Mining Model** pane down to the grid, and into the **Criteria/Argument** cell. In the **Alias** field, type **Predicted Service Grade**.
6. Click the next empty line in the grid. In the **Source** column, select **Prediction Function**. In the **Field** column, select **PredictProbability**.
7. Drag the column ServiceGrade from the **Mining Model** pane down to the grid, and into the **Criteria/Argument** cell. In the **Alias** field, type **Probability**.
8. Click **Switch to query result view** to view the predictions.

The following table shows sample results for each shift.

Shift	WageType	Predicted Service Grade	Probability
AM	holiday	0.165	0.377520666
midnight	holiday	0.105	0.364105573
PM1	holiday	0.165	0.40056055
PM2	holiday	0.165	0.338532973

Shift	WageType	Predicted Service Grade	Probability
AM	weekday	0.165	0.370847617
midnight	weekday	0.08	0.352999173
PM1	weekday	0.165	0.317419177
PM2	weekday	0.105	0.311672027

Predicting the Effect of Reduced Response Time on Service Grade

You generated some average values for each shift, and used those values as input to the logistic regression model. However, given that the business objective is to keep abandon rate within the range 0.00-0.05, the results are not encouraging.

Therefore, based on the original model, which showed a strong influence of response time on service grade, the Operations team decides to run some predictions to assess whether reducing the average time for responding to calls might improve service quality. For example, if you cut the call response time to 90 percent or even to 80 percent of the current call response time, what would happen to service grade values?

It is easy to create a data source view (DSV) that calculates the average response times for each shift, and then add columns that calculate 80% or 90% of the average response time. You can then use the DSV as input to the model.

Although the exact steps are not shown here, the following table compares the effects on service grade when you reduce response times to 80% or to 90% of current response times.

From these results, you might conclude that on targeted shifts you should reduce the response time to 90 percent of the current rate in order to improve service quality.

Shift, wage, and day	Predicted service quality with current average response time	Predicted service quality with 90 percent reduction in response time	Predicted service quality with 80 percent reduction in response time
Holiday AM	0.165	0.05	0.05
Holiday PM1	0.05	0.05	0.05
Holiday Midnight	0.165	0.05	0.05

There are a variety of other prediction queries that you can create on this model. For example, you could predict how many operators are required to meet a certain service level or to respond to a certain number of incoming calls. Because you can include

multiple outputs in a logistic regression model, it is easy to experiment with different independent variables and outcomes without having to create many separate models.

Remarks

The Data Mining Add-Ins for Excel 2007 provide logistic regression wizards that make it easy to answer complex questions, such as how many Level Two Operators would be required to improve service grade to a target level for a specific shift. The data mining add-ins are a free download, and include wizards that are based on the neural network or logistic regression algorithms. For more information, see the following links:

- SQL Server 2005 Data Mining Add-Ins for Office 2007: Goal Seek and What If Scenario Analysis
- SQL Server 2008 Data Mining Add-Ins for Office 2007: Goal Seek Scenario Analysis, What If Scenario Analysis, and Prediction Calculator

Conclusion

You have learned to create, customize, and interpret mining models that are based on the Microsoft Neural Network algorithm and the Microsoft Logistic Regression algorithm. These model types are sophisticated and permit almost infinite variety in analysis, and therefore can be complex and difficult to master.

However, these algorithms can iterate through many combinations of factors and automatically identify the strongest correlations, providing statistical support for insights that would be very difficult to discover through manual exploration of data using Transact-SQL or even PowerPivot.

See Also

Querying a Logistic Regression Model (Analysis Services - Data Mining)

Microsoft Logistic Regression Algorithm

Microsoft Neural Network Algorithm

Querying a Neural Network Model (Analysis Services- Data Mining)

Creating and Querying Data Mining Models with DMX: Tutorials (Analysis Services - Data Mining)

After you have created a data mining solution by using Microsoft SQL Server Analysis Services, you can create queries against the data mining models to predict trends, retrieve patterns in the data, and measure the accuracy of the mining models.

The step-by-step tutorials in the following list will help you learn how to build and run data mining queries by using Analysis Services so that you can get the most from your data.

In this Section

- Bike Buyer DMX Tutorial

  This tutorial walks you through the creation of a new mining structure and mining models by using the Data Mining Extensions (DMX) language, and explains how to create DMX prediction queries.
- Market Basket DMX Tutorial

  This tutorial uses a typical market basket scenario, where you find associations between the products that customers purchase together. This tutorial also demonstrates how to use nested tables when you create a mining structure. You build and train a model based on this structure, and then create predictions using DMX.
- Time Series Prediction DMX Tutorial

  This tutorial creates a forecasting model to illustrate the use of the CREATE MODEL (DMX) statement. You then add related models and customize the behavior of each by changing the parameters of the Microsoft Time Series algorithm. Finally you create predictions and update the predictions with new data. The ability to update a time series while making predictions was added in SQL Server 2008.

Reference

Data Mining Algorithms (Analysis Services - Data Mining)

Data Mining Extensions (DMX) Reference

Related Sections

- Basic Data Mining Tutorial

  This tutorial introduces basic concepts, such as how to create a project and how to build mining structures and mining models.
- Intermediate Data Mining Tutorial (Analysis Services - Data Mining)

  This tutorial contains a number of independent lessons, each introducing you to a different model type. Each lesson walks you through the process of creating a model, exploring the model, and then customizing the model and creating prediction queries.

See Also

Working with Data Mining

Using the Data Mining Tools

Designing and Implementing (Analysis Services - Data Mining)

Bike Buyer DMX Tutorial

In this tutorial, you will learn how create, train, and explore mining models by using the Data Mining Extensions (DMX) query language. You will then use these mining models to create predictions that determine whether a customer will purchase a bicycle.

The mining models will be created from the data contained in the sample database, which stores data for the fictitious company Adventure Works Cycles. Adventure Works Cycles is a large, multinational manufacturing company. The company manufactures and sells metal and composite bicycles to North American, European, and Asian commercial markets. Its base operation is located in Bothell, Washington, with 290 employees, and it has several regional sales teams located throughout their international market base. For more information about the sample database, see Data Mining Concepts.

Tutorial Scenario

Adventure Works Cycles has decided to extend their data analysis by creating a custom application that uses data mining functionality. Their goal for the custom application is to be able to:

- Take as input specific characteristics about a potential customer and predict whether they will buy a bicycle.
- Take as input a list of potential customers, as well as characteristics about the customers, and predict which ones will buy a bicycle.

In the first case, customer data is provided by a customer registration page, and in the second case, a list of potential customers is provided by the Adventure Works Cycles marketing department.

In addition, the marketing department has asked for the ability to group existing customers into categories based on characteristics such as where they live, the number of children they have, and their commute distance. They want to see whether these clusters can be used to help target specific kinds of customers. This will require an additional mining model.

Microsoft SQL Server Analysis Services provides several tools that can be used to accomplish these tasks:

- The DMX query language
- The Microsoft Decision Trees Algorithm and the Microsoft Clustering Algorithm
- Query Editor in SQL Server Management Studio

Data Mining Extensions (DMX) is a query language provided by Analysis Services that you can use to create and work with mining models. The Microsoft Decision Trees algorithm creates models that can be used to predict whether someone will purchase a bicycle. The resulting model can take an individual customer or a table of customers as an input. The Microsoft Clustering algorithm can create groupings of customers based on shared characteristics. The goal of this tutorial is to provide the DMX scripts that will be used in the custom application.

For more information: Working with Data Mining

Mining Structure and Mining Models

Before you begin to create DMX statements, it is important to understand the main objects that Analysis Services uses to create mining models. The mining structure is a data structure that defines the data domain from which mining models are built. A single mining structure can contain multiple mining models that share the same domain. A mining model applies a mining model algorithm to the data, which is represented by a mining structure.

The building blocks of the mining structure are the mining structure columns, which describe the data that the data source contains. These columns contain information such as data type, content type, and how the data is distributed.

Mining models must contain the key column described in the mining structure, as well as a subset of the remaining columns. The mining model defines the usage for each column and defines the algorithm that is used to create the mining model. For example, in DMX you can specify that a column is a Key column or a PREDICT column. If a column is left unspecified, it is assumed to be an input column.

In DMX, there are two ways to create mining models. You can either create the mining structure and associated mining model together by using the CREATE MINING MODEL statement, or you can first create a mining structure by using the CREATE MINING STRUCTURE statement, and then add a mining model to the structure by using the ALTER STRUCTURE statement. These methods are described in the following table.

CREATE MINING MODEL

Use this statement to create a mining structure and associated mining model together using the same name. The mining model name is appended with "Structure" to differentiate it from the mining structure. This statement is useful if you are creating a mining structure that will contain a single mining model.

For more information, see CREATE MINING MODEL (DMX).

ALTER MINING STRUCTURE

Use this statement to add a mining model to a mining structure that already exists on the server. This statement is useful if you want to create a mining structure that contains several different mining models. There are several reasons that you would want to add more than one mining model in a single mining structure. For example, you might create several mining models that use different algorithms to see which algorithm works best. You might create several mining models that use the same algorithm, but with a parameter set differently for each mining model to find the best setting for the parameter.

For more information, see ALTER MINING STRUCTURE (DMX).

Because you will create a mining structure that contains several mining models, you will use the second method in this tutorial.

For More Information

Data Mining Extensions (DMX) Reference, Understanding the Select Statement (DMX), Prediction Queries (DMX)

What You Will Learn

This tutorial is divided into the following lessons:

Lesson 1: Creating the Predictive Mining Structure

In this lesson, you will learn how to use the **CREATE** statement to create mining structures.

Lesson 2: Adding Mining Models to the Predictive Mining Structure

In this lesson, you will learn how to use the **ALTER** statement to add mining models to a mining structure.

Lesson 3: Processing the Predictive Mining Structure

In this lesson you will learn how to use the **INSERT INTO** statement to process mining structures and their associated mining models.

Lesson 4: Browsing the Content the Mining Models

In this lesson, you will learn how to use the **SELECT** statement to explore the content of the mining models.

Lesson 5: Creating Predictions

In this lesson, you will learn how to use the **PREDICTION JOIN** statement to create predictions against mining models.

Requirements

Before doing this tutorial, make sure that the following are installed:

- Microsoft SQL Server
- Microsoft SQL Server 2005 Analysis Services (SSAS), SQL Server 2008 Analysis Services (SSAS), SQL Server 2012 Analysis Services (SSAS), or SQL Server Analysis Services
- The database. By default, the sample databases are not installed, to enhance security. To install official sample databases for Microsoft SQL Server, visit the Microsoft SQL Sample Databases page and select the databases that you want to install. For more information about how to install the sample databases, see Initial Installation (Analysis Services).

Note

When you review tutorials, we recommend that you add **Next topic** and **Previous topic** buttons to the document viewer toolbar. For more information, see Adding Next and Previous Buttons to Help.

See Also

Lesson 1: Creating the Bike Buyer Mining Structure

In this lesson, you will create a mining structure that allows you to predict whether a potential customer of Adventure Works Cycles will purchase a bicycle. If you are unfamiliar with mining structures and their role in data mining, see Mining Structures.

The Bike Buyer mining structure that you will create in this lesson supports adding mining models based on the Microsoft Clustering Algorithm and the Microsoft Decision Trees Algorithm. In later lessons, you will use the clustering mining models to explore the different ways in which customers can be grouped, and will use decision tree mining models to predict whether or not a potential customer will purchase a bicycle.

CREATE MINING STRUCTURE Statement

To create a mining structure, you use the CREATE MINING STRUCTURE (DMX) statement. The code in the statement can be broken into the following parts:

- Naming the structure.
- Defining the key column.
- Defining the mining columns.
- Defining an optional testing data set.

The following is a generic example of the CREATE MINING STRUCTURE statement:

```
CREATE MINING STRUCTURE [<mining structure name>]
(
    <key column>,
    <mining structure columns>
)
WITH HOLDOUT (<holdout specifier>)
```

The first line of the code defines the name of the structure:

```
CREATE MINING STRUCTURE [<mining structure name>]
```

For information about naming an object in Data Mining Extensions (DMX), see Identifiers (DMX).

The next line of the code defines the key column for the mining structure, which uniquely identifies an entity in the source data:

```
<key column>,
```

In the mining structure you will create, the customer identifier, `CustomerKey`, defines an entity in the source data.

The next line of the code is used to define the mining columns that will be used by the mining models associated with the mining structure:

```
<mining structure columns>
```

You can use the DISCRETIZE function within <mining structure columns> to discretize continuous columns by using the following syntax:

```
DISCRETIZE(<method>,<number of buckets>)
```

For more information about discretizing columns, see Discretization Methods. For more information about the types of mining structure columns that you can define, see Mining Structure Columns.

The final line of the code defines an optional partition in the mining structure:

```
WITH HOLDOUT (<holdout specifier>)
```

You specify some portion of the data to use for testing mining models that are related to the structure, and the remaining data is used for training the models. By default, Analysis Services creates a test data set that contains 30 percent of all case data. You will add the specification that the test data set should contain 30 percent of the cases up to a maximum of 1000 cases. If 30 percent of the cases is less than 1000, the test data set will contain the smaller amount.

Lesson Tasks

You will perform the following tasks in this lesson:

- Create a new blank query.
- Alter the query to create the mining structure.
- Execute the query.

Creating the Query

The first step is to connect to an instance of Analysis Services and create a new DMX query in SQL Server Management Studio.

To create a new DMX query in SQL Server Management Studio

1. Open SQL Server Management Studio.
2. In the **Connect to Server** dialog box, for **Server type**, select **Analysis Services**. In **Server name**, type **LocalHost**, or type the name of the instance of Analysis Services that you want to connect to for this lesson. Click **Connect**.
3. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX** to open the **Query Editor** and a new, blank query.

Altering the Query

The next step is to modify the CREATE MINING STRUCTURE statement described above to create the Bike Buyer mining structure.

To customize the CREATE MINING STRUCTURE statement

1. In the Query Editor, copy the generic example of the CREATE MINING

STRUCTURE statement into the blank query.

2. Replace the following:

```
[<mining structure>]
```

with:

```
[Bike Buyer]
```

3. Replace the following:

```
<key column>
```

with:

```
CustomerKey LONG KEY
```

4. Replace the following:

```
<mining structure columns>
```

with:

```
    [Age] LONG DISCRETIZED(Automatic,10),
    [Bike Buyer] LONG DISCRETE,
    [Commute Distance] TEXT DISCRETE,
    [Education] TEXT DISCRETE,
    [Gender] TEXT DISCRETE,
    [House Owner Flag] TEXT DISCRETE,
    [Marital Status] TEXT DISCRETE,
    [Number Cars Owned] LONG DISCRETE,
    [Number Children At Home] LONG DISCRETE,
    [Occupation] TEXT DISCRETE,
    [Region] TEXT DISCRETE,
    [Total Children]LONG DISCRETE,
    [Yearly Income] DOUBLE CONTINUOUS
```

5. Replace the following:

```
WITH HOLDOUT (holdout specifier>)
```

with:

```
WITH HOLDOUT (30 PERCENT or 1000 CASES)
```

The complete mining structure statement should now be as follows:

```
CREATE MINING STRUCTURE [Bike Buyer]
(
   [Customer Key] LONG KEY,
   [Age]LONG DISCRETIZED(Automatic,10),
```

```
    [Bike Buyer] LONG DISCRETE,
    [Commute Distance] TEXT DISCRETE,
    [Education] TEXT DISCRETE,
    [Gender] TEXT DISCRETE,
    [House Owner Flag] TEXT DISCRETE,
    [Marital Status] TEXT DISCRETE,
    [Number Cars Owned]LONG DISCRETE,
    [Number Children At Home]LONG DISCRETE,
    [Occupation] TEXT DISCRETE,
    [Region] TEXT DISCRETE,
    [Total Children]LONG DISCRETE,
    [Yearly Income] DOUBLE CONTINUOUS
)
WITH HOLDOUT (30 PERCENT or 1000 CASES)
```

6. On the **File** menu, click **Save DMXQuery1.dmx As**.
7. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Bike Buyer Structure.dmx**.

Executing the Query

The final step is to execute the query. After a query is created and saved, it needs to be executed. That is, the statement needs to be run in order to create the mining structure on the server. For more information about executing queries in Query Editor, see SQL Server Management Studio Transact SQL Query.

To execute the query

1. In Query Editor, on the toolbar, click **Execute**.

   The status of the query is displayed in the **Messages** tab at the bottom of Query Editor after the statement finishes executing. Messages should display:

   ```
   Executing the query
   Execution complete
   ```

   A new structure named **Bike Buyer** now exists on the server.

In the next lesson, you will add mining models to the structure you just created.

Next Lesson

Lesson 2: Adding Mining Models to the Predictive Mining Structure

Lesson 2: Adding Mining Models to the Bike Buyer Mining Structure

In this lesson, you will add two mining models to the Bike Buyer mining structure that you created Lesson 1: Creating the Bike Buyer Mining Structure. These mining models will allow you to explore the data using one model, and to create predictions using another.

To explore how potential customers can be categorized by their characteristics, you will create a mining model based on the Microsoft Clustering Algorithm. In a later lesson, you will explore how this algorithm finds clusters of customers who share similar characteristics. For example, you might find that certain customers tend to live close to each other, commute by bicycle, and have similar education backgrounds. You can use these clusters to better understand how different customers are related, and to use the information to create a marketing strategy that targets specific customers.

To predict whether a potential customer is likely to buy a bicycle, you will create a mining model based on the Microsoft Decision Trees Algorithm. This algorithm looks through the information that is associated with each potential customer, and finds characteristics that are useful in predicting if they will buy a bicycle. It then compares the values of the characteristics of previous bike buyers against new potential customers to determine whether the new potential customers are likely to buy a bicycle.

ALTER MINING STRUCTURE Statement

In order to add a mining model to the mining structure, you use the ALTER MINING STRUCTURE (DMX) statement. The code in the statement can be broken into the following parts:

- Identifying the mining structure
- Naming the mining model
- Defining the key column
- Defining the input and predictable columns
- Identifying the algorithm and parameter changes

The following is a generic example of the ALTER MINING MODEL statement:

```
ALTER MINING STRUCTURE [<mining structure name>]
ADD MINING MODEL [<mining model name>]
(
    [<key column>],
    <mining model columns>,
) USING <algorithm name>( <algorithm parameters> )
WITH FILTER (<expression>)
```

The first line of the code identifies the existing mining structure to which the mining models will be added:

```
ALTER MINING STRUCTURE [<mining structure name>]
```

The next line of the code names the mining model that will be added to the mining structure:

```
ADD MINING MODEL [<mining model name>]
```

For information about naming an object in DMX, see Identifiers (DMX).

The next lines of the code define columns from the mining structure that will be used by the mining model:

```
[<key column>],
<mining model columns>
```

You can only use columns that already exist in the mining structure, and the first column in the list must be the key column from the mining structure.

The next line of the code defines the mining algorithm that generates the mining model and the algorithm parameters that you can set on the algorithm:

```
) USING <algorithm name>( <algorithm parameters> )
```

For more information about the algorithm parameters that you can adjust, see Microsoft Decision Trees Algorithm and Microsoft Clustering Algorithm.

You can specify that a column in the mining model be used for prediction by using the following syntax:

```
<mining model column> PREDICT
```

The final line of the code, which is optional, defines a filter that is applied when training and testing the model. For more information about how to apply filters to mining models, see Creating Filters for Mining Models (Analysis Services - Data Mining).

Lesson Tasks

You will perform the following tasks in this lesson:

- Add a decision tree mining model to the Bike Buyer structure by using the Microsoft Decision Trees algorithm
- Add a clustering mining model to the Bike Buyer structure by using the Microsoft Clustering algorithm
- Because you want to see results for all cases, you will not yet add a filter to either model.

Adding a Decision Tree Mining Model to the Structure

The first step is to add a mining model based on the Microsoft Decision Trees algorithm.

To add a decision tree mining model

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX** to open Query Editor and a new, blank query.
2. Copy the generic example of the ALTER MINING STRUCTURE statement into the

blank query.

3. Replace the following:

```
<mining structure name>
```

with:

```
[Bike Buyer]
```

4. Replace the following:

```
<mining model name>
```

with:

```
Decision Tree
```

5. Replace the following:

```
<mining model columns>,
```

with:

```
(
   CustomerKey,
   [Age],
   [Bike Buyer] PREDICT,
   [Commute Distance],
   [Education],
   [Gender],
   [House Owner Flag],
   [Marital Status],
   [Number Cars Owned],
   [Number Children At Home],
   [Occupation],
   [Region],
   [Total Children],
   [Yearly Income]
```

In this case, the `[Bike Buyer]` column has been designated as the PREDICT column.

6. Replace the following:

```
USING <algorithm name>( <algorithm parameters> )
```

with:

```
Using Microsoft_Decision_Trees
WITH DRILLTHROUGH
```

The WITH DRILLTHROUGH statement allows you to explore the cases that were used to build the mining model.

The resulting statement should now be as follows:

```
ALTER MINING STRUCTURE [Bike Buyer]
ADD MINING MODEL [Decision Tree]
(
   CustomerKey,
   [Age],
   [Bike Buyer] PREDICT,
   [Commute Distance],
   [Education],
   [Gender],
   [House Owner Flag],
   [Marital Status],
   [Number Cars Owned],
   [Number Children At Home],
   [Occupation],
   [Region],
   [Total Children],
   [Yearly Income]
) USING Microsoft_Decision_Trees
WITH DRILLTHROUGH
```

7. On the **File** menu, click **Save DMXQuery1.dmx As**.
8. In the **Save As** dialog box, browse to the appropriate folder, and name the file **DT_Model.dmx**.
9. On the toolbar, click the **Execute** button.

Adding a Clustering Mining Model to the Structure

You can now add a mining model to the Bike Buyer mining structure based on the Microsoft Clustering algorithm. Because the clustering mining model will use all the columns defined in the mining structure, you can use a shortcut to add the model to the structure by omitting the definition of the mining columns.

To add a Clustering mining model

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX** to open Query Editor opens and a new, blank query.

2. Copy the generic example of the ALTER MINING STRUCTURE statement into the blank query.
3. Replace the following:

   ```
   <mining structure name>
   ```

   with:

   ```
   [Bike Buyer]
   ```
4. Replace the following:

   ```
   <mining model>
   ```

   with:

   ```
   Clustering Model
   ```
5. Delete the following:

   ```
   (
       [<key column>],
       <mining model columns>,
   )
   ```
6. Replace the following:

   ```
   USING <algorithm name>( <algorithm parameters> )
   ```

   with:

   ```
   USING Microsoft_Clustering
   ```

   The complete statement should now be as follows:

   ```
   ALTER MINING STRUCTURE [Bike Buyer]
   ADD MINING MODEL [Clustering]
   USING Microsoft_Clustering
   ```
7. On the **File** menu, click **Save DMXQuery1.dmx As**.
8. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Clustering_Model.dmx**.
9. On the toolbar, click the **Execute** button.

In the next lesson, you will process the models and the mining structure.

Next Lesson

Lesson 3: Processing the Predictive Mining Structure

Lesson 3: Processing the Bike Buyer Mining Structure

In this lesson, you will use the INSERT INTO statement and the vTargetMail view from the sample database to process the mining structures and mining models that you created in

Lesson 1: Creating the Predictive Mining Structure and Lesson 2: Adding Mining Models to the Predictive Mining Structure.

When you process a mining structure, Analysis Services reads the source data and builds the structures that support mining models. When you process a mining model, the data defined by the mining structure is passed through the data mining algorithm that you choose. The algorithm searches for trends and patterns, and then stores this information in the mining model. The mining model, therefore, does not contain the actual source data, but instead contains the information that was discovered by the algorithm. For more information about processing mining models, see Processing Data Mining Objects.

You need to reprocess a mining structure only if you change a structure column or change the source data. If you add a mining model to a mining structure that has already been processed, you can use the INSERT INTO MINING MODEL statement to train the new mining model.

Train Structure Template

In order to train the mining structure and its associated mining models, use the INSERT INTO (DMX) statement. The code in the statement can be broken into the following parts:

- Identifying the mining structure
- Listing the columns in the mining structure
- Defining the training data

The following is a generic example of the INSERT INTO statement:

```
INSERT INTO MINING STRUCTURE [<mining structure name>]
(
   <mining structure columns>
)
OPENQUERY([<datasource>],'<SELECT statement>')
```

The first line of the code identifies the mining structure that you will train:

```
INSERT INTO MINING STRUCTURE [<mining structure name>]
```

The next line of the code specifies the columns that are defined by the mining structure. You must list each column in the mining structure, and each column must map to a column contained within the source query data.

```
(
   <mining structure columns>
)
```

The final line of the code defines the data that will be used to train the mining structure:

```
OPENQUERY([<datasource>],'<SELECT statement>')
```

In this lesson, you use **OPENQUERY** to define the source data. For information about other methods of defining the source query, see <source data query>.

Lesson Tasks

You will perform the following task in this lesson:

- Process the Bike Buyer mining structure

Processing the Predictive Mining Structure

To process the mining structure by using INSERT INTO

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.
   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the INSERT INTO statement into the blank query.
3. Replace the following:

   ```
   [<mining structure name>]
   ```

   with:

   ```
   Bike Buyer
   ```

4. Replace the following:

   ```
   <mining structure columns>
   ```

   with:

   ```
   [Customer Key],
   [Age],
   [Bike Buyer],
   [Commute Distance],
   [Education],
   [Gender],
   [House Owner Flag],
   [Marital Status],
   [Number Cars Owned],
   [Number Children At Home],
   [Occupation],
   [Region],
   [Total Children],
   [Yearly Income]
   ```

5. Replace the following:

   ```
   OPENQUERY([<datasource>],'<SELECT statement>')
   ```

with:

```
OPENQUERY([Adventure Works DW],
   'SELECT CustomerKey, Age, BikeBuyer,
         CommuteDistance,EnglishEducation,
         Gender,HouseOwnerFlag,MaritalStatus,
         NumberCarsOwned,NumberChildrenAtHome,
         EnglishOccupation,Region,TotalChildren,
         YearlyIncome
    FROM dbo.vTargetMail')
```

The OPENQUERY statement references the Adventure Works DW Multidimensional 2012 data source to access the view vTargetMail. The view contains the source data that will be used to train the mining models.

The complete statement should now be as follows:

```
INSERT INTO MINING STRUCTURE [Bike Buyer]
(
   [Customer Key],
   [Age],
   [Bike Buyer],
   [Commute Distance],
   [Education],
   [Gender],
   [House Owner Flag],
   [Marital Status],
   [Number Cars Owned],
   [Number Children At Home],
   [Occupation],
   [Region],
   [Total Children],
   [Yearly Income]
)
OPENQUERY([Adventure Works DW],
   'SELECT CustomerKey, Age, BikeBuyer,
         CommuteDistance,EnglishEducation,
         Gender,HouseOwnerFlag,MaritalStatus,
```

```
        NumberCarsOwned,NumberChildrenAtHome,
        EnglishOccupation,Region,TotalChildren,
        YearlyIncome
    FROM dbo.vTargetMail')
```

6. On the **File** menu, click **Save DMXQuery1.dmx As**.
7. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Process Bike Buyer Structure.dmx**.
8. On the toolbar, click the **Execute** button.

In the next lesson, you will explore content in the mining models you added to the mining structure in this lesson.

Next Lesson

Lesson 4: Browsing the Content the Mining Models

Lesson 4: Browsing the Bike Buyer Mining Models

In this lesson, you will use the SELECT (DMX) statement to explore the content in the decision tree and clustering mining models that you created in Lesson 2: Adding Mining Models to the Predictive Mining Structure.

The columns contained in a mining model are not the columns defined by the mining structure, but instead are a specific set of columns that describe the trends and patterns that are found by the algorithm. These mining model columns are described in the DMSCHEMA_MINING_MODEL_CONTENT Rowset schema rowset. For example, the MODEL_NAME column in the content schema rowset contains the name of the mining model. For a clustering mining model, the NODE_CAPTION column contains the name of each cluster, and the NODE_DESCRIPTION column contains a description of the characteristics of each cluster. You can browse these columns by using the SELECT FROM <model>.CONTENT statement in DMX. You can also use this statement to explore the data that was used to create the mining model. Drillthrough must be enabled on the mining structure in order to use this statement. For more information about the statement, see Lesson 5: Executing Prediction Queries.

You can also return all the states of a discrete column by using the SELECT DISTINCT statement. For example, if you perform this operation on a gender column, the query will return **male** and **female**.

Lesson Tasks

You will perform the following tasks in this lesson:

- Explore the content contained within the mining models
- Return the cases from the source data that was used to train the mining models
- Explore the different states available for a specific discrete column

Returning the Content of a Mining Model

In this lesson, you use the SELECT FROM <model>.CONTENT (DMX) statement to return the contents of the clustering model.

The following is a generic example of the SELECT FROM <model>.CONTENT statement:

```
SELECT <select list> FROM [<mining model>].CONTENT
WHERE <where clause>
```

The first line of the code defines the columns to return from the mining model content, and the mining model they are associated with:

```
SELECT <select list> FROM [<mining model].CONTENT
```

The .CONTENT clause next to the name of the mining model specifies that you are returning content from the mining model. For more information about the columns contained in the mining model, see DMSCHEMA_MINING_MODEL_CONTENT Rowset.

You can optionally use the final line of the code to filter the results returned by the statement:

```
WHERE <where clause>
```

For example, if you want to restrict the results of the query to only the clusters that contain a high number of cases, you can add the following WHERE clause to the SELECT statement:

```
WHERE NODE_SUPPORT > 100
```

For more information about using the WHERE statement, see SELECT (DMX).

To return the content of the clustering mining model

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the SELECT FROM <model>.CONTENT statement into the blank query.
3. Replace the following:

   ```
   <select list>
   ```

   with:

   ```
   *
   ```

   You can also replace * with a list of any of the columns contained within the DMSCHEMA_MINING_MODEL_CONTENT Rowset.
4. Replace the following:

   ```
   [<mining model>]
   ```

   with:

   ```
   [Clustering]
   ```

The complete statement should now be as follows:

```
SELECT * FROM [Clustering].CONTENT
```

5. On the **File** menu, click **Save DMXQuery1.dmx As**.
6. In the **Save As** dialog box, browse to the appropriate folder, and name the file **SELECT_CONTENT.dmx**.
7. On the toolbar, click the **Execute** button.

   The query returns the content of the mining model.

Use Drillthrough

The next step is to use the drillthrough statement to return a sampling of the cases that were used to train the decision tree mining model. In this lesson, you use the SELECT FROM <model>.CASES (DMX) statement to return the contents of the decision tree model.

The following is a generic example of the SELECT FROM <model>.CASES statement:

```
SELECT <select list>
FROM [<mining model>].CASES
WHERE IsInNode('<node id>')
```

The first line of the code defines the columns to return from the source data, and the mining model they are contained within:

```
SELECT <select list> FROM [<mining model>].CASES
```

The .CASES clause specifies that you are performing a drillthrough query. In order to use drillthrough you must enable drillthrough when you create the mining model.

The final line of the code is optional and specifies the node in the mining model that you are requesting cases from:

```
WHERE IsInNode('<node id>')
```

For more information about using the WHERE statement with IsInNode, see SELECT FROM <model>.CASES (DMX).

To return the cases that were used to train the mining model

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the SELECT FROM <model>.CASES statement into the blank query.
3. Replace the following:

   ```
   <select list>
   ```

   with:

```
*
```

You can also replace * with a list of any of the columns contained within the source data (such as [Bike Buyer]).

4. Replace the following:

```
[<mining model>]
```

with:

```
[Decision Tree]
```

The complete statement should now be as follows:

```
SELECT *
FROM [Decision Tree].CASES
```

5. On the **File** menu, click **Save DMXQuery1.dmx As**.
6. In the **Save As** dialog box, browse to the appropriate folder, and name the file **SELECT_DRILLTHROUGH.dmx**.
7. On the toolbar, click the **Execute** button.

   The query returns the source data that was used to train the decision tree mining model.

Return the States of a Discrete Mining Model Column

The next step is to use the SELECT DISTINCT statement to return the different possible states in the specified mining model column.

The following is a generic example of the SELECT DISTINCT statement:

```
SELECT DISTINCT [<column>]
FROM [<mining model>]
```

The first line of the code defines the mining model columns for which the states are returned:

```
SELECT DISTINCT [<column>]
```

You must include DISTINCT in order to return all of the states of the column. If you exclude DISTINCT, then the full statement becomes a shortcut for a prediction and returns the most likely state of the specified column. For more information, see SELECT (DMX).

To return the states of a discrete column

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the SELECT Distinct statement into the blank query.
3. Replace the following:

```
[<column,name>
```

with:

```
[Bike Buyer]
```

4. Replace the following:

   ```
   [<mining model>]
   ```

   with:

   ```
   [Decision Tree]
   ```

   The complete statement should now be as follows:

   ```
   SELECT DISTINCT [Bike Buyer]
   FROM [Decision Tree]
   ```

5. On the **File** menu, click **Save DMXQuery1.dmx As**.
6. In the **Save As** dialog box, browse to the appropriate folder, and name the file **SELECT_DISCRETE.dmx**.
7. On the toolbar, click the **Execute** button.

   The query returns the possible states of the Bike Buyer column.

In the next lesson, you will predict whether potential customers will be bike buyers by using the decision tree mining model.

Next Lesson

Lesson 5: Creating Predictions

Lesson 5: Executing Prediction Queries

In this lesson, you will use the SELECT FROM <model> PREDICTION JOIN (DMX) form of the SELECT statement to create two different types of predictions based on the decision tree model you created in Lesson 2: Adding Mining Models to the Association Mining Structure. These prediction types are defined below.

Singleton Query

Use a singleton query to provide ad hoc values when making predictions. For example, you can determine whether a single customer is likely to be a bike buyer, by passing inputs to the query such as the commute distance, the area code, or the number of children of the customer. The singleton query returns a value that indicates how likely the person is to purchase a bicycle based on those inputs.

Batch Query

Use a batch query to determine who in a table of potential customers is likely to purchase a bicycle. For example, if your marketing department provides you with a list of customers and customer attributes, then you can use a batch prediction to determine who from the table is likely to purchase a bicycle.

The SELECT FROM <model> PREDICTION JOIN (DMX) form of the SELECT statement contains three parts:

- A list of the mining model columns and prediction functions that are returned in the results. The results can also contain input columns from the source data.
- The source query defining the data that is being used to create a prediction. For example, in a batch query this could be a list of customers.
- A mapping between the mining model columns and the source data. If these names match, then you can use NATURAL syntax and leave out the column mappings.

You can further enhance the query by using prediction functions. Prediction functions provide additional information, such as the probability of a prediction occurring, and provide support for the prediction in the training dataset. For more information about prediction functions, see Market Basket DMX Tutorial.

The predictions in this tutorial are based on the ProspectiveBuyer table in the sample database. The ProspectiveBuyer table contains a list of potential customers and their associated characteristics. The customers in this table are independent of the customers that were used to create the decision tree mining model.

You can also create predictions by using the prediction query builder in SQL Server Data Tools (SSDT). For more information, see Using the Prediction Query Builder to Create DMX Prediction Queries.

Lesson Tasks

You will perform the following tasks in this lesson:

- Create a singleton query to determine whether a specific customer is likely to purchase a bicycle.
- Create a batch query to determine which customers, listed in a table of customers, are likely to purchase a bicycle.

Singleton Query

The first step is to use the SELECT FROM <model> PREDICTION JOIN (DMX) in a singleton prediction query. The following is a generic example of the singleton statement:

```
SELECT <select list> FROM [<mining model name>]

NATURAL PREDICTION JOIN

(SELECT '<value>' AS [<column>], ...)

AS [<input alias>]
```

The first line of the code defines the columns from the mining model that the query should return, and specifies the mining model that is used to generate the prediction:

```
SELECT <select list> FROM [<mining model name>]
```

The next lines of the code define the characteristics of the customer that you use to create a prediction:

```
NATURAL PREDICTION JOIN
(SELECT '<value>' AS [<column>], ...)
AS [<input alias>]
ORDER BY <expression>
```

If you specify NATURAL PREDICTION JOIN, the server matches each column from the model to a column from the input, based on column names. If column names do not match, the columns are ignored.

To create a singleton prediction query

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.
   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the singleton statement into the blank query.
3. Replace the following:
   ```
   <select list>
   ```
   with:
   ```
   [Bike Buyer] AS Buyer, PredictHistogram([Bike Buyer]) AS
   Statistics
   ```
   The AS statement is used to alias columns returned by the query. The PredictHistogram function returns statistics about the prediction, including the probability and the support. For more information about the functions that can be used in a prediction statement, see Functions (DMX).
4. Replace the following:
   ```
   [<mining model>]
   ```
   with:
   ```
   [Decision Tree]
   ```
5. Replace the following:
   ```
   (SELECT '<value>' AS [<column name>], ...)  AS t
   ```
   with:
   ```
   (SELECT 35 AS [Age],
     '5-10 Miles' AS [Commute Distance],
     '1' AS [House Owner Flag],
     2 AS [Number Cars Owned],
     2 AS [Total Children]) AS t
   ```
   The complete statement should now be as follows:
   ```
   SELECT
   ```

```
    [Bike Buyer] AS Buyer,
    PredictHistogram([Bike Buyer]) AS Statistics
FROM
    [Decision Tree]
NATURAL PREDICTION JOIN
(SELECT 35 AS [Age],
    '5-10 Miles' AS [Commute Distance],
    '1' AS [House Owner Flag],
    2 AS [Number Cars Owned],
    2 AS [Total Children]) AS t
```

6. On the **File** menu, click **Save DMXQuery1.dmx As**.
7. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Singleton_Query.dmx**.
8. On the toolbar, click the **Execute** button.

   The query returns a prediction about whether a customer with the specified characteristics will purchase a bicycle, as well as statistics about that prediction.

Batch Query

The next step is to use the SELECT FROM <model> PREDICTION JOIN (DMX) in a batch prediction query. The following is a generic example of a batch statement:

```
SELECT TOP <number> <select list>
FROM [<mining model name>]
PREDICTION JOIN
OPENQUERY([<datasource>],'<SELECT statement>')
  AS [<input alias>]
ON <on clause, mapping,>
WHERE <where clause, boolean expression,>
ORDER BY <expression>
```

As in the singleton query, the first two lines of the code define the columns from mining model that the query returns, as well as the name of the mining model that is used to generate the prediction. The TOP <number> statement specifies that the query will only return the number or the results specified by <number>.

The next lines of the code define the source data that the predictions are based on:

```
OPENQUERY([<datasource>],'<SELECT statement>')
  AS [<input alias>]
```

You have several options for the method of retrieving the source data, but in this tutorial, you will use OPENQUERY. For more information about the options available, see <source data query>.

The next line defines the mapping between the source columns in the mining model and the columns in the source data:

```
ON <column mappings>
```

The WHERE clause filters the results returned by the prediction query:

```
WHERE <where clause, boolean expression,>
```

The last and optional line of the code specifies the column that the results will be ordered by:

```
ORDER BY <expression> [DESC|ASC]
```

Use ORDER BY in combination with the TOP <number> statement, to filter the results that are returned. For example, in this prediction you will return the top ten bike buyers, ordered by the probability of the prediction being correct. You can use [DESC|ASC] syntax to control the order in which the results are displayed.

To create a batch prediction query

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the batch statement into the blank query.
3. Replace the following:

   ```
   <select list>
   ```

   with:

   ```
   SELECT
     TOP 10
     t.[LastName],
     t.[FirstName],
     [Decision Tree].[Bike Buyer],
     PredictProbability([Bike Buyer])
   ```

   The TOP 10 specifies that only the top ten results will be returned by the query. The ORDER BY statement in this query orders the results by the probability of the prediction being correct, so only the ten most likely results will be returned.
4. Replace the following:

   ```
   [<mining model>]
   ```

   with:

```
[Decision Tree]
```

5. Replace the following:

```
OPENQUERY([<datasource>],'<SELECT statement>')
```

with:

```
OPENQUERY([Adventure Works DW2008R2],
  'SELECT
    [LastName],
    [FirstName],
    [MaritalStatus],
    [Gender],
    [YearlyIncome],
    [TotalChildren],
    [NumberChildrenAtHome],
    [Education],
    [Occupation],
    [HouseOwnerFlag],
    [NumberCarsOwned]
  FROM
    [dbo].[ProspectiveBuyer]
  ') AS t
```

6. Replace the following:

```
<ON clause, mapping,>
WHERE <where clause, boolean expression,>
ORDER BY <expression>
```

with:

```
[Decision Tree].[Marital Status] = t.[MaritalStatus] AND
  [Decision Tree].[Gender] = t.[Gender] AND
  [Decision Tree].[Yearly Income] = t.[YearlyIncome] AND
  [Decision Tree].[Total Children] = t.[TotalChildren] AND
  [Decision Tree].[Number Children At Home] =
t.[NumberChildrenAtHome] AND
  [Decision Tree].[Education] = t.[Education] AND
  [Decision Tree].[Occupation] = t.[Occupation] AND
  [Decision Tree].[House Owner Flag] = t.[HouseOwnerFlag] AND
```

```
  [Decision Tree].[Number Cars Owned] = t.[NumberCarsOwned]
WHERE [Decision Tree].[Bike Buyer] =1
ORDER BY PredictProbability([Bike Buyer]) DESC
```

Specify `DESC` in order to list the results with the highest probability first. The complete statement should now be as follows:

```
SELECT
  TOP 10
  t.[LastName],
  t.[FirstName],
  [Decision Tree].[Bike Buyer],
  PredictProbability([Bike Buyer])
FROM
  [Decision Tree]
PREDICTION JOIN
  OPENQUERY([Adventure Works DW2008R2],
    'SELECT
      [LastName],
      [FirstName],
      [MaritalStatus],
      [Gender],
      [YearlyIncome],
      [TotalChildren],
      [NumberChildrenAtHome],
      [Education],
      [Occupation],
      [HouseOwnerFlag],
      [NumberCarsOwned]
    FROM
      [dbo].[ProspectiveBuyer]
    ') AS t
ON
  [Decision Tree].[Marital Status] = t.[MaritalStatus] AND
  [Decision Tree].[Gender] = t.[Gender] AND
```

```
    [Decision Tree].[Yearly Income] = t.[YearlyIncome] AND
    [Decision Tree].[Total Children] = t.[TotalChildren] AND
    [Decision Tree].[Number Children At Home] =
t.[NumberChildrenAtHome] AND
    [Decision Tree].[Education] = t.[Education] AND
    [Decision Tree].[Occupation] = t.[Occupation] AND
    [Decision Tree].[House Owner Flag] = t.[HouseOwnerFlag] AND
    [Decision Tree].[Number Cars Owned] = t.[NumberCarsOwned]
WHERE [Decision Tree].[Bike Buyer] =1
ORDER BY PredictProbability([Bike Buyer]) DESC
```

7. On the **File** menu, click **Save DMXQuery1.dmx As**.
8. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Batch_Prediction.dmx**.
9. On the toolbar, click the **Execute** button.

   The query returns a table containing customer names, a prediction of whether each customer will purchase a bicycle, and the probability of the prediction.

This is the last step in the Bike Buyer tutorial. You now have a set of mining models that you can use to explore similarities between you customers and predict whether potential customers will purchase a bicycle.

To learn how to use DMX in a Market Basket scenario, see Market Basket DMX Tutorial.

Market Basket DMX Tutorial

In this tutorial, you will learn how to create, train, and explore mining models by using the Data Mining Extensions (DMX) query language. You will then use these mining models to create predictions that describe which products tend to be purchased at the same time.

The mining models will be created from the data contained in the sample database, which stores data for the fictitious company Adventure Works Cycles. Adventure Works Cycles is a large, multinational manufacturing company. The company manufactures and sells metal and composite bicycles to North American, European, and Asian commercial markets. Its base operation is located in Bothell, Washington, with 290 employees, and it has several regional sales teams are located throughout their international market base.

Tutorial Scenario

Adventure Works Cycles has decided to create a custom application that employs data mining functionality to predict what types of products their customers tend to purchase at the same time. The goal for the custom application is to be able to specify a set of products, and predict what additional products will be purchased with the specified

products. Adventure Works Cycles will then use this information to add a "suggest" feature to their website, and also to better organize the way that they present information to their customers.

Microsoft SQL Server Analysis Services provides several tools that can be used to accomplish this task:

- The DMX query language
- The Microsoft Association Algorithm
- Query Editor in SQL Server Management Studio

Data Mining Extensions (DMX) is a query language provided by Analysis Services that you can use to create and work with mining models. The Microsoft Association algorithm creates models that can predict the products that are likely to be purchased together.

The goal of this tutorial is to provide the DMX queries that will be used in the custom application.

For more information: Working with Data Mining

Mining Structure and Mining Models

Before you begin to create DMX statements, it is important to understand the main objects that Analysis Services uses to create mining models. The *mining structure* is a data structure that defines the data domain from which mining models are built. A single mining structure can contain multiple *mining models* that share the same domain. A mining model applies a mining model algorithm to the data, which is represented by a mining structure.

The building blocks of the mining structure are the mining structure columns, which describe the data that the data source contains. These columns contain information such as data type, content type, and how the data is distributed.

Mining models must contain the key column described in the mining structure, as well as a subset of the remaining columns. The mining model defines the usage for each column and defines the algorithm that is used to create the mining model. For example, in DMX you can specify that a column is a Key column or a PREDICT column. If a column is left unspecified, it is assumed to be an input column.

In DMX, there are two ways to create mining models. You can either create the mining structure and associated mining model together by using the **CREATE MINING MODEL** statement, or you can first create a mining structure by using the **CREATE MINING STRUCTURE** statement, and then add a mining model to the structure by using the **ALTER STRUCTURE** statement. These methods are described below.

CREATE MINING MODEL

Use this statement to create a mining structure and associated mining model together using the same name. The mining model name is appended with "Structure" to differentiate it from the mining structure.

This statement is useful if you are creating a mining structure that will contain a single

mining model.

For more information, see CREATE MINING MODEL (DMX).

CREATE MINING STRUCTURE

Use this statement to create a new mining structure without any models.

When you use CREATE MINING STRUCTURE, you can also create a holdout data set that can be used for testing any models that are based on the same mining structure.

For more information, see CREATE MINING STRUCTURE (DMX).

ALTER MINING STRUCTURE

Use this statement to add a mining model to a mining structure that already exists on the server.

There are several reasons that you would want to add more than one mining model in a single mining structure. For example, you might create several mining models using different algorithms to see which one works best. Alternatively, you might create several mining models using the same algorithm, but with a parameter set differently for each mining model to find the best setting for that parameter.

For more information, see ALTER MINING STRUCTURE (DMX).

Because you will create a mining structure that contains several mining models, you will use the second method in this tutorial.

For More Information

Data Mining Extensions (DMX) Reference, Understanding the Select Statement (DMX), Prediction Queries (DMX)

What You Will Learn

This tutorial is divided into the following lessons:

Lesson 1: Creating the Association Mining Structure

In this lesson, you will learn how to use the **CREATE** statement to create mining structures.

Lesson 2: Adding Mining Models to the Association Mining Structure

In this lesson, you will learn how to use the **ALTER** statement to add mining models to a mining structure.

Lesson 3: Processing the Association Mining Structure

In this lesson, you will learn how to use the **INSERT INTO** statement to process mining structures and their associated mining models.

Lesson 4: Creating Association Predictions

In this lesson, you will learn how to use the **PREDICTION JOIN** statement to create predictions against mining models.

Requirements

Before doing this tutorial, make sure that the following are installed:

- Microsoft SQL Server
- Microsoft SQL Server Analysis Services
- The database

By default, the sample databases are not installed, to enhance security. To install the official sample databases for Microsoft SQL Server, go to http://www.CodePlex.com/MSFTDBProdSamples or on the Microsoft SQL Server Samples and Community Projects home page in the section Microsoft SQL Server Product Samples. Click **Databases**, then click the **Releases** tab and select the databases that you want.

Note
When you review tutorials, we recommend that you add **Next topic** and **Previous topic** buttons to the document viewer toolbar. For more information, see Adding Next and Previous Buttons to Help.

See Also

Bike Buyer DMX Tutorial

Data Mining Tutorial

Lesson 3: Building a Market Basket Scenario (Intermediate Data Mining Tutorial)

Lesson 1: Creating the Market Basket Mining Structure

In this lesson, you will create a mining structure that allows you to predict what Adventure Works Cycles products a customer tends to purchase at the same time. If you are unfamiliar with mining structures and their role in data mining, see Lesson 2: Adding Mining Models to the Market Basket Mining Structure.

The association mining structure that you will create in this lesson supports adding mining models based on the Microsoft Association Algorithm. In later lessons, you will use the mining models to predict the type of products a customer tends to purchase at the same time, which is called a market basket analysis. For example, you may find that customers tend to buy mountain bikes, bike tires, and helmets at the same time.

In this lesson, the mining structure is defined by using nested tables. Nested tables are used because the data domain that will be defined by the structure is contained within two different source tables. For more information on nested tables, see Nested Tables.

CREATE MINING STRUCTURE Statement

In order to create a mining structure containing a nested table, you use the CREATE MINING STRUCTURE (DMX) statement. The code in the statement can be broken into the following parts:

- Naming the structure
- Defining the key column

- Defining the mining columns
- Defining the nested table columns

The following is a generic example of the CREATE MINING STRUCTURE statement:

```
CREATE MINING STRUCTURE [<Mining Structure Name>]
(
   <key column>,
   <mining structure columns>,
   <table columns>
   (  <nested key column>,
      <nested mining structure columns> )
)
```

The first line of the code defines the name of the structure:

```
CREATE MINING STRUCTURE [Mining Structure Name]
```

For information about naming an object in DMX, see Identifiers (DMX).

The next line of the code defines the key column for the mining structure, which uniquely identifies an entity in the source data:

```
<key column>
```

The next line of the code is used to define the mining columns that will be used by the mining models associated with the mining structure:

```
<mining structure columns>
```

The next lines of the code define the nested table columns:

```
<table columns>
(  <nested key column>,
   <nested mining structure columns> )
```

For information about the types of mining structure columns that you can define, see Mining Structure Columns.

Note
By default, SQL Server Data Tools (SSDT) creates a 30 percent holdout data set for each mining structure; however, when you use DMX to create a mining structure, you must manually add the holdout data set, if desired.

Lesson Tasks

You will perform the following tasks in this lesson:

- Create a new blank query
- Alter the query to create the mining structure

- Execute the query

Creating the Query

The first step is to connect to an instance of Analysis Services and create a new DMX query in SQL Server Management Studio.

To create a new DMX query in SQL Server Management Studio

1. Open SQL Server Management Studio.
2. In the **Connect to Server** dialog box, for **Server type**, select **Analysis Services**. In **Server name**, type **LocalHost**, or the name of the instance of Analysis Services that you want to connect to for this lesson. Click **Connect**.
3. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.

Altering the Query

The next step is to modify the CREATE MINING STRUCTURE statement described above to create the Market Basket mining structure.

To customize the CREATE MINING STRUCTURE statement

1. In Query Editor, copy the generic example of the CREATE MINING STRUCTURE statement into the blank query.
2. Replace the following:

   ```
   [mining structure name]
   ```

   with:

   ```
   [Market Basket]
   ```
3. Replace the following:

   ```
   <key column>
   ```

   with:

   ```
   OrderNumber TEXT KEY
   ```
4. Replace the following:

   ```
   <table columns>
   (  <nested key column>,
      <nested mining structure columns> )
   ```

   with:

   ```
   [Products] TABLE (
       [Model] TEXT KEY
   )
   ```

The TEXT KEY language specifies that the Model column is the key column for the nested table.

The complete mining structure statement should now be as follows:

```
CREATE MINING STRUCTURE [Market Basket] (
    OrderNumber TEXT KEY,
    [Products] TABLE (
        [Model] TEXT KEY
    )
)
```

5. On the **File** menu, click **Save DMXQuery1.dmx As**.
6. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Market Basket Structure.dmx**.

Executing the Query

The final step is to execute the query. After a query is created and saved, it needs to be executed (that is, the statement needs to be run) in order to create the mining structure on the server. For more information about executing queries in Query Editor, see SQL Server Management Studio Transact SQL Query.

To execute the query

- In Query Editor, on the toolbar, click **Execute**.

  The status of the query is displayed in the **Messages** tab at the bottom of Query Editor after the statement finishes executing. Messages should display:

  ```
  Executing the query
  Execution complete
  ```

  A new structure named **Market Basket** now exists on the server.

In the next lesson, you will add mining models to the Market Basket mining structure you just created.

Next Lesson

Lesson 2: Adding Mining Models to the Association Mining Structure

Lesson 2: Adding Mining Models to the Market Basket Mining Structure

In this lesson, you will add two mining models to the Market Basket mining structure that you created in Lesson 3: Processing the Market Basket Mining Structure. These mining models will allow you to create predictions.

To predict the types of products that customers tend to purchase at the same time, you will create two mining models using the Microsoft Association Algorithm and two different values for the MINIMUM_PROBABILTY parameter.

MINIMUM_PROBABILTY is a Microsoft Association algorithm parameter that helps to determine the number of rules that a mining model will contain by specifying the minimum probability that a rule must have. For example, setting this value to 0.4 specifies that a rule can be generated only if the combination of products that the rule describes has at least a forty percent probability of occurring.

You will view the effect of changing the MINIMUM_PROBABILTY parameter in a later lesson.

ALTER MINING STRUCTURE Statement

To add a mining model that contains a nested table to a mining structure, you use the ALTER MINING STRUCTURE (DMX) statement. The code in the statement can be broken into the following parts:

- Identifying the mining structure
- Naming the mining model
- Defining the key column
- Defining the input and predictable columns
- Defining the nested table columns
- Identifying the algorithm and parameter changes

The following is a generic example of the **ALTER MINING STRUCTURE** statement that adds a mining model to a structure that includes nested table columns:

```
ALTER MINING STRUCTURE [<Mining Structure Name>]
ADD MINING MODEL [<Mining Model Name>]
(
    [<key column>],
    <mining model column> <usage>,
    <table columns>
    (  [<nested key column>],
       <nested mining model columns> )
) USING <algorithm>( <algorithm parameters> )
```

The first line of the code identifies the existing mining structure to which the mining model will be added:

```
ALTER MINING STRUCTURE [<mining structure name>]
```

The next line of the code names the mining model that will be added to the mining structure:

```
ADD MINING MODEL [<mining model name>]
```

For information about naming an object in Data Mining Extensions (DMX), see Identifiers (DMX).

The next lines of the code define the columns in the mining structure that will be used by the mining model:

```
[<key column>],
<mining model columns> <usage>,
```

You can only use columns that already exist in the mining structure.

The first column in the list of mining model columns must be the key column in the mining structure. However, you do not have to type **KEY** after the key column to specify usage. That is because you have already defined the column as a key when you created the mining structure.

The remaining lines specify the usage of the columns in the new mining model. You can specify that a column in the mining model will be used for prediction by using the following syntax:

```
<column name> PREDICT,
```

If you do not specify usage, you do not have to include a data mining structure column in the list. All columns that are used by the referenced data mining structure are automatically available for use by the mining models that are based on that structure. However, the model will not use the columns for training unless you specify the usage.

The last line in the code defines the algorithm and algorithm parameters that will be used to generate the mining model.

```
) USING <algorithm>( <algorithm parameters> )
```

Lesson Tasks

You will perform the following tasks in this lesson:

- Add an association mining model to the structure using the default probability
- Add an association mining model to the structure using a modified probability

Adding an Association Mining Model to the Structure Using the Default MINIMUM_PROBABILITY

The first task is to add a new mining model to the Market Basket mining structure based on the Microsoft Association algorithm using the default value for MINIMUM_PROBABILITY.

To add an Association mining model

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.

   Note

   To create a DMX query against a specific Analysis Services database,

right-click the database instead of the instance.

2. Copy the generic example of the **ALTER MINING STRUCTURE** statement into the blank query.
3. Replace the following:

   ```
   <mining structure name>
   ```

   with:

   ```
   [Market Basket]
   ```

4. Replace the following:

   ```
   <mining model name>
   ```

   with:

   ```
   [Default Association]
   ```

5. Replace the following:

   ```
   [<key column>],
   <mining model columns>,
   <table columns>
   (  [<nested key column>],
      <nested mining model columns> )
   ```

   with:

   ```
   OrderNumber,
        [Products] PREDICT (
            [Model]
        )
   ```

   In this case, the `[Products]` table has been designated as the predictable column**.** Also, the `[Model]` column is included in the list of nested table columns because it is the key column of the nested table.

   Note
   Remember that a nested key is different from a case key. A case key is a unique identifier of the case, whereas the nested key is an attribute that you want to model.

6. Replace the following:

   ```
   USING <algorithm>( <algorithm parameters> )
   ```

   with:

   ```
   Using Microsoft_Association_Rules
   ```

   The resulting statement should now be as follows:

   ```
   ALTER MINING STRUCTURE [Market Basket]
   ```

```
ADD MINING MODEL [Default Association]
(
    OrderNumber,
    [Products] PREDICT (
        [Model]
    )
)
Using Microsoft_Association_Rules
```

7. On the **File** menu, click **Save DMXQuery1.dmx As**.
8. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Default_Association_Model.dmx**.
9. On the toolbar, click the **Execute** button.

Adding an Association Mining Model to the Structure Changing the Default MINIMUM_PROBABILITY

The next task is to add a new mining model to the Market Basket mining structure based on the Microsoft Association algorithm, and change the default value for MINIMUM_PROBABILITY to 0.01. Changing the parameter will cause the Microsoft Association algorithm to create more rules.

To add an Association mining model

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the **ALTER MINING STRUCTURE** statement into the blank query.
3. Replace the following:

   ```
   <mining structure name>
   ```

   with:

   ```
   Market Basket
   ```
4. Replace the following:

   ```
   <mining model name>
   ```

   with:

   ```
   [Modified Association]
   ```
5. Replace the following:

   ```
   <mining model columns>,
   ```

```
<table columns>
(  [<nested key column>],
   <nested mining model columns> )
```

with:

```
OrderNumber,
[Products] PREDICT (
        [Model]
    )
```

In this case, the `[Products]` table has been designated as the predictable column. Also, the `[MODEL]` column is included in the list because it is the key column in the nested table.

6. Replace the following:

```
USING <algorithm>( <algorithm parameters> )
```

with:

```
USING Microsoft_Association_Rules (Minimum_Probability = 0.1)
```

The resulting statement should now be as follows:

```
ALTER MINING STRUCTURE [Market Basket]
ADD MINING MODEL [Modified Assocation]
(
    OrderNumber,
    [Products] PREDICT (
        [Model]
    )
)
USING Microsoft_Association_Rules (Minimum_Probability = 0.1)
```

7. On the **File** menu, click **Save DMXQuery1.dmx As**.
8. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Modified Association_Model.dmx**.
9. On the toolbar, click the **Execute** button.

In this next lesson you will process the Market Basket mining structure together with its associated mining models.

Next Lesson

Lesson 3: Processing the Association Mining Structure

Lesson 3: Processing the Market Basket Mining Structure

In this lesson, you will use the INSERT INTO statement and the vAssocSeqLineItems and vAssocSeqOrders from the sample database to process the mining structures and mining models that you created in Lesson 1: Creating the Association Mining Structure and Lesson 2: Adding Mining Models to the Association Mining Structure.

When you process a mining structure, Analysis Services reads the source data and builds the structures that support mining models. When you process a mining model, the data defined by the mining structure is passed through the data mining algorithm that you chose. The algorithm searches for trends and patterns, and then stores this information in the mining model. The mining model, therefore, does not contain the actual source data, but instead contains the information that was discovered by the algorithm. For more information about processing mining models, see Processing Data Mining Objects.

You only have to reprocess a mining structure if you change a structure column or change the source data. If you add a mining model to a mining structure that has already been processed, you can use the **INSERT INTO MINING MODEL** statement to train the new mining model on the existing data.

Because the Market Basket mining structure contains a nested table, you will have to define the mining columns to be trained using the nested table structure, and use the **SHAPE** command to define the queries that pull the training data from the source tables.

INSERT INTO Statement

In order to train the Market Basket mining structure and its associated mining models, use the INSERT INTO (DMX) statement. The code in the statement can be broken into the following parts.

- Identifying the mining structure
- Listing the columns in the mining structure
- Defining the training data using **SHAPE**

The following is a generic example of the **INSERT INTO** statement:

```
INSERT INTO MINING STRUCTURE [<mining structure name>]
(
   <mining structure columns>
   [<nested table>]
   ( SKIP, <skipped column> )
)
SHAPE {
  OPENQUERY([<datasource>],'<SELECT statement>') }
APPEND
(
```

```
  {OPENQUERY([<datasource>],'<nested SELECT statement>')
}
RELATE [<case key>] TO [<foreign key>]
) AS [<nested table>]
```

The first line of the code identifies the mining structure that you will train:

```
INSERT INTO MINING STRUCTURE [<mining structure name>]
```

The next lines of the code specify the columns that are defined by the mining structure. You must list each column in the mining structure, and each column must map to a column contained within the source query data. You can use **SKIP** to ignore columns that exist in the source data but do not exist in the mining structure. For more information about how to use **SKIP**, see INSERT INTO (DMX).

```
(
   <mining structure columns>
   [<nested table>]
   ( SKIP, <skipped column> )
)
```

The final lines of the code define the data that will be used to train the mining structure. Because the source data is contained within two tables, you will use **SHAPE** to relate the tables.

```
SHAPE {
  OPENQUERY([<datasource>],'<SELECT statement>') }
APPEND
(
  {OPENQUERY([<datasource>],''<nested SELECT statement>'')
}
RELATE [<case key>] TO [<foreign key>]
) AS [<nested table>]
```

In this lesson, you use **OPENQUERY** to define the source data. For information about other methods of defining a query on the source data, see <source data query>.

Lesson Tasks

You will perform the following task in this lesson:

- Process the Market Basket mining structure

Processing the Market Basket Mining Structure

To process the mining structure by using INSERT INTO

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New**

Query, and then click **DMX**.

Query Editor opens and contains a new, blank query.

2. Copy the generic example of the INSERT INTO statement into the blank query.
3. Replace the following:

```
[<mining structure>]
```

with:

```
Market Basket
```

4. Replace the following:

```
    <mining structure columns>
    [<nested table>]
    ( SKIP, <skipped column> )
```

with:

```
[OrderNumber],
[Products]
(SKIP, [Model])
```

In the statement, **Products** refers to the Products table defined by the SHAPE statement. **SKIP** is used to ignore the Model column, which exists in the source data as a key, but is not used by the mining structure.

5. Replace the following:

```
SHAPE {
  OPENQUERY([<datasource>],'<SELECT statement>') }
APPEND
(
  {OPENQUERY([<datasource>],'<nested SELECT statement>')
}
RELATE [<case key>] TO [<foreign key>]
) AS [<nested table>]
```

with:

```
SHAPE {
  OPENQUERY([Adventure Works DW],'SELECT OrderNumber
            FROM vAssocSeqOrders ORDER BY OrderNumber')}
APPEND
(
  {OPENQUERY([Adventure Works DW],'SELECT OrderNumber, Model
```

```
FROM
    dbo.vAssocSeqLineItems ORDER BY OrderNumber, Model')
}
RELATE OrderNumber to OrderNumber
) AS [Products]
```

The source query references the data source defined in the sample project. It uses this data source to access the vAssocSeqLineItems and vAssocSeqOrders views. These views contain the source data that will be used to train the mining model. If you have not created this project or these views, see Basic Data Mining Tutorial.

Within the **SHAPE** command, you will use **OPENQUERY** to define two queries. The first query defines the parent table, and the second query defines the nested table. The two tables are related using the OrderNumber column, which exists in both tables.

The complete statement should now be as follows:

```
INSERT INTO MINING STRUCTURE [Market Basket]
(
   [OrderNumber],[Products] (SKIP, [Model])
)
SHAPE {
  OPENQUERY([Adventure Works DW],'SELECT OrderNumber
            FROM vAssocSeqOrders ORDER BY OrderNumber')}
APPEND
(
  {OPENQUERY([Adventure Works DW],'SELECT OrderNumber, Model
FROM
    dbo.vAssocSeqLineItems ORDER BY OrderNumber, Model')
}
RELATE OrderNumber to OrderNumber
) AS [Products]
```

6. On the **File** menu, click **Save DMXQuery1.dmx As**.
7. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Process Market Basket.dmx**.
8. On the toolbar, click the **Execute** button.

After the query has finished running, you can view the patterns and itemsets that were found, view associations, or filter by itemset, probability, or importance. To view this

information, in SQL Server Management Studio, right-click the name of the data model, and then click **Browse**.

In the next lesson, you will create several predictions based on the mining models that you added to the Market Basket structure.

Next Lesson

Lesson 4: Creating Association Predictions

Lesson 4: Executing Market Basket Predictions

In this lesson, you will use the DMX **SELECT** statement to create predictions based on the association models you created in Lesson 2: Adding Mining Models to the Market Basket Mining Structure. A prediction query is created by using the DMX **SELECT** statement and adding a **PREDICTION JOIN** clause. For more information about the syntax of a prediction join, see SELECT FROM <model> PREDICTION JOIN (DMX).

The **SELECT FROM <model> PREDICTION JOIN** form of the **SELECT** statement contains three parts:

- A list of the mining model columns and prediction functions that are returned in the result set. This list can also contain input columns from the source data.
- A source query that defines the data that is being used to create a prediction. For example, if you are creating many predictions in a batch, the source query could retrieve a list of customers.
- A mapping between the mining model columns and the source data. If the columns names match, you can use the **NATURAL PREDICTION JOIN** syntax and omit the column mappings.

You can enhance the query by using prediction functions. Prediction functions provide additional information, such as the probability of a prediction occurring, or the support for a prediction in the training dataset. For more information about prediction functions, see Functions (DMX).

You can also use the prediction query builder in SQL Server Data Tools (SSDT) to create prediction queries. For more information, see Using the Prediction Query Builder to Create DMX Prediction Queries.

Singleton PREDICTION JOIN Statement

The first step is to create a singleton query, by using the **SELECT FROM <model> PREDICTION JOIN** syntax and supplying a single set of values as input. The following is a generic example of the singleton statement:

```
SELECT <select list>
    FROM [<mining model>]
[NATURAL] PREDICTION JOIN
(SELECT '<value>' AS [<column>],
```

```
    (SELECT 'value' AS [<nested column>] UNION
        SELECT 'value' AS [<nested column>] ...)
    AS [<nested table>])
AS [<input alias>]
```

The first line of the code defines the columns from the mining model that the query returns, and specifies the name of the mining model used to generate the prediction:

```
SELECT <select list> FROM [<mining model>]
```

The next line of the code indicates the operation to perform. Because you will specify values for each of the columns and type the column names exactly so as to match the model, you can use the **NATURAL PREDICTION JOIN** syntax. However, if the column names were different, you would have to specify mappings between the columns in the model and the columns in the new data by adding an **ON** clause.

```
[NATURAL] PREDICTION JOIN
```

The next lines of the code define the products in the shopping cart that will be used to predict additional products that a customer will add:

```
(SELECT '<value>' AS [<column>],
    (SELECT 'value' AS [<nested column>] UNION
        SELECT 'value' AS [<nested column>] ...)
    AS [<nested table>])
```

Lesson Tasks

You will perform the following tasks in this lesson:

- Create a query that predicts what other items a customer will likely purchase, based on items already existing in their shopping cart. You will create this query by using the mining model with the default MINIMUM_PROBABILITY.
- Create a query that predicts what other items a customer will likely purchase based on items already existing in their shopping cart. This query is based on a different model, in which MINIMUM_PROBABILITY has been set to 0.01. Because the default value for MINIMUM_PROBABILITY in association models is 0.3, the query on this model should return more possible items than the query on the default model.

Create a Prediction by Using a Model with the Default MINIMUM_PROBABILITY

To create an association query

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX** to open the Query Editor.
2. Copy the generic example of the **PREDICTION JOIN** statement into the blank query.

3. Replace the following:

```
<select list>
```

with:

```
PREDICT([Default
Association].[Products],INCLUDE_STATISTICS,3)
```

You could just include the column name [Products], but by using the Predict function, you can limit the number of products that are returned by the algorithm to three. You can also use **INCLUDE_STATISTICS**, which returns the support, probability, and adjusted probability for each product. These statistics help you rate the accuracy of the prediction.

4. Replace the following:

```
[<mining model>]
```

with:

```
[Default Association]
```

5. Replace the following:

```
(SELECT '<value>' AS [<column>],
    (SELECT 'value' AS [<nested column>] UNION
        SELECT 'value' AS [<nested column>] ...)
    AS [<nested table>])
```

with:

```
(SELECT (SELECT 'Mountain Bottle Cage' AS [Model]
  UNION SELECT 'Mountain Tire Tube' AS [Model]
  UNION SELECT 'Mountain-200' AS [Model]) AS [Products]) AS t
```

This statement uses the **UNION** statement to specify three products that must be included in the shopping cart together with the predicted products. The Model column in the **SELECT** statement corresponds to the model column that is contained in the nested products table.

The complete statement should now be as follows:

```
SELECT
  PREDICT([Default
Association].[Products],INCLUDE_STATISTICS,3)
From
  [Default Association]
NATURAL PREDICTION JOIN
(SELECT (SELECT 'Mountain Bottle Cage' AS [Model]
  UNION SELECT 'Mountain Tire Tube' AS [Model]
```

```
UNION SELECT 'Mountain-200' AS [Model]) AS [Products]) AS t
```

6. On the **File** menu, click **Save DMXQuery1.dmx As**.
7. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Association Prediction.dmx**.
8. On the toolbar, click the **Execute** button.

   The query returns a table that contains three products: HL Mountain Tire, Fender Set - Mountain, and ML Mountain Tire. The table lists these returned products in order of probability. The returned product that is most likely to be included in the same shopping cart as the three products specified in the query appears at the top of the table. The two products that follow are the next most likely to be included in the shopping cart. The table also contains statistics describing the accuracy of the prediction.

Create a Prediction by Using a Model with a MINIMUM_PROBABILITY of 0.01

To create an association query

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX** to open the Query Editor.
2. Copy the generic example of the **PREDICTION JOIN** statement into the blank query.
3. Replace the following:

   ```
   <select list>
   ```

   with:

   ```
   PREDICT([Modified
   Association].[Products],INCLUDE_STATISTICS,3)
   ```

4. Replace the following:

   ```
   [<mining model>]
   ```

   with:

   ```
   [Modified Association]
   ```

5. Replace the following:

   ```
   (SELECT '<value>' AS [<column>],
       (SELECT 'value' AS [<nested column>] UNION
           SELECT 'value' AS [<nested column>] ...)
       AS [<nested table>])
   ```

   with:

   ```
   (SELECT (SELECT 'Mountain Bottle Cage' AS [Model]
   ```

```
  UNION SELECT 'Mountain Tire Tube' AS [Model]
  UNION SELECT 'Mountain-200' AS [Model]) AS [Products]) AS t
```

This statement uses the **UNION** statement to specify three products that must be included in the shopping cart together with the predicted products. The `[Model]` column in the **SELECT** statement corresponds to the column in the nested products table.

The complete statement should now be as follows:

```
SELECT
  PREDICT([Modified
Association].[Products],INCLUDE_STATISTICS,3)
From
  [Modified Association]
NATURAL PREDICTION JOIN
(SELECT (SELECT 'Mountain Bottle Cage' AS [Model]
  UNION SELECT 'Mountain Tire Tube' AS [Model]
  UNION SELECT 'Mountain-200' AS [Model]) AS [Products]) AS t
```

6. On the **File** menu, click **Save DMXQuery1.dmx As**.
7. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Modified Association Prediction.dmx**.
8. On the toolbar, click the **Execute** button.

   The query returns a table that contains three products: HL Mountain Tire, Water Bottle, and Fender Set - Mountain. The table lists these products in order of probability. The product that appears at the top of the table is the product that is most likely to be included in the same shopping cart as the three products specified in the query. The remaining products are the next most likely to be included in the shopping cart. The table also contains statistics that describe the accuracy of the prediction.

   You can see from the results of this query that the value of the MINIMUM_PROBABILITY parameter affects the results returned by the query.

This is the last step in the Market Basket tutorial. You now have a set of models that you can use to predict the products that customers might purchase at the same time.

To learn how to use DMX in another predictive scenario, see Bike Buyer DMX Tutorial.

See Also

Querying an Association Model (Analysis Services - Data Mining)

Creating DMX Prediction Queries

Time Series Prediction DMX Tutorial

In this tutorial, you will learn how to create a time series mining structure, create three custom time series mining models, and then make predictions by using those models.

The mining models are based on the data contained in the sample database, which stores data for the fictitious company Adventure Works Cycles. Adventure Works Cycles is a large, multinational manufacturing company.

Tutorial Scenario

Adventure Works Cycles has decided to use data mining to generate sales projections. They have already built some regional forecasting models; for more information, see Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial). However, the Sales Department needs to be able to periodically update the data mining model with new sales data. They also want to customize the models to provide different projections.

Microsoft SQL Server Analysis Services provides several tools that can be used to accomplish this task:

- The Data Mining Extensions (DMX) query language
- The Microsoft Time Series Algorithm
- Query Editor in SQL Server Management Studio

The Microsoft Time Series algorithm creates models that can be used for prediction of time-related data. Data Mining Extensions (DMX) is a query language provided by Analysis Services that you can use to create mining models and prediction queries.

What You Will Learn

This tutorial assumes that you are already familiar with the objects that Analysis Services uses to create mining models. If you have not previously created a mining structure or mining model by using DMX, see Bike Buyer DMX Tutorial.

This tutorial is divided into the following lessons:

Creating a Time Series Mining Structure

In this lesson, you will learn how to use the **CREATE MINING MODEL** statement to add a new forecasting model and a related mining model.

Adding Mining Models to the Time Series Mining Structure

In this lesson, you will learn how to use the ALTER MINING STRUCTURE statement to add new mining models to the time series structure. You will also learn how to customize the algorithm used for analyzing a time series.

Lesson 3: Processing the Time Series Structure and Models

In this lesson, you will learn how to train the models by using the **INSERT INTO** statement and populating the structure with data from the database.

Creating Time Series Predictions Using DMX

In this lesson, you will learn how to create time series predictions.

Extending the Time Series Model

In this lesson, you will learn how to use the **EXTEND_MODEL_CASES** parameter to update the model with new data when you make predictions.

Requirements

Before doing this tutorial, make sure that the following are installed:

- Microsoft SQL Server
- Microsoft SQL Server Analysis Services
- The database

By default, the sample databases are not installed, to enhance security. To install the official sample databases for Microsoft SQL Server, go to http://www.CodePlex.com/MSFTDBProdSamples or on the Microsoft SQL Server Samples and Community Projects home page in the section Microsoft SQL Server Product Samples. Click **Databases**, then click the **Releases** tab and select the databases that you want.

Note

When you review tutorials, we recommend that you add **Next topic** and **Previous topic** buttons to the document viewer toolbar. For more information, see Adding Next and Previous Buttons to Help.

See Also

Data Mining Tutorial

Intermediate Data Mining Tutorial (Analysis Services - Data Mining)

Lesson 1: Creating a Time Series Mining Model and Mining Structure

In this lesson, you will create a mining model that allows you to predict values over time, based on historical data. When you create the model, the underlying structure will be generated automatically and can be used as the basis for additional mining models.

This lesson assumes that you are familiar with forecasting models and with the requirements of the Microsoft Time Series algorithm. For more information, see Microsoft Time Series Algorithm (Analysis Services - Data Mining).

CREATE MINING MODEL Statement

In order to create a mining model directly and automatically generate the underlying mining structure, you use the CREATE MINING MODEL (DMX) statement. The code in the statement can be broken into the following parts:

- Naming the model

- Defining the time stamp
- Defining the optional series key column
- Defining the predictable attribute or attributes

The following is a generic example of the CREATE MINING MODEL statement:

```
CREATE MINING MODEL [<Mining Structure Name>]
(
   <key columns>,
   <predictable attribute columns>
)
USING <algorithm name>([parameter list])
WITH DRILLTHROUGH
```

The first line of the code defines the name of the mining model:

```
CREATE MINING MODEL [Mining Model Name]
```

Analysis Services automatically generates a name for the underlying structure, by appending "_structure" to the model name, which ensures that the structure name is unique from the model name. For information about naming an object in DMX, see Identifiers (DMX).

The next line of the code defines the key column for the mining model, which in the case of a time series model uniquely identifies a time step in the source data. The time step is identified with the **KEY TIME** keywords after the column name and data types. If the time series model has a separate series key, it is identified by using the **KEY** keyword.

```
<key columns>
```

The next line of the code is used to define the columns in the model that will be predicted. You can have multiple predictable attributes in a single mining model. When there are multiple predictable attributes, the Microsoft Time Series algorithm generates a separate analysis for each series:

```
<predictable attribute columns>
```

Lesson Tasks

You will perform the following tasks in this lesson:

- Create a new blank query
- Alter the query to create the mining model
- Execute the query

Creating the Query

The first step is to connect to an instance of Analysis Services and create a new DMX query in SQL Server Management Studio.

To create a new DMX query in SQL Server Management Studio

1. Open SQL Server Management Studio.
2. In the **Connect to Server** dialog box, for **Server type**, select **Analysis Services**. In **Server name**, type **LocalHost**, or the name of the instance of Analysis Services that you want to connect to for this lesson. Click **Connect**.
3. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.

Altering the Query

The next step is to modify the CREATE MINING MODEL statement to create the mining model used for forecasting, together with its underlying mining structure.

To customize the CREATE MINING MODEL statement

1. In Query Editor, copy the generic example of the CREATE MINING MODEL statement into the blank query.
2. Replace the following:

   ```
   [mining model name]
   ```

   with:

   ```
   [Forecasting_MIXED]
   ```
3. Replace the following:

   ```
   <key columns>
   ```

   with:

   ```
   [Reporting Date] DATE KEY TIME,
   [Model Region] TEXT KEY
   ```

   The **TIME KEY** keyword indicates that the ReportingDate column contains the time step values used to order the values. Time steps can be dates and times, integers, or any ordered data type, so long as the values are unique and the data is sorted.

   The **TEXT** and **KEY** keywords indicate that the ModelRegion column contains an additional series key. You can have only one series key, and the values in the column must be distinct.
4. Replace the following:

   ```
   < predictable attribute columns> )
   ```

   with:

   ```
   [Quantity] LONG CONTINUOUS PREDICT,
   [Amount] DOUBLE CONTINUOUS PREDICT
   )
   ```

5. Replace the following:

```
USING <algorithm name>([parameter list])
WITH DRILLTHROUGH
```

with:

```
USING Microsoft_Time_Series(AUTO_DETECT_PERIODICITY = 0.8,
FORECAST_METHOD = 'MIXED')
WITH DRILLTHROUGH
```

The algorithm parameter, **AUTO_DETECT_PERIODICITY** = 0.8, indicates that you want the algorithm to detect cycles in the data. Setting this value closer to 1 favors the discovery of many patterns but can slow processing.

The algorithm parameter, **FORECAST_METHOD**, indicates whether you want the data to be analyzed using ARTXP, ARIMA, or a mixture of both.

The keyword, **WITH DRILLTHROUGH**, specify that you want to be able to view detailed statistics in the source data after the model is complete. You must add this clause if you want to browse the model by using the Microsoft Time Series Viewer. It is not required for prediction.

The complete statement should now be as follows:

```
CREATE MINING MODEL [Forecasting_MIXED]
     (
    [Reporting Date] DATE KEY TIME,
    [Model Region] TEXT KEY,
    [Quantity] LONG CONTINUOUS PREDICT,
    [Amount] DOUBLE CONTINUOUS PREDICT
    )
USING Microsoft_Time_Series (AUTO_DETECT_PERIODICITY = 0.8,
FORECAST_METHOD = 'MIXED')
WITH DRILLTHROUGH
```

6. On the **File** menu, click **Save DMXQuery1.dmx As**.
7. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Forecasting_MIXED.dmx**.

Executing the Query

The final step is to execute the query. After a query is created and saved, it needs to be executed to create the mining model and its mining structure on the server. For more information about executing queries in Query Editor, see SQL Server Management Studio Transact SQL Query.

To execute the query

- In Query Editor, on the toolbar, click **Execute**.

  The status of the query is displayed in the **Messages** tab at the bottom of Query Editor after the statement finishes executing. Messages should display:

  ```
  Executing the query

  Execution complete
  ```

  A new structure named **Forecasting_MIXED_Structure** now exists on the server, together with the related mining model **Forecasting_MIXED**.

In the next lesson, you will add a mining model to the **Forecasting_MIXED** mining structure that you just created.

Next Lesson

Adding Mining Models to the Time Series Mining Structure

See Also

Mining Model Content for Time Series Models (Analysis Services - Data Mining)

Microsoft Time Series Algorithm Technical Reference (Analysis Services - Data Mining)

Lesson 2: Adding Mining Models to the Time Series Mining Structure

In this lesson, you will add a new mining model to the mining structure that you just created in Creating a Time Series Mining Structure.

ALTER MINING STRUCTURE Statement

In order to add a new mining model to an existing mining structure, you use the ALTER MINING STRUCTURE (DMX) statement. The code in the statement can be broken into the following parts:

- Identifying the mining structure
- Naming the mining model
- Defining the key column
- Defining the predictable columns
- Specifying the algorithm and any parameter changes

The following is a generic example of the ALTER MINING STRUCTURE statement:

```
ALTER MINING STRUCTURE [<mining structure name>]
ADD MINING MODEL [<mining model name>]
   ([<key columns>],
    <mining model columns>
   )
```

```
USING <algorithm name>([<algorithm parameters>])
[WITH DRILLTHROUGH]
```

The first line of the code identifies the existing mining structure to which the mining models will be added:

```
ALTER MINING STRUCTURE [<mining structure name>]
```

The next line of the code names the mining model that will be added to the mining structure:

```
ADD MINING MODEL [<mining model name>]
```

For information about naming an object in DMX, see Identifiers (DMX).

The next lines of the code define columns from the mining structure that will be used by the mining model:

```
[<key columns>],
<mining model columns>
```

You can only use columns that already exist in the mining structure, and the first column in the list must be the key column from the mining structure.

The next lines of the code defines the mining algorithm that generates the mining model and the algorithm parameters that you can set on the algorithm, and specify whether you can drill down from the mining model into view detailed data in the training cases:

```
USING <algorithm name>([<algorithm parameters>])
WITH DRILLTHROUGH
```

For more information about the algorithm parameters that you can adjust, see Microsoft Time Series Algorithm Technical Reference (Analysis Services - Data Mining).

You can specify that a column in the mining model be used for prediction by using the following syntax:

```
<mining model column> PREDICT
```

Lesson Tasks

You will perform the following tasks in this lesson:

- Add a new time series mining model to the structure.
- Change the algorithm parameters to use a different method of analysis and prediction

Adding an ARIMA Time Series Model to the Structure

The first step is to add a new forecasting mining model to the existing structure. By default, the Microsoft Time Series algorithm creates time series mining models by using two algorithms, ARIMA and ARTXP, and blending the results. However, you can specify a single algorithm to use, or you can specify the exact blend of algorithms. In this step, you will add a new model that uses only the ARIMA algorithm. This algorithm is optimized for long-term prediction.

To add an ARIMA time series mining model

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX** to open Query Editor and a new, blank query.
2. Copy the generic example of the ALTER MINING STRUCTURE statement into the blank query.
3. Replace the following:

   ```
   <mining structure name>
   ```

   with:

   ```
   [Forecasting_MIXED_Structure]
   ```
4. Replace the following:

   ```
   <mining model name>
   ```

   with:

   ```
   Forecasting_ARIMA
   ```
5. Replace the following:

   ```
   <key columns>,
   ```

   with:

   ```
   [ReportingDate],
   [ModelRegion]
   ```

   Note that you do not need to repeat any of the date type or content type information that you provided in the CREATE MINING MODEL statement, because this information is already stored in the mining structure.
6. Replace the following:

   ```
   <mining model columns>
   ```

   with:

   ```
   ([Quantity] PREDICT,
   [Amount] PREDICT
   )
   ```
7. Replace the following:

   ```
   USING <algorithm name>([<algorithm parameters>])
   [WITH DRILLTHROUGH]
   ```

   with:

   ```
   USING Microsoft_Time_Series (AUTO_DETECT_PERIODICITY = .08,
   FORECAST_METHOD = 'ARIMA')
   WITH DRILLTHROUGH
   ```

The resulting statement should now be as follows:

```
ALTER MINING STRUCTURE [Forecasting_MIXED_Structure]
ADD MINING MODEL [Forecasting_ARIMA]
   (
   ([ReportingDate],
    [ModelRegion],
    ([Quantity] PREDICT,
    [Amount] PREDICT
   )
USING Microsoft_Time_Series (AUTO_DETECT_PERIODICITY = .08,
FORECAST_METHOD = 'ARIMA')
WITH DRILLTHROUGH
```

8. On the **File** menu, click **Save DMXQuery1.dmx As**.
9. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Forecasting_ARIMA.dmx**.
10. On the toolbar, click the **Execute** button.

Adding an ARTXP Time Series Model to the Structure

The ARTXP algorithm was the default time series algorithm in SQL Server 2005 and is optimized for short-term prediction. To compare predictions by using all three time series algorithms, you will add one more model that is based on the ARTXP algorithm.

To add an ARTXP time series mining model

1. Copy the following code into a blank query window.

   Note that you do not need to change anything except the name of the new mining model, and the value of the FORECAST_METHOD parameter.

```
ALTER MINING STRUCTURE [Forecasting_MIXED_Structure]
ADD MINING MODEL [Forecasting_ARTXP]
   (
   ([ReportingDate],
    [ModelRegion],
    ([Quantity] PREDICT,
    [Amount] PREDICT
   )
USING Microsoft_Time_Series (AUTO_DETECT_PERIODICITY = .08,
FORECAST_METHOD = 'ARTXP')
```

```
WITH DRILLTHROUGH
```

2. On the **File** menu, click **Save DMXQuery1.dmx As**.
3. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Forecasting_ARTXP.dmx**.
4. On the toolbar, click the **Execute** button.

In the next lesson, you will process all of the models and the mining structure.

Next Lesson

Lesson 3: Processing the Time Series Structure and Models

See Also

Microsoft Time Series Algorithm (Analysis Services - Data Mining)

Microsoft Time Series Algorithm Technical Reference (Analysis Services - Data Mining)

Lesson 3: Processing the Time Series Structure and Models

In this lesson, you will use the INSERT INTO statement to process the time series mining structures and mining models that you created.

When you process a mining structure, Analysis Services reads the source data and builds the structures that support mining models. You always have to process a mining model and structure when you first create it. If you specify the mining structure when using INSERT INTO, the statement processes the mining structure and all its associated mining models.

When you add a mining model to a mining structure that has already been processed, you can use the **INSERT INTO MINING MODEL** statement to process just the new mining model by using the existing data.

For more information about processing mining models, see Processing Data Mining Objects.

INSERT INTO Statement

In order to train the time series mining structure and all its associated mining models, use the INSERT INTO (DMX) statement. The code in the statement can be broken into the following parts.

- Identifying the mining structure
- Listing the columns in the mining structure
- Defining the training data

The following is a generic example of the **INSERT INTO** statement:

```
INSERT INTO MINING STRUCTURE [<mining structure name>]
(
   <mining structure columns>
)
```

```
OPENQUERY (<source data definition>)
```

The first line of the code identifies the mining structure that you will train:

```
INSERT INTO MINING STRUCTURE [<mining structure name>]
```

The next lines of the code specify the columns that are defined by the mining structure. You must list each column in the mining structure, and each column must map to a column contained within the source query data.

```
(
   <mining structure columns>
)
```

The final lines of the code define the data that will be used to train the mining structure.

```
OPENQUERY (<source data definition>)
```

In this lesson, you use **OPENQUERY** to define the source data. For more information about other methods of defining a query on the source data, see <source data query>.

Lesson Tasks

You will perform the following task in this lesson:

- Process the mining structure Forecasting_MIXED_Structure
- Process the related mining models Forecasting_MIXED, Forecasting_ARIMA, and Forecasting_ARTXP

Processing the Time Series Mining Structure

To process the mining structure and related mining models by using INSERT INTO

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the INSERT INTO statement into the blank query.
3. Replace the following:

   ```
   [<mining structure>]
   ```

   with:

   ```
   Forecasting_MIXED_Structure
   ```
4. Replace the following:

   ```
      <mining structure columns>
   ```

   with:

   ```
   [ReportingDate],
   [ModelRegion]
   ```
5. Replace the following:

```
OPENQUERY(<source data definition>)
```

with:

```
OPENQUERY([Adventure Works DW 2008R2],'SELECT
[ReportingDate], [ModelRegion], [Quantity], [Amount]
FROM vTimeSeries ORDER BY [ReportingDate]')
```

The source query references the data source defined in the IntermediateTutorial sample project. It uses this data source to access the view vTimeSeries. This view contains the source data that will be used to train the mining model. If you are not familiar with this project or this views, see Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial).

The complete statement should now be as follows:

```
INSERT INTO MINING STRUCTURE [Forecasting_MIXED_Structure]
(
   [ReportingDate],[ModelRegion],[Quantity],[Amount])
)
OPENQUERY(
[Adventure Works DW 2008R2],
'SELECT [ReportingDate],[ModelRegion],[Quantity],[Amount]
FROM vTimeSeries ORDER BY [ReportingDate]'
)
```

6. On the **File** menu, click **Save DMXQuery1.dmx As**.
7. In the **Save As** dialog box, browse to the appropriate folder, and name the file **ProcessForecastingAll.dmx**.
8. On the toolbar, click the **Execute** button.

After the query has finished running, you can create predictions by using the processed mining models. In the next lesson, you will create several predictions based on the mining models that you created.

Next Lesson

Lesson 4: Creating Time Series Predictions Using DMX

See Also

Processing Data Mining Objects

<source data query>

OPENQUERY (DMX)

Lesson 4: Creating Time Series Predictions Using DMX

In this lesson and the following lesson, you will use Data Mining Extensions (DMX) to create different types of predictions based on the time series models that you created in Lesson 1: Creating a Time Series Mining Model and Mining Structure and Lesson 2: Adding Mining Models to the Time Series Mining Structure.

With a time series model, you have many options for making predictions:

- Use the existing patterns and data in the mining model
- Use the existing patterns in the mining model but supply new data
- Add new data to the model or update the model.

The syntax for making these prediction types is summarized below:

Default time series prediction

Use PredictTimeSeries (DMX) to return the specified number of predictions from the trained mining model.

For example, see PredictTimeSeries (DMX) or Querying a Time Series Model (Analysis Services - Data Mining).

EXTEND_MODEL_CASES

Use PredictTimeSeries (DMX) with the EXTEND_MODEL_CASES argument to add new data, extend the series, and create predictions based on the updated mining model.

This tutorial contains an example of how to use EXTEND_MODEL_CASES.

REPLACE_MODEL_CASES

Use PredictTimeSeries (DMX) with the REPLACE_MODEL_CASES argument to replace the original data with a new data series, and then create predictions based on applying the patterns in the mining model to the new data series.

For an example of how to use REPLACE_MODEL_CASES, see Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial).

Lesson Tasks

You will perform the following tasks in this lesson:

- Create a query to get the default predictions based on existing data.

In the following lesson you will perform the following related tasks:

- Create a query to supply new data and get updated predictions.

In addition to creating queries manually by using DMX, you can also create predictions by using the prediction query builder in SQL Server Data Tools (SSDT). For more information, see Using the Prediction Query Builder to Create DMX Prediction Queries or Mining Model Prediction Tab: How-to Topics.

Simple Time Series Prediction Query

The first step is to use the **SELECT FROM** statement together with the **PredictTimeSeries** function to create time series predictions. Time series models

support a simplified syntax for creating predictions: you do not need to supply any inputs, but only have to specify the number of predictions to create. The following is a generic example of the statement you will use:

```
SELECT <select list>
FROM [<mining model name>]
WHERE [<criteria>]
```

The select list can contain columns from the model, such as the name of the product line that you are creating the predictions for, or prediction functions, such as Lag (DMX) or PredictTimeSeries (DMX), which are specifically for time series mining models.

To create a simple time series prediction query

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the statement into the blank query.
3. Replace the following:

   ```
   <select list>
   ```

   with:

   ```
   [Forecasting_MIXED].[ModelRegion],
   PredictTimeSeries([Forecasting_MIXED].[Quantity],6) AS
   PredictQty,
   PredictTimeSeries ([Forecasting_MIXED].[Amount],6) AS
   PredictAmt
   ```

   The first line retrieves a value from the mining model that identifies the series.

   The second and third lines use the **PredictTimeSeries** function. Each line predicts a different attribute, `[Quantity]` or `[Amount]`. The numbers after the names of the predictable attributes specify the number of time steps to predict.

   The **AS** clause is used to provide a name for the column that is returned by each prediction function. If you do not supply an alias, by default both columns are returned with the label, `Expression`.
4. Replace the following:

   ```
   [<mining model>]
   ```

   with:

   ```
   [Forecasting_MIXED]
   ```
5. Replace the following:

   ```
   WHERE [criteria>]
   ```

with:

```
WHERE [ModelRegion] = 'M200 Europe' OR
[ModelRegion] = 'M200 Pacific'
```

The complete statement should now be as follows:

```
SELECT
[Forecasting_MIXED].[ModelRegion],
PredictTimeSeries([Forecasting_MIXED].[Quantity],6) AS
PredictQty,
PredictTimeSeries ([Forecasting_MIXED].[Amount],6) AS
PredictAmt
FROM
[Forecasting_MIXED]
WHERE [ModelRegion] = 'M200 Europe' OR
[ModelRegion] = 'M200 Pacific'
```

6. On the **File** menu, click **Save DMXQuery1.dmx As**.
7. In the **Save As** dialog box, browse to the appropriate folder, and name the file **SimpleTimeSeriesPrediction.dmx**.
8. On the toolbar, click the **Execute** button.

   The query returns 6 predictions for each of the two combinations of product and region that are specified in the **WHERE** clause.

In the next lesson, you will create a query that supplies new data to the model, and compare the results of that prediction with the one you just created.

Next Task in Lesson

Lesson 5: Extending the Time Series Model

See Also

PredictTimeSeries (DMX)

Lag (DMX)

Querying a Time Series Model (Analysis Services - Data Mining)

Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial)

Lesson 5: Extending the Time Series Model

In SQL Server 2012 Enterprise, you can add new data to a time series model and automatically incorporate the new data into the model. You add new data to a time series mining model in one of two ways:

- Use a PREDICTION JOIN to join data in an external source to the training data.
- Use a singleton prediction query to provide data one slice at a time.

For example, assume that you trained the mining model on existing sales data some months ago. When you get new sales, you might want to update the sales predictions to incorporate the new data. You can do this in one step, by supplying the new sales figures as input data and generating new predictions based on the composite data set.

Making Predictions with EXTEND_MODEL_CASES

The following are generic examples of a time series prediction using EXTEND_MODEL_CASES. The first example enables you to specify the number of predictions starting from the last time step of the original model:

```
SELECT [<model columns>,] PredictTimeSeries(<table column reference>,
n, EXTEND_MODEL_CASES)
FROM <mining model>
PREDICTION JOIN <source query>
[WHERE <criteria>]
```

The second example enables you to specify the time step where predictions should start, and where they should end. This option is important when you extend the model cases because, by default, the time steps used for prediction queries always start at the end of the original series.

```
SELECT [<model columns>,] PredictTimeSeries(<table column reference>,
n-start, n-end, EXTEND_MODEL_CASES)
FROM <mining model>
PREDICTION JOIN <source query>
[WHERE <criteria>}
```

In this tutorial, you will create both kinds of queries.

To create a singleton prediction query on a time series model

1. In **Object Explorer**, right-click the instance of Analysis Services, point to **New Query**, and then click **DMX**.

   Query Editor opens and contains a new, blank query.
2. Copy the generic example of the singleton statement into the blank query.
3. Replace the following:

   ```
   SELECT [<model columns>,] PredictTimeSeries(<table column
   reference>, n, EXTEND_MODEL_CASES)
   ```

   with:

   ```
   SELECT [Model Region],
   PredictTimeSeries([Quantity],6, EXTEND_MODEL_CASES) AS
   PredictQty
   ```

   The first line retrieves a value from the model that identifies the series.

The second line contains the prediction function, which gets 6 predictions for Quantity. An alias, `PredictQty`, is assigned to the prediction result column to make it easier to understand the results.

4. Replace the following:

```
FROM <mining model>
```

with:

```
FROM [Forecasting_MIXED]
```

5. Replace the following:

```
PREDICTION JOIN <source query>
```

with:

```
NATURAL PREDICTION JOIN
(
   SELECT 1 AS [Reporting Date],
   '10' AS [Quantity],
   'M200 Europe' AS [Model Region]
   UNION SELECT
   2 AS [Reporting Date],
   15 AS [Quantity]),
   'M200 Europe' AS [Model Region]
) AS t
```

6. Replace the following:

```
[WHERE <criteria>]
```

with:

```
WHERE [ModelRegion] = 'M200 Europe' OR
[ModelRegion] = 'M200 Pacific'
```

The complete statement should now be as follows:

```
SELECT [Model Region],
PredictTimeSeries([Quantity],6, EXTEND_MODEL_CASES) AS
PredictQty
FROM
   [Forecasting_MIXED]
NATURAL PREDICTION JOIN
   SELECT 1 AS [ReportingDate],
  '10' AS [Quantity],
```

```
  'M200 Europe' AS [ModelRegion]
UNION SELECT
  2 AS [ReportingDate],
  15 AS [Quantity]),
  'M200 Europe' AS [ModelRegion]
) AS t
WHERE [ModelRegion] = 'M200 Europe' OR
[ModelRegion] = 'M200 Pacific'
```

7. On the **File** menu, click **Save DMXQuery1.dmx As**.
8. In the **Save As** dialog box, browse to the appropriate folder, and name the file **Singleton_TimeSeries_Query.dmx**.
9. On the toolbar, click the **Execute** button.

   The query returns predictions of sales quantity for the M200 bicycle in the Europe and Pacific regions.

Understanding Prediction Start with EXTEND_MODEL_CASES

Now that you have created predictions based on the original model, and with new data, you can compare the results to see how updating the sales data affects the predictions. Before you do so, review the code that you just created, and notice the following:

- You supplied new data for only the Europe region.
- You supplied only two months' worth of new data.

The following table shows how the new values supplied for M200 Europe affect predictions. You did not provide any new data for the M200 product in the Pacific region, but this series is presented for comparison:

Product and Region	**Existing model (**PredictTimeSeries**)**	**Model with updated sales data (**PredictTimeSeries **with** EXTEND_MODEL_CASES**)**
M200 Europe	M200 Europe, 7/25/2008 12:00:00 AM, 77 M200 Europe, 8/25/2008 12:00:00 AM, 64 M200 Europe, 9/25/2008 12:00:00 AM, 59 M200, 10/25/2008, 56	M200 Europe, 7/25/2008 12:00:00 AM, 10 M200 Europe, 8/25/2008 12:00:00 AM, 15 M200 Europe, 9/25/2008 12:00:00 AM, 72 M200, 10/25/2008, 69

Product and Region	Existing model (PredictTimeSeries)			Model with updated sales data (PredictTimeSeries with EXTEND_MODEL_CASES)		
	Europe	12:00:00 AM		Europe	12:00:00 AM	
	M200 Europe	11/25/2008 12:00:00 AM	56	M200 Europe	11/25/2008 12:00:00 AM	68
	M200 Europe	12/25/2008 12:00:00 AM	74	M200 Europe	12/25/2008 12:00:00 AM	89
M200 Pacific						
	M200 Pacific	7/25/2008 12:00:00 AM	41	M200 Pacific	7/25/2008 12:00:00 AM	41
	M200 Pacific	8/25/2008 12:00:00 AM	44	M200 Pacific	8/25/2008 12:00:00 AM	44
	M200 Pacific	9/25/2008 12:00:00 AM	38	M200 Pacific	9/25/2008 12:00:00 AM	38
	M200 Pacific	10/25/2008 12:00:00 AM	41	M200 Pacific	10/25/2008 12:00:00 AM	41
	M200 Pacific	11/25/2008 12:00:00 AM	36	M200 Pacific	11/25/2008 12:00:00 AM	36
	M200 Pacific	12/25/2008 12:00:00 AM	39	M200 Pacific	12/25/2008 12:00:00 AM	39

From these results, you can see two things:

- The first two predictions for the M200 Europe series are exactly the same as the new data you supplied. By design, Analysis Services returns the actual new data points instead of making a prediction. That is because when you extend the model cases, the time steps used for prediction queries always start at the end of the original series. Therefore, if you add two new data points, the first two predictions returned overlap with the new data.

- After all the new data points are used up, Analysis Services makes predictions based on the updated model. Therefore, starting in September 2005, you can see the difference between predictions for M200 Europe from the original model, in the left-hand column, and the model that uses EXTEND_MODEL_CASES, in the right-hand column. The predictions are different because the model has been updated with the new data.

Using Start and End Time Steps to Control Predictions

When you extend a model, the new data is always attached to the end of the series. However, for the purpose of prediction, the time slices used for prediction queries start at the end of the original series. If you want to obtain only the new predictions when you add the new data, you must specify the starting point as a number of time slices. For example, if you are adding two new data points and want to make four new predictions, you would do the following:

- Create a PREDICTION JOIN on a time series model, and specify two months of new data.
- Request predictions for four time slices, where the starting point is 3, and the ending point is time slice 6.

In other words, if your new data contains time slices, and you request predictions for time steps 1 through , the predictions will coincide with the same period as the new data. To get new predictions for a time periods not covered by your data, you must either start predictions at the time slice after the new data series, or make sure that you request additional time slices.

Note

You cannot make historical predictions when you add new data.

The following example shows the DMX statement that lets you get only the new predictions for the two series in the previous example.

```
SELECT [Model Region],
PredictTimeSeries([Quantity],3,6, EXTEND_MODEL_CASES) AS PredictQty
FROM
   [Forecasting_MIXED]
NATURAL PREDICTION JOIN
   SELECT 1 AS [ReportingDate],
  '10' AS [Quantity],
  'M200 Europe' AS [ModelRegion]
UNION SELECT
  2 AS [ReportingDate],
  15 AS [Quantity]),
```

```
  'M200 Europe' AS [ModelRegion]
) AS t
WHERE [ModelRegion] = 'M200 Europe'
```

The prediction results start at time slice 3, which is after the 2 months of new data that you supplied.

Product and Region	Model with updated data (PredictTimeSeries with EXTEND_MODEL_CASES)
M200 Europe	M200 Europe, 9/25/2008 12:00:00 AM, 72 M200 Europe, 10/25/2008 12:00:00 AM, 69 M200 Europe, 11/25/2008 12:00:00 AM, 68 M200 Europe, 12/25/2008 12:00:00 AM, 89

Making Predictions with REPLACE_MODEL_CASES

Replacing the model cases is useful when you want to train a model on one set of cases and then apply that model to a different data series. A detailed walkthrough of this scenario is presented in Lesson 2: Building a Forecasting Scenario (Intermediate Data Mining Tutorial).

See Also

Querying a Time Series Model (Analysis Services - Data Mining)

PredictTimeSeries (DMX)