CHAPTER 10

Data Mining

As you have seen, Analysis Services enables you to build powerful Business Intelligence (BI) solutions that enable users to really understand the business. However, many business problems rely on the ability to spot patterns and trends across data sets that are far too large or complex for human analysts. Data mining can be used to explore your data and find these patterns, allowing you to begin to ask why things happen and to predict what will happen in the future.

In this chapter, we look at how to use some of the data mining features in Analysis Services 2005 to perform tasks such as customer segmentation and market basket analysis. The data mining results are presented in the form of new dimensions in cubes and are used in Web applications.

Business Problem

Our customer for this chapter is a large music retailer with stores across the country, and which also has an e-commerce site where customers can buy CDs. The retailer has also moved into the broader entertainment market and added product lines such as videos, computer games, and, more recently, DVDs. This latest product line has just been added to the Web site so that customers can buy DVDs online.

Problem Statement

The retailer faces strong competition in the online DVD market and is struggling to achieve profitability and gain market share. Its e-commerce system has built-in capabilities for conducting marketing campaigns and performing analysis; however, this is restricted to information learned from customers' online behavior and does not tie back into the retailer's

extensive data warehouse, which is populated mostly with information from their stores.

This has led to the following challenges:

- There is currently no way to segment customers by combining the extensive customer profile information with the Internet-usage metrics. This segmentation is needed so that they can target direct mail and other marketing to segments that will potentially use the Internet channel.
- The profit margin on DVD sales is low because of extensive competition. The retailer needs to find ways to increase the value of items sold in a transaction, such as by promoting and cross-selling additional products at the time of the purchase.

Solution Overview

We will build an extraction, transformation, and loading (ETL) process to add the Web site's visit-tracking data to the corporate data warehouse. We will use the data mining features of Analysis Services to help discover patterns in this data and provide the information back to the business.

Business Requirements

The high-level requirements to support the business objectives are as follows:

- **Customer segmentation.** The data warehouse already has excellent profiling information on customers that is obtained through a popular store loyalty card program. This information includes demographic profiles and detailed purchasing histories, because the customer's unique card number can be used to identify store transactions. However, the business also needs a profile of customers' online activities.

  The main areas of interest are *frequency,* or how often the customer uses the Web site, and *recency,* or how much time has elapsed since they visited the site. There is already information in the data warehouse on the third area of interest, which is *intensity,* or how much money the customer is spending through the Internet channel.

When these Internet profiling attributes are available, customers can be segmented into groups with relatively similar behavior. Analysts can use the information for marketing purposes, such as producing lists of customers for direct mail campaigns, as well as performing further analysis using the attributes and groups that we identified.

- **Online recommendations.** They would like to add an online recommendations feature to the new DVD area of the Web site to drive additional profit per online transaction. When a customer adds a DVD to her shopping basket, she must be prompted with a short list of other titles that she may be interested in.

  The performance of this recommendation needs to be good because any delay in the responsiveness of the Web site has been shown to lead to more abandoned transactions. Also, the recommendation must include items sold through the physical stores as well as the Web site, because the stores currently make up the bulk of the sales.

High-Level Architecture

We will add the Internet visit information to the existing data warehouse and Analysis Services cubes. Because the e-commerce application already extracts data from the Web logs and inserts it into a relational database, we will use this as the source for the ETL process. The data in this source database already has discrete user sessions identified.

Many e-commerce applications (including those based on the Microsoft Commerce Server platform) provide this kind of extraction and log processing functionality, but for custom Web sites, the only available tracking information may be the raw Internet Information Server (IIS) logs. A full treatment of the steps to extract this kind of information from Web log files is beyond the scope of this chapter; see the sidebar "Extracting Information from IIS Logs" for a high-level explanation.

After this information is in the data warehouse, we will use the data mining features of Analysis Services to achieve the business goals for segmentation and recommendations, as shown in Figure 10-1. For each area, we will create a data mining structure that describes the underlying business problem and then run the appropriate data mining algorithm against the data to build a mathematical model. This model can then be used both for predictions such as recommending a list of products or for grouping information in cubes together in new ways to enable more complex analyses.

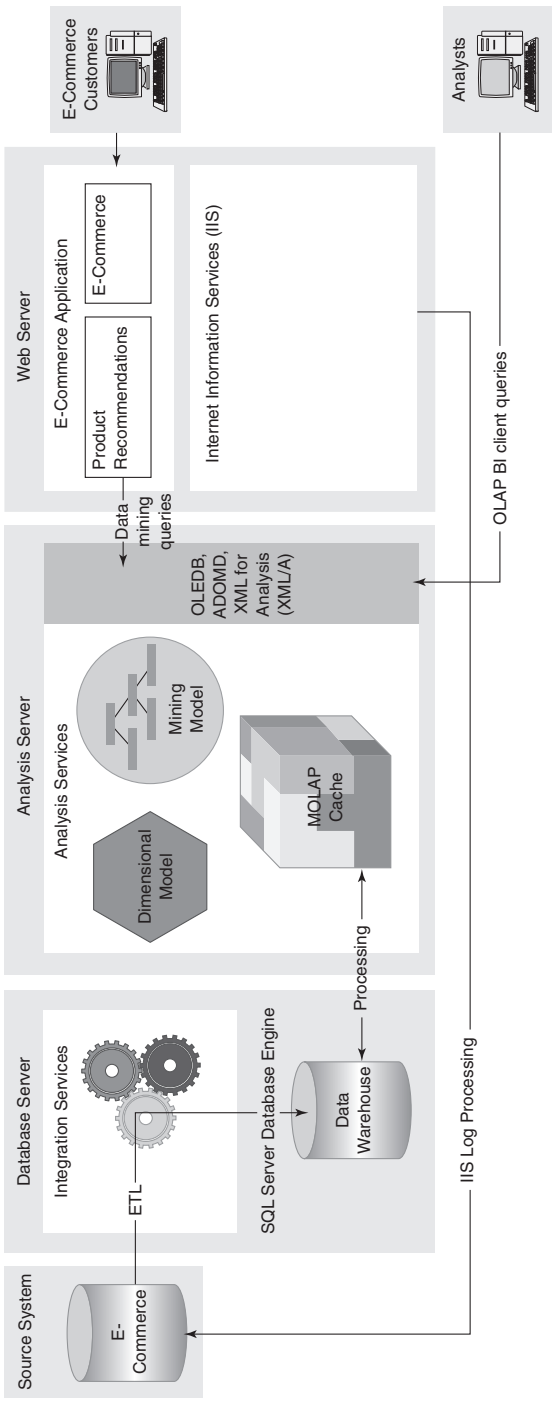

Figure 10-1 High-level architecture

Data mining in Analysis Services has several different types of algorithms to perform tasks such as classification, regression, and segmentation. We will use the **Microsoft Clustering** algorithm to create a customer segmentation mining model, and then the model will provide these categories of customers' online behavior as a new dimension for cube analysis. We will use the **Microsoft Association** algorithm to build a data mining model that can be used to make product recommendations, and then add code to the Web site to query this model to suggest appropriate DVDs for online shoppers.

Alternative Solution: Build a Feed from the Data Warehouse to the E-Commerce System

Because the e-commerce application already has some built-in BI capabilities, we could use these features for customer segmentation and product recommendations if we built a feed from the corporate data warehouse to supply extra information, such as detailed customer profile information or even sales totals for other channels.

However, this approach is not recommended in this case because it will be impractical to meet the business requirements. Product recommendations need to be based on sales through the physical store channel as well as online transactions, and copying all the sales transactions to the e-commerce data warehouse database is not viable. Also, customer segmentation is a core activity for the marketing department, and they need to have access to all the rich information in the data warehouse.

In summary, although many e-commerce applications have built-in analytical functionality, most large retailers that also have physical stores will already have an investment in a data warehouse, and the most appropriate approach will often be to find ways to extend this with information from the Internet channel.

Extracting Information from IIS Logs

Although in our example solution, we will be taking advantage of the log parsing facilities that are built in to the e-commerce application, many companies have built a custom Web site where the only available tracking information is the raw IIS logs.

The first step in extracting this information is to parse the log files and extract the information into a staging database. You could create an Integration

Services package to perform this extraction, possibly with an additional tool to make the data easier to work with. Microsoft has a free Log Parser utility (www.logparser.com), and third-party parsers are also available.

However, after you have extracted the raw information, the real fun begins, and is not for the faint of heart. Finding discrete "sessions" involves looking for an identifier in the logs such as username or cookie and then identifying a time period that could identify a "visit" fact.

If you also want to look at what pages the users visited, you need to parse URLs to deal with pages that are parameterized with identifiers such as product IDs (for example, product.aspx?ID=322442). On the whole, it is generally much easier to take advantage of an e-commerce application's existing parsing facilities if they exist, or otherwise find an existing tool that meets your needs.

Business Benefits

The solution will deliver the following benefits:

- Targeting direct mail and other marketing to identified groups of customers that will probably use the Internet channel will decrease the overall cost of marketing and increase the company's market share.
- Profitability will be improved by increasing the average number of items sold per transaction, such as selling more DVDs in one transaction and still incurring one shipping cost.

Data Model

Most of the data model for e-commerce is similar to the standard retail data model. The data that we need for product recommendations is simply a Sales fact that shows products that are sold over time. The interesting new areas in e-commerce are the facts that allow us to understand the behavior of visitors to the Web site.

Many BI solutions for Internet applications focus on how the site is used. Information such as the order that people visit pages, which page they start at, what Web site they came from—all of this can help companies to improve the effectiveness of their Web sites. Tracking

information such as "click-through" rates, which measure how often users click an advertisement, can produce more optimized campaigns and a better experience for users.

However, for this solution, we focus on the other side of the Web equation: Who are the people visiting our site? To really be able to understand customers well enough to produce rich customer profiles, we need to keep track of the visits that users make to our site.

How Often Are Users Visiting the Web Site?

The new fact table in the data warehouse is Visit, which has one record for each completed customer visit to the site, as shown in Figure 10-2. So, if a customer signed on at 3:10 p.m. and then clicked through several pages with the last page hit logged at 3:25 p.m., the e-commerce application's log parser will see the last page hit and create a single record that spans the whole time period.

The measures that we will be tracking are the duration of the visit and the number of requests (or page hits) during the visit. Because we are interested in the date that the visit took place as well as the time, we will use the approach explained in earlier chapters and have separate Date and Time of Day dimensions. We can also include the Referrer Domain that the user came from which helps us to determine which Web sites are sending the most traffic to our site, and the type of browser platform that the customer was using, including browser version and operating system. This dimension is often called User Agent rather than Browser Platform because other software such as search spiders can also visit the site; however, we always use business-friendly names in the data warehouse rather than terms such as User Agent, which probably only makes sense to Web geeks.

If your log parser supports it, one useful dimension that we can add is Visit Result, which has values such as Browsed, Abandoned Transaction, and Completed Transaction. This is somewhat difficult for parsers to derive from the Web logs, however, because they would need to look for specific marker pages in the log, such as a confirmation page when the user completes a transaction.

The e-commerce application's database also includes another table with the actual page hits that took place, so in some ways it seems we are breaking one of the cardinal rules of dimensional modeling—always use the most detailed grain available. By using the summarized Visit table, we are losing the ability to analyze by a Page dimension, which shows

which pages the user hit. Although powerful, the Page Hits fact table will inevitably be huge, and we would need a good business case to go to the trouble of managing this volume of data. Also, the kinds of analysis that Page Hits would enable are often already provided directly by e-commerce applications, and in this case don't need to be augmented with the extra information stored in the data warehouse.

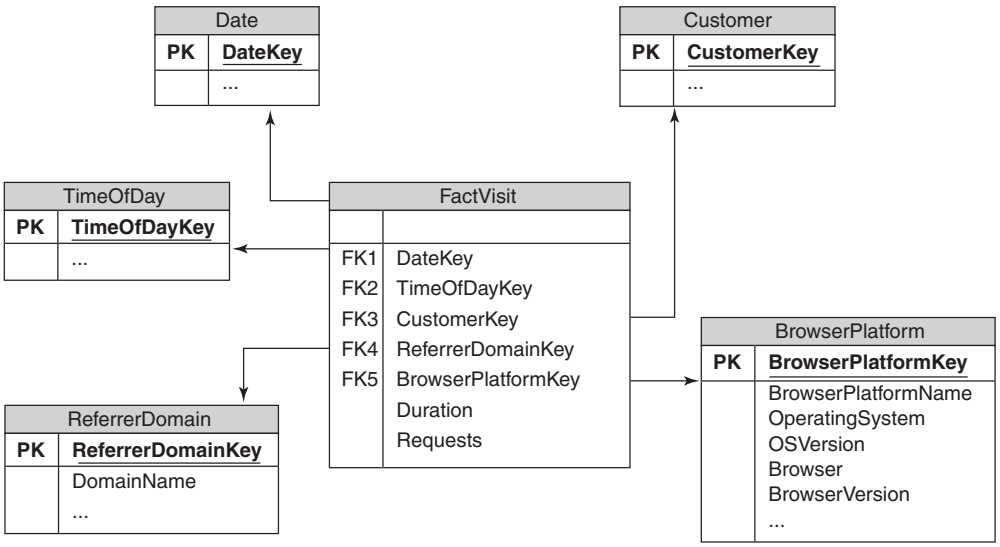

Figure 10-2 Visit fact

One question that arises is whether we can tie the Visit fact back to the Sales Transaction fact. If we could do that, maybe we could show information such as how profitable visits were. It turns out that for SQL Server 2005, it doesn't really matter if you have the information on the same fact record. Because a single cube can contain measure groups for both Sales and Visits, if a user has selected a date range and a customer or grouping of customers, measures such as total visits and total revenue display properly anyway.

In other words, the common dimensions between the two fact tables provide the means to tie the information together, and we don't actually need to link the two fact records. The site activity is tied to the sales transactions by way of the fact that they occurred in the same time interval to the same customer.

Who Is Using the Web Site?

The most important dimension is certainly Customer, but data warehouse architects face an interesting challenge when it comes to the Web—we often don't know who the customers are. Most e-commerce sites require users to create an online profile, and if the customer must sign in before placing any orders, we can usually associate the Web site activity after the user signs in with the customer's profile. However, online profiles usually contain little mandatory information (and as many online retailers will testify, the information they contain is often meaningless).

The goal for effective BI from the Web site visit information is to augment the minimal Internet profile information with rich, accurate demographic information. In our solution, the site profile includes an optional loyalty card number that is issued by the physical stores. Because customers build up credits to earn free CDs, this information is well populated and can be used to connect the online behavior from the Web site's customer profile with the data warehouse Customer dimension. (In case you have ever wondered why stores are so keen to hand out loyalty cards, now you know—they are trying to build a good Customer dimension!)

For customers who don't have a loyalty card number and an existing Customer record in the data warehouse, we have two choices: Either we can create new Customer records for each of the unmatched customer profiles with minimal information or we can use a single "Unknown Internet Customer" record that groups all these customer together. Because even the minimal online profile will allow us to track information such as how long they have been using our Web site, we will still be able to achieve some of our segmentation goals such as identifying frequent visitors, and so we will go ahead and create new Customer records for every distinct online profile that doesn't have a loyalty card.

Note that the CustomerKey will be blank for all visits where the user did not sign on but just browsed the site. If the user actually places an order, he must sign on and therefore there will be a customer key, but we will allocate all the other facts to an "Unknown Customer" record. It is important not to just discard these fact records, because even without the customer information, the Visit fact table is a valuable source of information about peak traffic levels on the site.

Alternatively, we could have solved the problem by modeling a separate "Internet Customer" dimension that is only used for this area and

not related to other facts such as Sales, and thus avoid creating extra records in our Customer dimension. However, this would mean that we couldn't create a cube with a common Customer dimension that combines measure groups for Internet visit measures with sales and other facts for the business, which is really the central goal of this solution.

What Interesting Attributes Can We Track?

The first attribute we can add to the data warehouse Customer dimension is one of the easiest but most interesting: an InternetUser flag that indicates whether the customer has created a profile on the site, as shown in Figure 10-3. This is easy to populate and will enable analysts to start to understand the characteristics of people who use the Internet channel versus those who don't.

A related InternetPurchaser attribute can be derived by looking at the Sales transactions and flagging every customer who has made any purchases using the Internet channel. All InternetPurchasers will, of course, be InternetUsers, but the reverse is not true because some users will have created a profile but not yet made a purchase online. Although analysts could easily and flexibly get a list of customers who had purchased through the Internet by browsing the Sales cube and selecting the Internet channel and a time period, it is still a good idea to add the InternetPurchaser flag so that it is easy for both analysts and data mining models to distinguish those customers who have ever used the Internet channel from those who haven't.

Customer	
PK	**CustomerKey**
	StoreLoyaltyCardNo ... InternetUser InternetPurchaser DateFirstInternetVisit DateLatestInternetVisit DateFirstInternetPurchase DateLatestInternetPurchase

Figure 10-3 Customer dimension

Other interesting customer attributes are DateFirstInternetVisit, which tells us how long they have been using our site, and DateLatestInternetVisit, which tells us how recently they have visited. Both of these

attributes are derived from the underlying Visit fact table, but will be added to the Customer table to make the dimension easy to query. Note that this means we will be updating our customer records much more often, so one way of simplifying the ETL process would be to create a view over Customer and the Visit fact table that returns the maximum date for each customer and is used as the source for the Analysis Services Customer dimension. We can also add equivalent date columns for the date of the first actual online purchase, and the most recent online purchase.

Technical Solution

We start this section by reviewing the changes that were made to add the Visit fact and customer information to the existing data warehouse, and then give a detailed description of the data mining sections of the solution.

Adding Visit Information to the Data Warehouse

To add the Visit fact table and associated dimensions to the database, we need to supplement the existing ETL procedures to load data from the e-commerce application's tables. As always, when adding a new data source to the warehouse, you need to perform data mapping to match any existing dimensions. We have already discussed that for the Customer dimension we will be using the store loyalty card number to map Internet profiles to customer records. Date and Time of Day keys are usually simple to map; however, because time stamps in Web server logs are either in the local time of the server or in UTC (coordinated universal time), we need to check this before implementing the ETL.

The Referrer Domain dimension will be sourced from the equivalent table in the e-commerce database, but if you are implementing ETL to extract this information from log files (see the sidebar "Extracting Information from IIS Logs"), you need to parse the URL of the referring page to extract the domain name. The Browser Platform attributes such as OperatingSystem and BrowserVersion also need to be extracted from the User Agent field in the log files.

Customer Dimension Changes

The new customer attributes can easily be added to the cube definition by refreshing the data source view (DSV) in BI Development Studio to pick up the new columns, and then adding these as attributes to the Customer dimension. However, they may not be in the best format for analysis purposes—having the specific date that a user first visited the site is not very illuminating for users of the cube. In fact, they would probably be better served by being able to select customers based on groups that show how long they have been Internet site users (for example, "3–6 months").

We can add this information into the DSV as a named calculation on Customer or add it to the underlying view in the database. You can implement the MonthsSinceFirstInternetVisit named calculation by using the `DateDiff` function to work out the number of months between the date of the first visit and the current system date:

```
DateDiff(m, DateFirstInternetVisit, GetDate())
```

Instead of showing the user a long list of numbers, it would be better to group the numbers together into ranges, such as 1–3 months, 3–6 months, and so on. Although we could do this manually using a lookup table of month ranges, we can take advantage of the Analysis Services **discretization** feature to do this for us. After adding the MonthsSinceFirstInternetVisit attribute to the Customer dimension, change the DiscretizationMethod property of the attribute to Automatic to allow Analysis Services to decide on the best method of grouping these time periods. If you want to specify the approximate number of groups (or "buckets") that are created, set the DiscretizationBucketCount property, too.

TIP:

Use the Data Load Date Rather Than System Dates

Although using the `GetDate` function to work out durations based on the current system date would work, bear in mind that because we will only be loading the data on a weekly basis, the `GetDate` function should probably be changed to return the latest date that the data was actually loaded. This date could be stored in a data warehouse table that is populated during the ETL process.

Visit Measure Group

Because we want to be able to look at both sales and visit information together, we can add the Visit fact table to the existing cube as a new measure group. Dimensions that are not used in the Visit measure group (such as Product and Promotion) will be grayed out in the Dimension Usage table of the cube editor.

One measure to be careful of is the Duration measure. Although this measure is additive across time (for example, we could determine the total duration that a group of customers spent on the site in the month of January), using the information by summing up the facts in this way does not make a lot of business sense. The Duration measure is there to provide an indication of how long people spent on the site; and so, we can change the AggregateFunction property of this measure to AverageOfChildren to display this information in the way that users will expect.

How We Will Be Using Data Mining

As discussed in the section "High-Level Architecture," we chose the Microsoft Clustering and Microsoft Association algorithms for our solution. Knowing which algorithm is appropriate for your business problem will take some experimentation and research in the documentation. In fact, in a lot of cases, there is no obvious candidate at the outset, and you will need to try different algorithms against the same underlying data to see which is most appropriate. The data mining designer also includes a Mining Accuracy Chart tab that you can use to compare algorithms.

The first decision we need to make is where the data will come from. Analysis Services can use either the relational tables in your data source view or the actual cube itself as the source of data for the models. Because data mining is even more sensitive to flawed data than most applications, it is important to ensure that you perform as much data cleansing as possible against the source data prior to processing your models; so, at the very least, you should probably be using the tables in your data warehouse rather than directly using source systems.

However, using the cube as the source for data mining has a number of benefits, so we will be using that approach for this solution. The cube data has already been supplemented with additional attributes and calculated measures that the data mining algorithms can take advantage of. Also, the load process for data mining models can take some time, so using the cube as the source means that the aggregates will be used if applicable, potentially speeding up the processing time.

Approaching the Customer-Segmentation Problem

Because users can slice and dice information by all the attributes in a dimension rather than just predefined drilldown hierarchies, analysts could use the new Internet-related attributes that we added to drill down through the data and start to understand how customers' online activities affect measures such as total sales or profitability. For example, they can learn that frequent visitors to the site often have high sales amounts, but that this isn't always the case—some frequent visitors are "just looking."

To really do a good job of targeting the DVD marketing campaign to customers likely to act on the information, analysts need to perform a segmentation exercise where all customers that have similar attributes are categorized into groups. Because the list of customers is huge and there is a large number of attributes, we can start this categorization process by using a data mining algorithm to search through the customers and group them into clusters.

The Microsoft Clustering algorithm is a great tool for segmentation and works by looking for relationships in the data and generating a list of clusters, as shown in Figure 10-4, and then gradually moving clusters around until they are a good representation of the data.

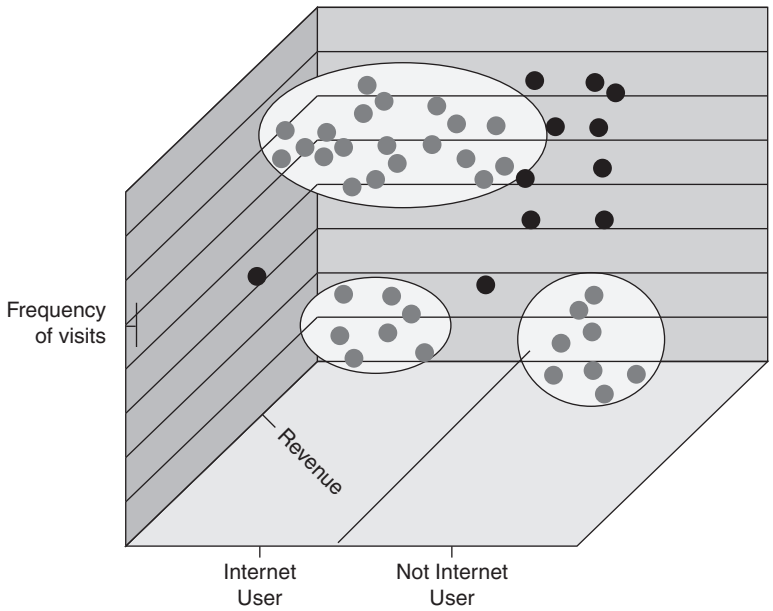

Figure 10-4 Clusters of data

Getting Started with Data Mining

We start the data mining process by creating a new mining model for the customer segmentation exercise, using an existing Analysis Services project that contains the cubes and dimensions with both the data warehouse information (such as in-store sales) and the new Internet information described earlier.

In Analysis Services data mining, we define a mining structure that describes the underlying data that will be used for data mining. Each mining structure can contain multiple mining models, such as a clustering model and an association model, that all use the same underlying data but in different ways.

QUICK START: Creating a Cluster Data Mining Model

We will be using the e-commerce sample project for this example because it already contains the necessary source cubes and dimensions:

1. Open the Business Intelligence Development Studio.
2. Select Open Project from the File menu and open the e-commerce sample project.
3. On the Project menu, select New Mining Structure.
4. After the first wizard page, select From Existing Cube for the definition method and click Next.
5. Select the Microsoft Clustering data mining technique and click Next.
6. Select the Customer dimension from the e-commerce cube and click Next.
7. For the case key, leave the default selection of the Customer attribute and click Next.
8. Select all the Internet activity-related attributes of the customer, such as Internet User and Internet Purchaser, as well as the Months Since attributes that we described in the first section. Also, select the Visits Count and the Sales Amount from the cube, and then click Next.
9. On the Mining Model Column Usage page, leave the default selections—all of our selected columns will be used as inputs. Click Next.

10. Leave the default settings for the column content and data types and click Next.

11. We will be using all the information in the cube for this mining model, so click Next on the Slice Source Cube page.

12. Specify Internet Models as the mining structure name, and Customer Internet Segmentation as the mining model name, and then click Finish (see Figure 10-5).

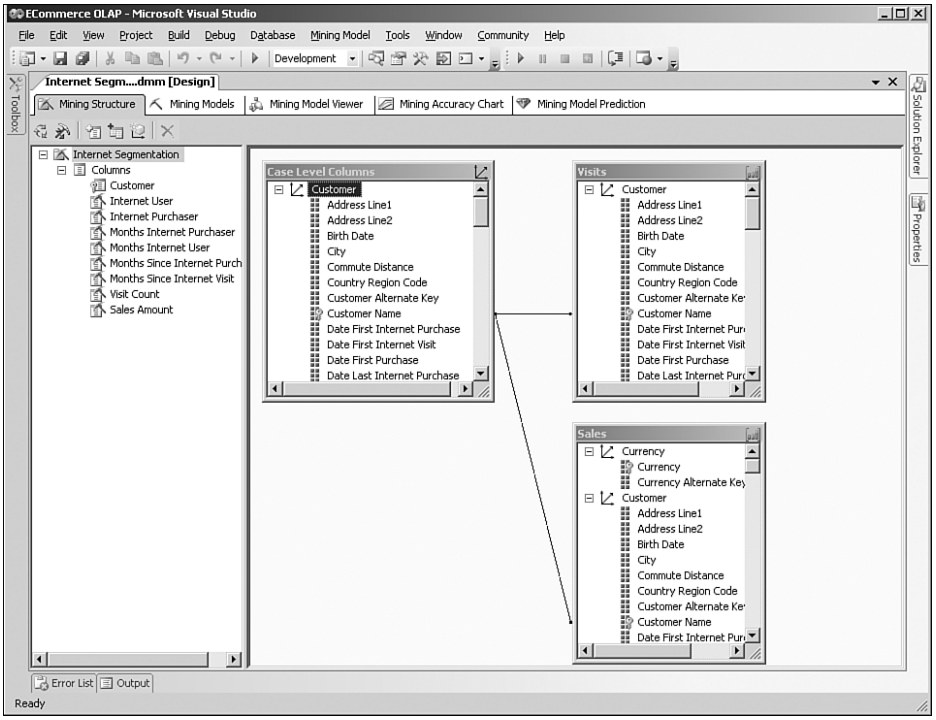

Figure 10-5 Creating a cluster data mining model

The wizard will create the mining structure and model and open the structure in the data mining designer. The underlying data that you selected is shown on the Mining Structure tab, and the Customer Internet Segmentation model is the only model in the list on the Mining Models tab.

Before working with the model, you need to deploy the solution and process the mining model. During processing, Analysis Services applies

the algorithm you selected (Microsoft Clustering) to the data from the cube to allocate all the customers to their appropriate clusters—your next task in data mining is to understand the information that has been produced and relate it to the real world.

Looking at the Clusters Created

The Mining Model Viewer tab in the model designer enables you to view the model that has been processed. Each algorithm produces a different type of model, so there are specific viewers for each model. The initial view for clusters is the Cluster Diagram, which shows all the clusters in the model with lines connecting them. Each cluster is positioned closer to other similar clusters, and the darkness of the line connecting two clusters shows the level of similarity. The shading of each cluster by default is related to the population of the cluster (that is, how many customers it contains—the darker clusters have the most customers).

For our Customer Internet Segmentation model, we can see ten clusters named Cluster 1 through Cluster 10. Each cluster represents a group of customers with similar attributes, such as customers who are fairly new to our Internet site and have not made a lot of purchases yet. Our task at this stage is to understand the kinds of customers in each cluster and hopefully come up with some more meaningful names for the clusters.

We can start by using the Cluster Diagram's shading variable and state parameters to look at each attribute and see which clusters contain the most customers with the selected attribute. For example, if I select Sales Amount > 1275 in Figure 10-6, I can see that Cluster 5 and Cluster 8 contain the most customers who have total sales of more than $1,275, as shown in Figure 10-6.

You can use the cluster diagram to help you comprehend each cluster by looking at one variable at a time. To really understand and compare the composition of clusters (that is, what types of customers are in each group), you need to use the Cluster Profiles and Cluster Discrimination views. We can see in the diagram that Cluster 1 contains a fairly high percentage of customers with high sales and is arranged near to Cluster 2 and Cluster 6, but we need more complete information to be able to assign a meaningful name to these clusters.

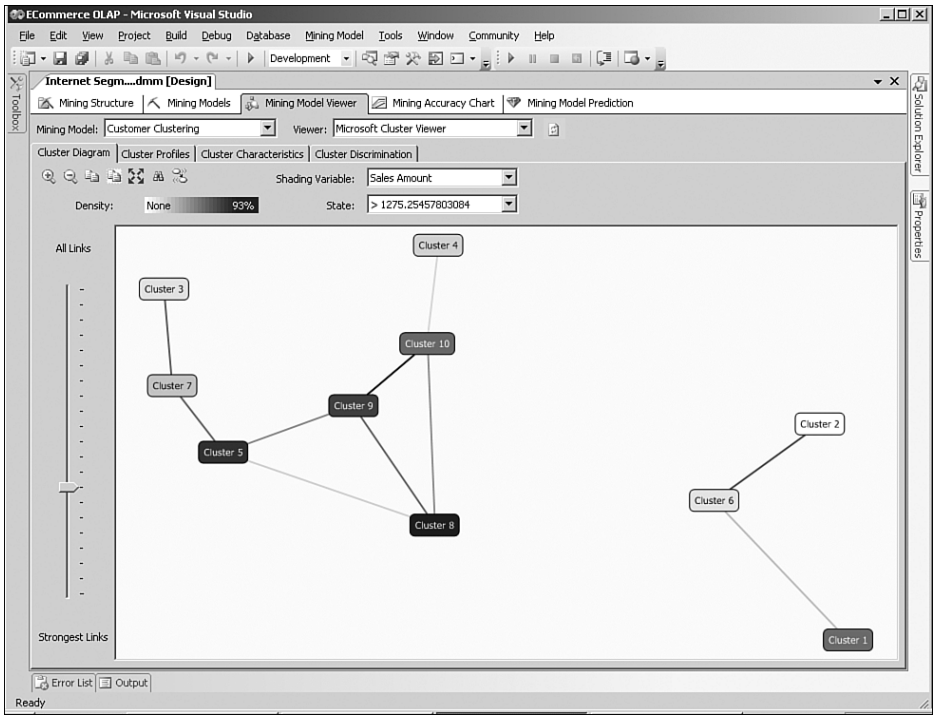

Figure 10-6 Cluster diagram

Understanding the Composition of Clusters

The Cluster Profiles view shows all the clusters that were identified as columns, and each attribute that you selected for your model as the rows, as shown in Figure 10-7. Looking first at Cluster 1, we can see that all the customers in the group have an Internet Purchaser attribute of False, as well as an Internet Visitor of False. So, the mining algorithm has grouped customers together who have never visited the site or purchased anything online—all their purchases have been at a physical store. Note that we can come to this rather useful conclusion only because we understand the underlying business, which is a key point about data mining.

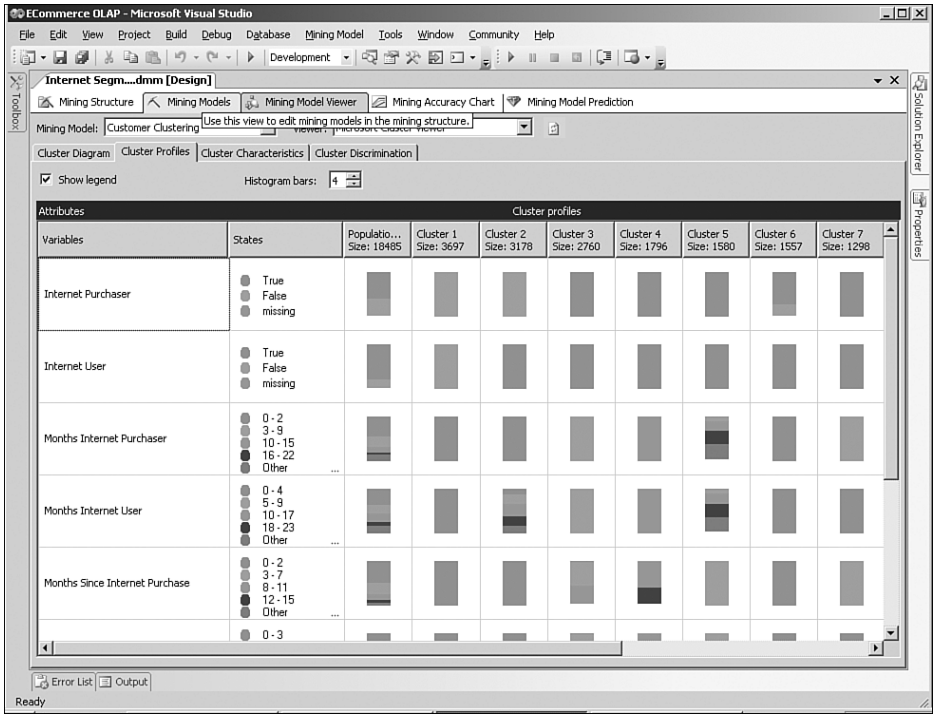

Figure 10-7 Cluster profiles

To give Cluster 1 the more sensible name of Store-Only Buyers, right-click the cluster name and select Rename Cluster. So, we now have a single cluster identified; what about the others? If you look at the next column, you can see that Cluster 2 differs from Store-Only Buyers in that all the customers in the cluster have actually visited the site, but they just haven't made any purchases online yet. We can call this cluster Browsers because they are customers who are (so far) using the site for information gathering only.

Cluster 6 contains visitors who have also made a purchase, but if we look closely at the Months Internet Purchaser and Months Internet User attributes, we learn that they are all relative newcomers to our site—all of them have been visitors and purchasers for between zero and three months (they are "Newbies"). We can continue the process of looking at each cluster, but the rest of the clusters are not quite so clear-cut, so we need a better tool for differentiating between them.

Discriminating Between Similar Clusters

If you look at the profiles of Clusters 8 and 9 in BI Development Studio, you will notice that they both have multiple values for the number of months that customers have been Internet visitors and Internet purchasers. This illustrates an important point about the clusters that the algorithm identifies: Every customer in the group does not have to have exactly the same value for every attribute. This is somewhat confusing when you start working with clusters; for example, you might have named a cluster Urban Professionals and then discover that it also contains a customer who lives in the countryside.

The reason for this is that the customer, when you look at all of his or her attributes together, is most similar to the customers who live in urban areas and have professional occupations. So naming a cluster Urban Professionals does not necessarily imply that it contains absolutely no manual laborers who live in the suburbs, but rather gives a high-level shorthand for the predominant combination of attributes in that cluster.

Because the clusters identified are therefore sometimes ambiguous, we need a way of discriminating between similar clusters to find out what exactly makes them different. We can use the Cluster Discrimination view, as shown in Figure 10-8, to select the two clusters we are interested in comparing and get an idea of what the most important differences are.

We can see in the discrimination view that although the cluster profiles of 8 and 9 look similar, in fact Cluster 9 contains mostly customers who have made a visit and purchase in the past few months, are fairly frequent visitors, and have spent a lot of money with us—we could call this group Frequent Visitors. Cluster 8, on the other hand, contains mostly customers who have not visited the site for many months, although in the past they have spent some money with us. This cluster is probably one that we want to pay careful attention to, because they may now be doing their shopping with a competitor. That is, they may be Defectors.

With the cluster profile and discrimination views, we can understand the clusters well enough to give them meaningful names, so we can now turn our attention to providing this information back to users to enable them to perform analyses on the data using the clusters.

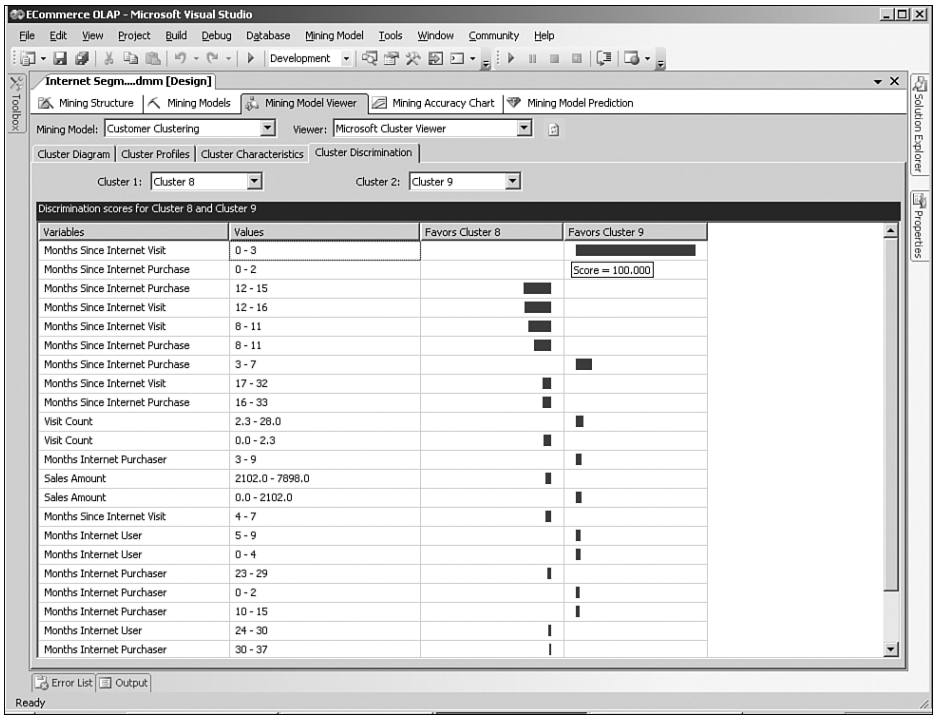

Figure 10-8 Cluster discrimination

Analyzing with Data Mining Information

Analysis Services allows you to create a special type of dimension called a **Data Mining dimension**, which is based on a data mining model and can be included in a cube just like an ordinary dimension. The Data Mining dimension includes all the clusters that were identified by the algorithm, including any specific names that you assigned to them.

Adding a Data Mining Dimension to a Cube

We will use the data mining model we created in the previous Quick Start exercise and create a new dimension called Customer Internet Segmentation, as well as a new cube that includes this dimension. The existing Visit and Sales measure groups from the e-commerce cube will be linked into the new cube to be analyzed by the new dimension.

To create the dimension, open the data mining structure and go to the Mining Structure tab. Select Create a Data Mining Dimension on the Mining Model menu. Specify the dimension and cube names and click OK. Before you can use the new objects, you must deploy the solution and process the new dimension and cube.

Using the Data Mining Dimension

Because the dimension has been added to the cube, marketing database analysts can use the new segmentation to understand measures such as profitability or total sales for each of the clusters and refine the set of customers who will be targeted by the direct mail campaign to publicize the site's new DVD products. The list of customers can be provided either from a drillthrough action in a BI client tool or by building a Reporting Services customer list report that allows the user to select the cluster and other attributes.

Creating a Model for Product Recommendations

Our approach for product recommendations is based on the idea that we can use a mining model to look at every customer and the DVDs that they have bought, and then look for patterns of DVDs that often occur together. The Association Rules mining model is often used for this kind of analysis (sometimes called market basket analysis) and produces a set of rules that say, for example, if the customer is buying a DVD of *The Godfather*, what are the other movies that other buyers of *The Godfather* have purchased?

Each of these rules has a probability associated with them. For example, many customers may also have bought similar films, such as *The Godfather Part II* or *Goodfellas*, so the rules that relate *The Godfather* to these DVDs would have a high probability. If only a single customer bought *It's a Wonderful Life* and *The Godfather*, this rule would have a low probability. In data mining terminology, the number of times that a set of items occurs together is called the **support**, so the example of *It's a Wonderful Life* and *The Godfather* appearing together would have a support of 1.

We can use these rules to make a prediction: For a specific DVD, give me a list of the most probable DVDs that a customer might also enjoy.

Asking the Right Question

The best way to successfully set up a sensible data mining model is to be precise about the question you ask. Because we are looking for DVDs that sell well together, is the question we are asking "Which other DVDs have been bought during the same shopping trip?" or rather "Which other DVDs did customers also buy at some point?." If you were doing product recommendations on groceries, the first question would probably be the most sensible. The reason is that if someone is buying beer and wine today, we can probably recommend ice and potato chips because those are often sold in the same transaction.

However, in our case, we are trying to determine the likes and dislikes of consumers, which have a longer duration than just a single transaction. We are really trying to understand what kind of movies customers enjoy, so the second question is more appropriate for this business solution. To set up the model, we need to look at each customer and determine the list of DVDs that they have purchased. The data we are looking for looks something like Table 10-1. In data mining terminology, the customer would be the **case**, and the list of products for each customer would be a **nested table**.

Table 10-1 Customer DVD Purchase History

Customer	DVD
Customer 3283	*The Godfather*
	The Godfather Part II
	Dark City
Customer 3981	*The Godfather Part II*
	Goodfellas
Customer 5488	*The Godfather*
	It's a Wonderful Life
…	…

QUICK START: Creating an Association Rules Data Mining Model

We can add the new product recommendations mining model to the same e-commerce solution as the segmentation model; but because we are using different underlying data, we need to create a new mining structure, too:

1. On the Project menu, select New Mining Structure.
2. After the first wizard page, select From Existing Cube for the definition method and click Next.
3. Select the Microsoft Association Rules data mining technique and click Next.
4. Select the Customer dimension from the e-commerce cube and click Next.
5. For the case key, leave the default selection of the Customer attribute and click Next.
6. Leave the other case level columns blank and click Next.
7. Click the Add Nested Tables button, select the Product dimension from the e-commerce cube, and click Next.
8. For the case key, leave the default selection of the Product attribute and click Next.
9. Leave the other case level columns blank and click Finish.
10. Check the Predict box next to the Product dimension. The wizard's column usage page should now look like Figure 10-9. Click Next.
11. Leave the default settings for the column data types and click Next.
12. On the Slice Source Cube page, select the Product Category hierarchy for the Product dimension and specify a filter expression of DVD.
13. Specify Cross Sell as the mining structure name and Product Recommendations as the mining model name and click Finish.

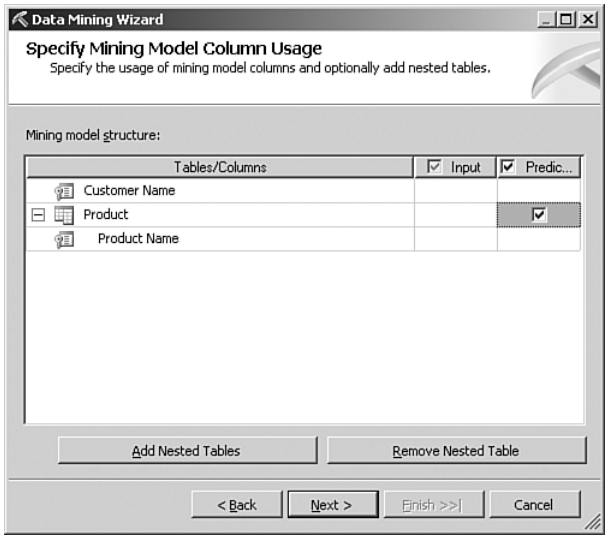

Figure 10-9 Product recommendations column usage

Understanding the Product Recommendation Rules

Once again, you need to deploy and process the model before you can view the results. The algorithm produces a list of products related to each other for a customer, and you can view these in the Itemsets tab of the mining model viewer. Each itemset also shows the support or number of times that the set occurred together. If you click the Rules tab, you can see the main feature of this mining model (see Figure 10-10): A set of rules that can be used to calculate the probability that a new DVD will be appropriate based on the existing DVDs in the customer's shopping basket.

For the Association Rules algorithm, the settings that you choose have a big impact on the set of rules created. You can change these settings by right-clicking the model in the Mining Models tab and selecting Set Algorithm Parameters. If you end up with long processing times and too many rules, you could increase the minimum probability parameter, which would discard rules with a low probability, or you could increase the minimum support, which would discard rules that do not occur very often in the data. If, on the other hand, you end up with too few rules, decrease the minimum probability and support.

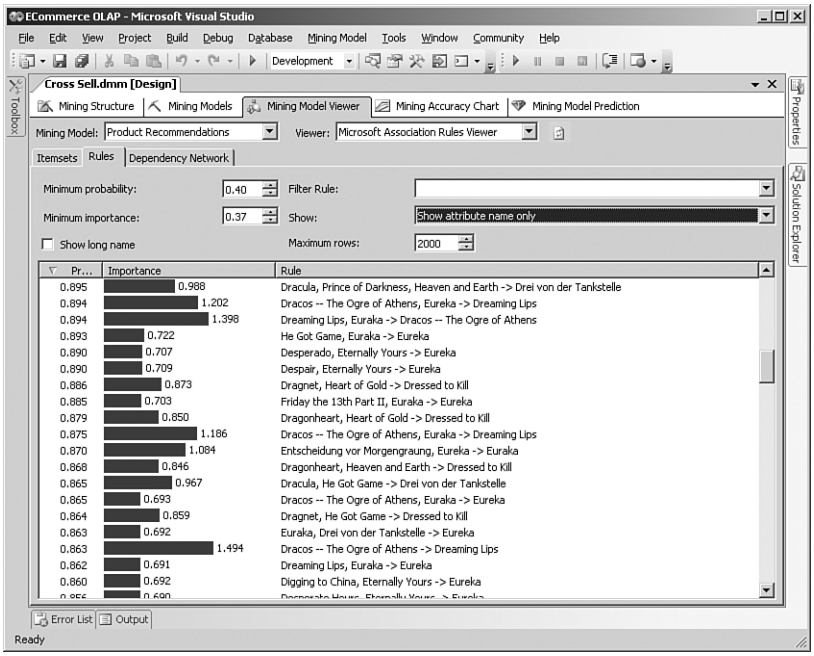

Figure 10-10 Product recommendation rules

When we have finished creating and training the mining model, we can move on to using the mining model to make predictions for our Web application.

Add Data Mining Intelligence into a Web Application

Web applications that include dynamic content typically access relational databases to provide information to users, usually by using a data access library such as ADO.NET to execute an SQL query against the database, then looping through the resulting rows to create a Web page. The process for adding data mining information to a Web application is similar. A programming library called ADOMD.NET provides classes for querying Analysis Services, and the Data Mining eXtensions (DMX) query language is used to request information from the mining model.

Querying the Mining Model Using DMX

The DMX language is similar to standard SQL, but there are enough differences that you will need to spend some time working with the

language before it becomes natural. As with all new query languages, it is often better to start out using graphical designers, such as the prediction query builder in SQL Server Management Studio, before moving on to hand-coding queries. There is also a thriving user community for Analysis Services data mining, and there are lots of samples available to get you started at www.sqlserverdatamining.com.

QUICK START: Using the Prediction Query Builder

We need to build a DMX query that recommends a likely set of products based on a product that the customer purchases. The prediction query builder enables you to create a query based on either a list of values that you supply (called a singleton query) or to make batch predictions on a whole set of records contained in an input table:

1. In SQL Server Management Studio, open the Analysis Services database in the Object Explorer.
2. Browse to the Product Recommendations mining model, right-click it, and choose Build Prediction Query.
3. On the Mining Model menu, select Singleton Query.
4. On the Singleton Query Input box, click on the Product row in the Value column and click the … button.
5. Add some products from the list to the input rows and click OK. We have now defined the input to the query. The next step is to define the output (that is, what we want to predict).
6. In the first row of the grid in the lower half of the designer, select Prediction Function in the source column, and select Predict in the Field column.
7. Now that we have selected the function we want to use, we need to supply the arguments to the function. In the Criteria/Argument column, specify `[Product], INCLUDE_STATISTICS`, as shown in Figure 10-11.
8. On the Mining Model menu, select Result to see the recommended product list.
9. To see the actual DMX that was produced by the designer, select Query from the Mining Model menu.

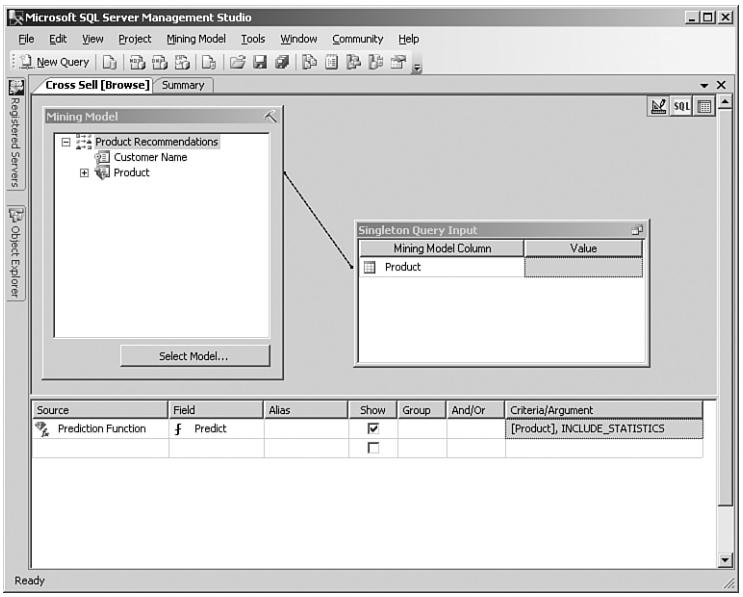

Figure 10-11 Prediction query designer

The prediction DMX query that we have built returns a long list of products, some of them with fairly low probabilities. What we actually need for the Web site is the top five or so best recommendations, so we can add a numeric parameter to the query to specify the maximum number of results to return. Our final DMX query looks like the following:

```
SELECT Predict ([Product], INCLUDE_STATISTICS, 5)
FROM [Product Recommendations]
NATURAL PREDICTION JOIN
  (SELECT
     (SELECT 'The Godfather' AS [Product Name]
      UNION SELECT 'Battlestar Galactica Season I' AS
➥[Product Name])
  AS [Product]) AS a
```

Executing DMX Queries from a Web Application

Our Web application is an ASP.NET application, so we can take advantage of the ADOMD.NET classes to execute the DMX query. The DMX

query that we designed earlier returns the information in a hierarchical format, and we would need to navigate through it to extract the product names. We can simplify the code by using the `SELECT FLATTENED` syntax, which returns the information as a simple list of products that we can add to the Web page.

Managing the Solution

Data mining features are provided as part of the Analysis Services engine, so many of the usual management tasks are still applicable, but data mining also adds some new areas to be aware of.

Deployment

We are adding the visit information and customer-segmentation mining models to an existing BI solution, so the main deployment challenge that we face is for the product recommendations mining model. Because this model will be used on the live Internet site, we need to deploy an additional Analysis Server that can be accessed by the Web application servers.

Deploying the Mining Model to the Production Server

Because we built our mining model using an Analysis Services cube as the source, we need to find a way to get the processed cube and mining models onto the production server. The easiest way to do this is to back up the Analysis Services database and restore it in production.

If we had built our mining model using a relational database as the source rather than an Analysis Services cube, we could easily have deployed the mining model using DMX's `EXPORT` command, by running a DMX query in SQL Server Management Studio with the following syntax:

```
EXPORT MINING MODEL [Product Recommendations] TO
'C:\ProductRec.abf' WITH PASSWORD='MyPassword'
```

Remember that the path you specify is on the server, not on the client machine that you are running the DMX query from. The file will

contain everything we need to make predictions, so we can copy it to the production server, create a new Analysis Services database, and import the model using a DMX query with the following syntax:

```
IMPORT FROM 'C:\ProductRec.abf' WITH PASSWORD='MyPassword'
```

Securing the Mining Model

Because we will be accessing the product recommendations mining model from a Web application, we need to set up the security correctly to allow the IIS user account to query the Analysis Server. We need to create a special role in the production Analysis Services database that has the IIS account (such as IUSR_machinename) as a member, and then enable Read and Read Definition permissions for the mining structure and model.

Maintenance

Any future changes that you make to the mining model definitions will need to be redeployed to the server and will also require processing the changed models. Mining structures can include more than one model on the same underlying data (for example, using different algorithms or even just different algorithm settings), so you might need to add new models to handle future business requirements.

Operations

Just like Analysis Services cubes, we need to reprocess mining models to train them with new input data. There is no "incremental process" option with mining models, however, so you need to reprocess using the full data set.

The product recommendations model probably needs to be reprocessed on a regular basis so that it is kept current with the latest products and sales trends. In our e-commerce solution, the model is reprocessed on a weekly basis and then copied from development into the Web production environment. The customer segmentation model will be used for marketing processes such as organizing marketing campaigns that take some time to complete, so the model will not be reprocessed often.

TIP:
Reprocessing Mining Models Drops Cluster Names
Another reason that the customer segmentation model is not reprocessed often is that the carefully selected cluster names such as Defector and Newbie are replaced with the default Cluster 1 and Cluster 2 names during processing. This is also important to know during development when you are designing your model, because you will be losing any cluster names every time you need to change anything in the model.

Next Steps

You can leverage data mining for various business requirements in many ways, and you have a whole set of new possibilities available if we add the Page Hits fact table to the data warehouse. This would involve handling a large fact table, which has ramifications for the relational database design, ETL processes, and even the cube structure. See Chapter 11, "Very Large Data Warehouses," for a full description of the issues associated with very large databases.

Sequence Clustering to Build Smarter Web Sites

Each visit has an associated path that the user took through the pages of the Web site. We could use this data with the Sequence Clustering algorithm, which finds clusters of cases that contain similar paths in a sequence. This mining model could then be used in the Web site to suggest the next page that the user might like to visit.

Other Data Mining Possibilities

This chapter has given a basic introduction to the rich and broad set of applications possible using the algorithms in Analysis Services data mining. One area that we have only scratched the surface of is prediction. For example, applications that require predictions of an attribute are possible using the classification algorithms including Decision Trees, Neural Network, and Naive Bayes; and continuous variables such as future profit levels can be predicted by the Time Series and Decision Trees algorithms.

Using Data Mining in Integration Services to Improve Data Quality

One of the major scenarios that we have not looked at in this chapter is the use of data mining in Integration Services. We could create a clustering model against a subset of data that is already in the data warehouse and is known to have clean, correct values, and then query this model in an Integration Services package that loads new data to determine the probability that each new record is valid. Records that are selected as likely "bad data" can be split out during the load process into a separate table for further validation or human checking and correction.

Integration Services also has other data mining features, such as loading data directly into data models within the data flow and specifying samples of data to be used for training models rather than just using all the data.

Summary

We added the Internet visit information to the data warehouse and Analysis Services database and built Integration Services packages to load the data from the e-commerce system's database. We created a mining model to add customer segmentation to the cube, and another model to supply product recommendations to the Web site. We decided not to use the e-commerce application's built-in BI features because the objectives required extensive data from the existing data warehouse, which was not available in the e-commerce database.

Marketing activities, such as Internet advertising and direct mail, can now be targeted more effectively at customers based on their use of the Web site and their customer profiles. The high-performance cross-sell feature on the Web site is recommending additional DVDs that the customer might like to purchase, hopefully leading to additional items sold per transaction.

Because many of the interesting measures in the Internet activity fact data are measures of the time elapsed between two dates, we added a set of time span calculations to the fact views that were used by the cube. To calculate the time up until the present day, the data load date was used rather than the current system date.

We used the Microsoft Clustering algorithm to create a customer segmentation mining model and data mining dimension so that analysts can use clusters of customers such as Newbies and Store-Only Buyers. We used the Microsoft Association Rules algorithm to create a product recommendations model, and added a DMX query to the e-commerce Web application to suggest a list of possible DVDs that a customer might also purchase.

The product recommendations mining model was deployed to a production server that can be accessed by the e-commerce Web application, and the security was configured so that the Web application can query the mining model. A new operations task is to periodically reprocess the product recommendations mining model so that it is kept current with the latest data.