DMX Language

Understand DMX concepts.

Apply DMX to a data mining model.

Overview

- History of DMX
- Introduction
- objects
- Query Syntax
- Prediction

- DMX was first introduced in the OLE DB for Data Mining specification authored by Microsoft in conjunction with other vendors in 1999.
- The goal of DMX is to define common concepts and common query expressions for the data mining world.
- Similar in spirit to SQL.

Consists of

- DDL (Data definition Language)

-Create new data mining models and structures, export and import mining structures, copy or transfer data from one mining model to another and delete existing data mining models and mining structures.

- DML (Data Manipulation Language)

-The DML of the DMX is used to search and browse data in the data mining models, update the data mining models by insertion and updating of the data and derive predictions using the prediction querv.

DMX Objects

- DMX is used to
- create the structure of new data mining models
- train these models
- browse
- manage
- predict

- Two major objects that are used to manifest this transformation:
- The mining structure
- The mining model

Mining Structure

- Defined as a list of columns, with their data types and information describing how they should be handled.
- When a mining structure is processed, it contains a compressed (cache) or copy of the source data.
- This cache is used to train any models that are subsequently added to the structure. which can be queried to return its data or the distinct states that exist in any structure column.
- The cache is only maintained temporarily, and can be dropped at any time.

Mining Model

- Object that transforms rows of data into cases and performs the machine learning using a specified data mining algorithm.
- A subset of columns from the structure, how those columns are to be used as attributes along with the algorithm and parameters to perform machine learning on the structure data.
- Statistics about predictions are available as well,
- The learned patterns themselves can be queried to discover what the algorithm found.
- These patterns are generally referred to as the - model content.

Query Syntax

- DMX statements are used to create, process, delete, copy, browse, and predict against data mining models.
- The three basic steps for data mining process are:
- Creation
- Prediction
- Training

Creating mining Structure

- • Creating mining structures is similar to creating tables in SQL.
- Syntax: CREATE (SESSION] MINING STRUCTURE ([(<column definition list>)]) Structure unique name for the structure.
- column definition list comma-separated list of column definitions

Create Mining Structure

- The following example creates a new mining structure called New Mailing.

```
CREATE MINING STRUCTURE [New Mailing]
 ( CustomerKey LONG KEY,
Gender TEXT DISCRETE,
[Number Cars Owned] LONG DISCRETE,
 [Bike Buyer] LONG DISCRETE )
```

Entry of the fields of the mining structure

Name of field, data type, content type

- [Number Cars Owned] LONG DISCRETE

Data Type

- Gives the algorithm information about the type of data in the columns.
- Eg. Integer, long , text , boolean

Datatype, Content Type

- Needed to be specified so the algorthims can be implemented efficiently.

Content Type

Describes the behavior of the content that the column

If it repeats in a specific interval, such as days of the week, you can specify the content type of that column as cyclical.

Query mining structure adjoining data from data base

- SELECT
- t.[CommuteDistance],
- t.[Age],
- t.[Gender],
- t.[HouseOwnerFlag],
- t.[LastName],
- t.[FirstName],
- t.[EnglishOccupation],
- t.[EnglishOccupation],
- t.[BikeBuyer]

From part – specify mining model

- From
- [Target Mail Decision Tree]
- PREDICTION JOIN

Specify data base table

- OPENQUERY([Adventure Works DW2012],
- 'SELECT
- [CommuteDistance],
- [Age],
- [Gender],
- [HouseOwnerFlag],
- [LastName],
- [FirstName],
- [EnglishOccupation],
- [BikeBuyer],
- [MaritalStatus],
- [YearlyIncome],
- [EnglishEducation]
- FROM
- [dbo].[vTargetMail]
- ') AS t

Joining Model columns to database table

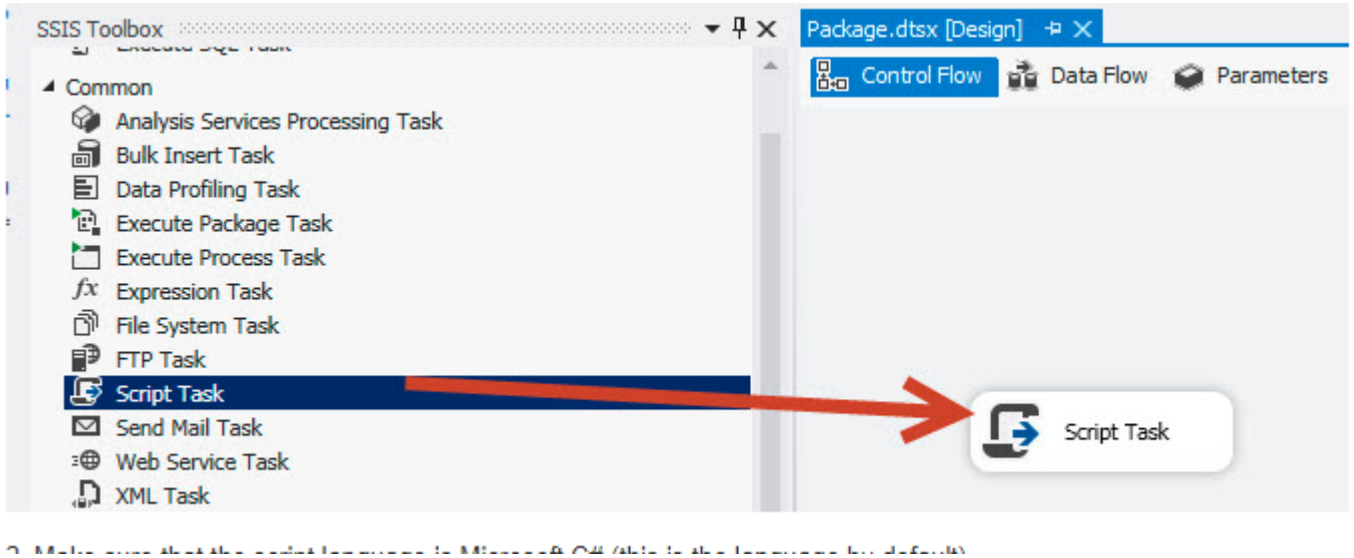
- ON
- [Target Mail Decision Tree].[Marital Status] = t.[MaritalStatus] AND
- [Target Mail Decision Tree].[Gender] = t.[Gender] AND
- [Target Mail Decision Tree].[Yearly Income] = t.[YearlyIncome] AND
- [Target Mail Decision Tree].[English Education] = t.[EnglishEducation] AND
- [Target Mail Decision Tree].[English Occupation] = t.[EnglishOccupation] AND
- [Target Mail Decision Tree].[House Owner Flag] = t.[HouseOwnerFlag] AND
- [Target Mail Decision Tree].[Commute Distance] = t.[CommuteDistance] AND
- [Target Mail Decision Tree].[Age] = t.[Age] AND
- [Target Mail Decision Tree].[Bike Buyer] = t.[BikeBuyer]

Results

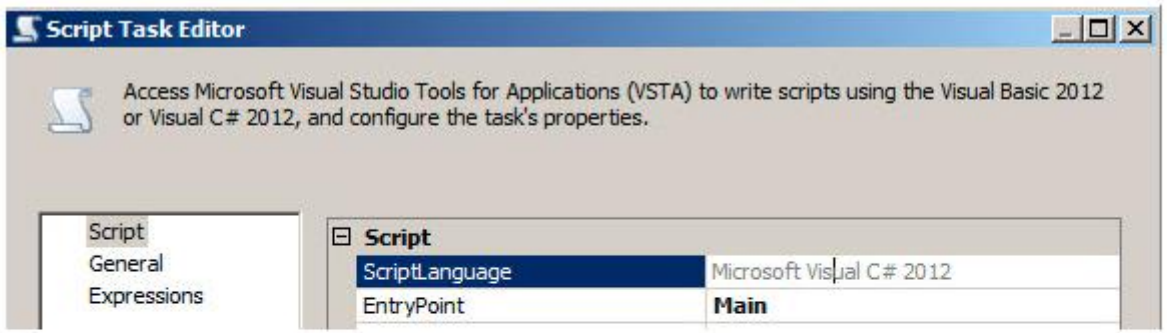
CommuteDistance	Age	Gender	HouseOwnerFlag	LastName	FirstName	EnglishOccupation	EnglishOccupation	BikeBuyer
1-2 Miles	52	M	1	Yang	Jon	Professional	Professional	1
0-1 Miles	53	M	0	Huang	Eugene	Professional	Professional	1
2-5 Miles	53	M	1	Torres	Ruben	Professional	Professional	1
5-10 Miles	50	F	0	Zhu	Christy	Professional	Professional	1
1-2 Miles	50	F	1	Johnson	Elizabeth	Professional	Professional	1
5-10 Miles	53	M	1	Ruiz	Julio	Professional	Professional	1
5-10 Miles	52	F	1	Alvarez	Janet	Professional	Professional	1
0-1 Miles	54	M	1	Mehta	Marco	Professional	Professional	1
10+ Miles	54	F	1	Verhoff	Rob	Professional	Professional	1
5-10 Miles	54	M	0	Carlson	Shannon	Professional	Professional	1
5-10 Miles	54	F	0	Suarez	Jacquelyn	Professional	Professional	1
10+ Miles	55	M	1	Lu	Curtis	Professional	Professional	1
1-2 Miles	50	F	1	Walker	Lauren	Management	Management	0
0-1 Miles	50	M	1	Jenkins	Ian	Management	Management	0
1-2 Miles	50	F	0	Bennett	Sydney	Management	Management	0
5-10 Miles	39	F	0	Young	Chloe	Skilled Manual	Skilled Manual	1
5-10 Miles	39	M	1	Hill	Wyatt	Skilled Manual	Skilled Manual	1
5-10 Miles	74	F	1	Wang	Shannon	Skilled Manual	Skilled Manual	1
5-10 Miles	74	M	1	Rai	Clarence	Clerical	Clerical	1
5-10 Miles	40	M	0	Lal	Luke	Skilled Manual	Skilled Manual	0
1-2 Miles	40	M	0	King	Jordan	Skilled Manual	Skilled Manual	1

Integration of Datamining Models

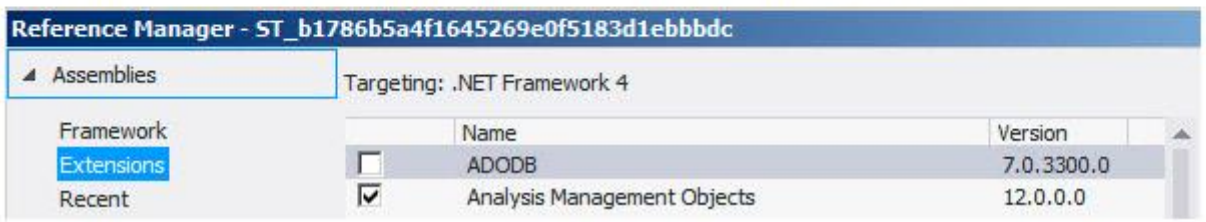
- **Programming with C# and AMO**



Selecting C# and appropriate libraries



3. You will also need to add the Analysis Management Objects, as you did in part 28.



Including the libraries

4. Add the "Using Microsoft.AnalysisServices" to the top of your code.

```
#region Namespaces
using System;
using System.Data;
using Microsoft.SqlServer.Dts.Runtime;
using System.Windows.Forms;
using Microsoft.AnalysisServices;
#endregion
```

5. Now add the following code:

```
public void Main()
{
// TODO: Add your code here
        //Add the server
        Server DM_Server= new Server();
        //Add the database
        Database AS_Database=new Database();
        //Connect to the Data Mining
        DM_Server.Connect("Data Source=infra4;Initial Catalog=AdventureWorksDW2014Multidimensional-EE");
        //Get the AS Database
        AS_Database = DM_Server.Databases["AdventureWorksDW2014Multidimensional-EE"];
        //Show the database name
        MessageBox.Show("Database: " + AS_Database.Name);
```

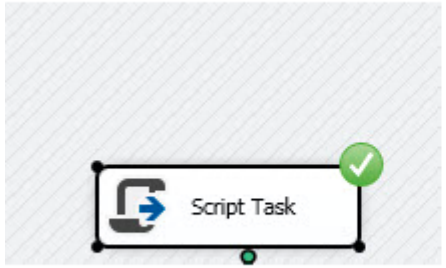
C# Call to process the mining model

```
    //Get the total days that the Mining structure was not processed
    int diff = Convert.ToInt32((Now - pd).TotalDays);

    //If the number of days of last process is higher than 5, then process the structure
    if (diff > 5)
    {
        AS_Database.MiningStructures[0].Process();
    }
    Dts.TaskResult = (int)ScriptResults.Success;
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
```

Run script in ETL framework

2. Now, run the script.



3. If you check in the SSMS, the Mining Structure last processed property was updated.

C# Client side code to call model

```
private void button1_Click(
   object sender, EventArgs e)
{
   string strASConnString =
      CustomerSegmentationClientWin.
      Properties.Settings.Default.ASConnString;
   try
   {
      IPredictCustomerMDXRequester objRequester =
         new PredictCustomerMDXRequester();

      this.m_tbResult.Text = "";
      this.m_tbResult.Text =
         objRequester.Command(
         strASConnString,
         this.m_tbCity.Text.Trim(),
         this.m_tbContactTitle.Text.Trim());
   }
   catch (Exception ex)
   {
      System.Windows.Forms.MessageBox.Show(
         ex.ToString());
   }
}
```