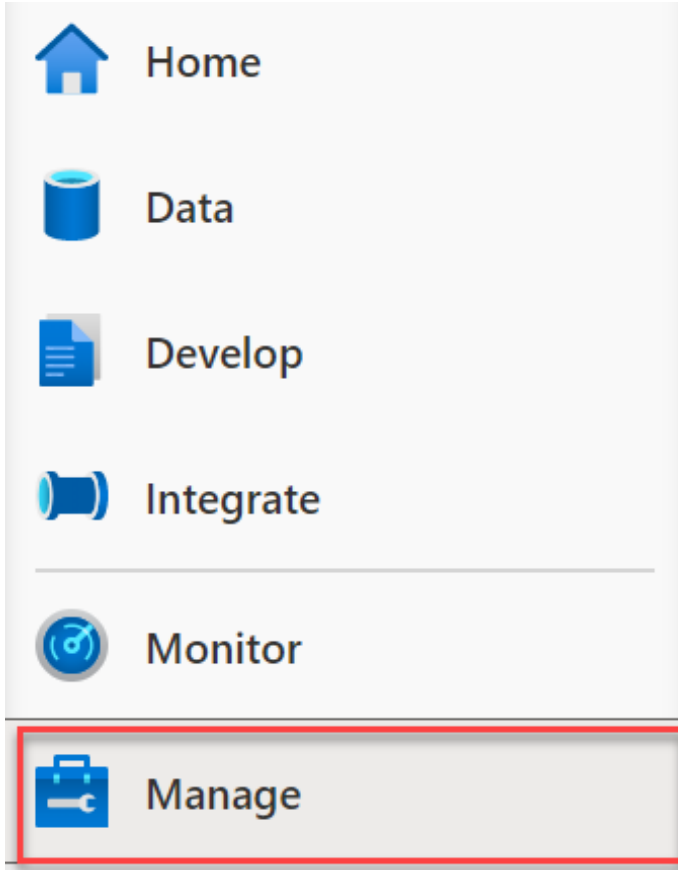


Machine Learning

Lab pre-requisite

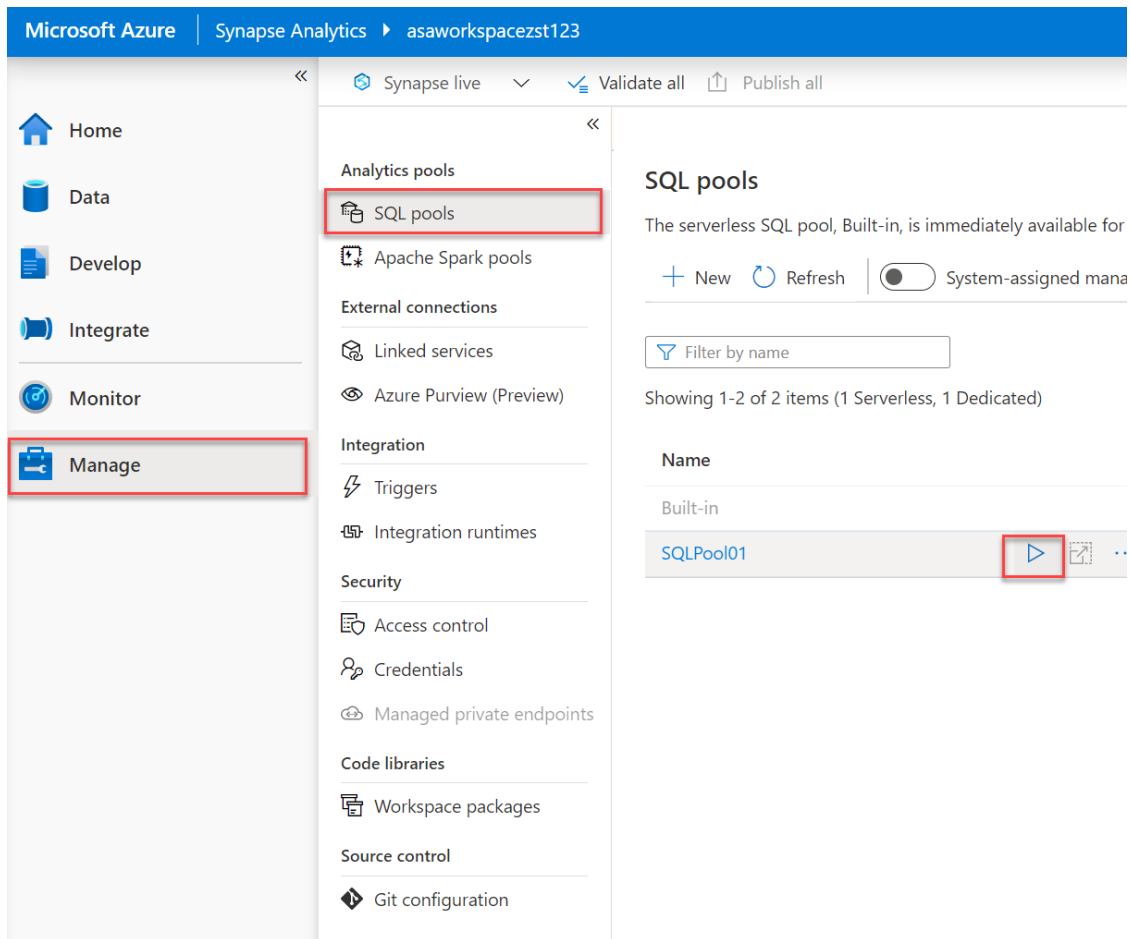
Start the SQL Pool in your lab environment.

1. Open the Synapse Studio workspace and navigate to the **Manage** hub.



The Manage menu item is highlighted.

2. From the center menu, select **SQL pools** from beneath the **Analytics pools** heading. Locate SQLPool01, and select the **Resume** button.



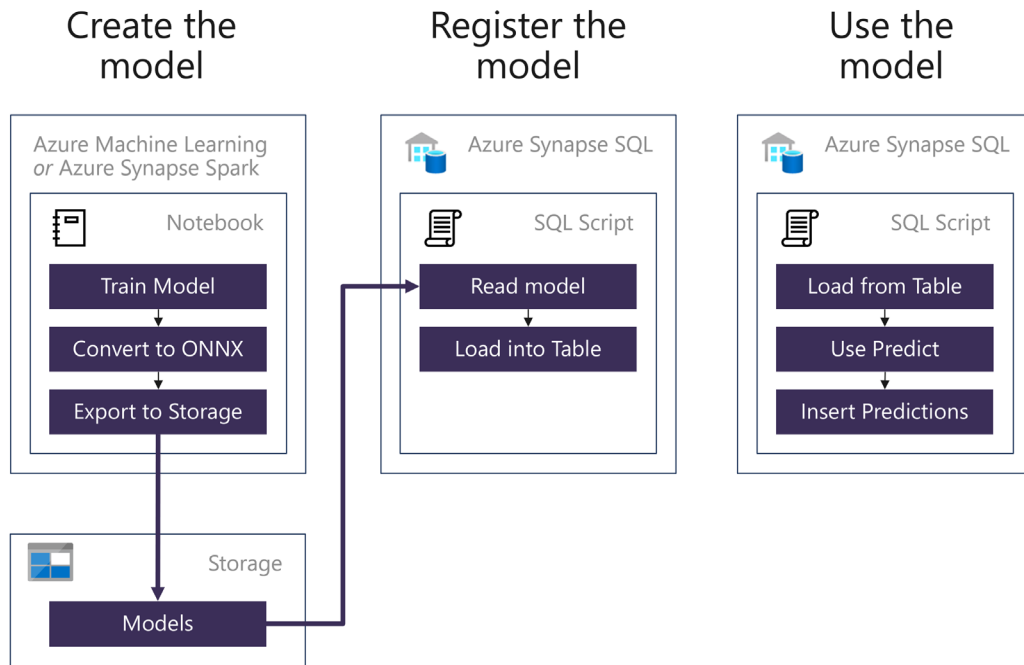
The Manage menu item is selected, with SQL pools selected from the center menu. The resume button is selected next to the SQLPool01 item.

Lab overview

Azure Synapse Analytics provides a unified environment for both data science and data engineering. What this means in practice, is that your data scientists can train and deploy models using Azure Synapse Analytics and your data engineers can write T-SQL queries that use those models to make predictions against tabular data stored in a SQL Pool database table.

In this lab, you will create several machine learning models using AutoML with Spark compute and Spark libraries like Synapse Machine Learning (Synapse ML). You will also experience the integration between Synapse ML and Cognitive Services. Finally, you will use one of the models registered in Azure Machine Learning to make predictions using the T-SQL Predict statement.

For context, the following are the high level steps taken to create a Spark ML based model and deploy it so it is ready for use from T-SQL.



The process for registering and using a model

All of the steps are performed within Synapse Studio.

- Within a notebook, a data scientist will:
 - a. Train a model using Synapse ML, the machine learning library included with Apache Spark. Models can also be trained using other approaches, including by using Azure Machine Learning Automated ML. The main requirement is that the model format must be supported by ONNX.
 - b. Deploy the ONNX model to a table in the SQL Pool database using Synapse Studio.
- To use the model for making predictions, in a SQL Script a data engineer will:
 - a. Read the model into a binary variable by querying it from the table in which it was stored.
 - b. Execute a query using the FROM PREDICT statement as you would a table. This statement defines both the model to use and the query to execute that will provide the data used for prediction. You can then take these predictions and insert them into a table for use by downstream analytics applications.

What is ONNX? **ONNX** is an acronym for the Open Neural Network eXchange and is an open format built to represent machine learning models, regardless of what frameworks were used to create the model. This enables model portability, as models in the ONNX format can be run using a wide variety of frameworks, tools,

runtimes and platforms. Think of it like a universal file format for machine learning models.

Exercise 1 - Synapse Machine Learning in action

Open the Lab 06 - Part 1 - Synapse ML notebook (located in the Develop hub, under Notebooks in Synapse Studio) and run it step by step to complete this exercise. Some of the most important tasks you will perform are:

- Install Synapse ML in a Spark session
- Use Synapse ML to perform Entity Recognition with Cognitive Services
- Prepare and analyze data
- Train classifier using Synapse ML and LightGBMClassifier
- Perform predictions and analyze classifier performance

Please note that each of these tasks will be addressed through several cells in the notebook.

Note: Please attach to SparkPool02, and ensure the proper Azure **location** is specified in the second code cell(matching the region of the deployed Cognitive Services account).

Lab 06 - Part 1 - Syn...

Run all | Undo | Publish | Outline | Attach to SparkPool02

Not started

Retrieve the Cognitive Services credentials and create the test dataset.

```
1 key = mssparkutils.credentials.getSecret('asakeyvault627867', 'COGNITI
2 location = 'eastus'
3
4 df = spark.createDataFrame(data=[
5     [1, "Muad'Dib learned rapidly because his first training was :
6     [2, "It's the ship that made the Kessel run in less than twelv
7 ],
8     schema=["id", "text"])
```

[] Press shift + enter to run

A Spark notebook displays attached to SparkPool02 and there is a value in the location variable.

Exercise 2 - Training and registering models with AutoML

Open the Lab 06 - Part 2 - AutoML with Spark notebook (located in the Develop hub, under Notebooks in Synapse Studio) and run it step by step to complete this exercise. Some of the most important tasks you will perform are:

- Use Azure Machine Learning AutoML with Synapse Spark compute to train a classification model (the local Spark session of the notebook is used as a compute resource by AutoML)
- Register the ONNX version of the model in the AML model registry using MLFlow
- Persist test data to the dedicated Synapse SQL pool

Please note that each of these tasks will be addressed through several cells in the notebook.

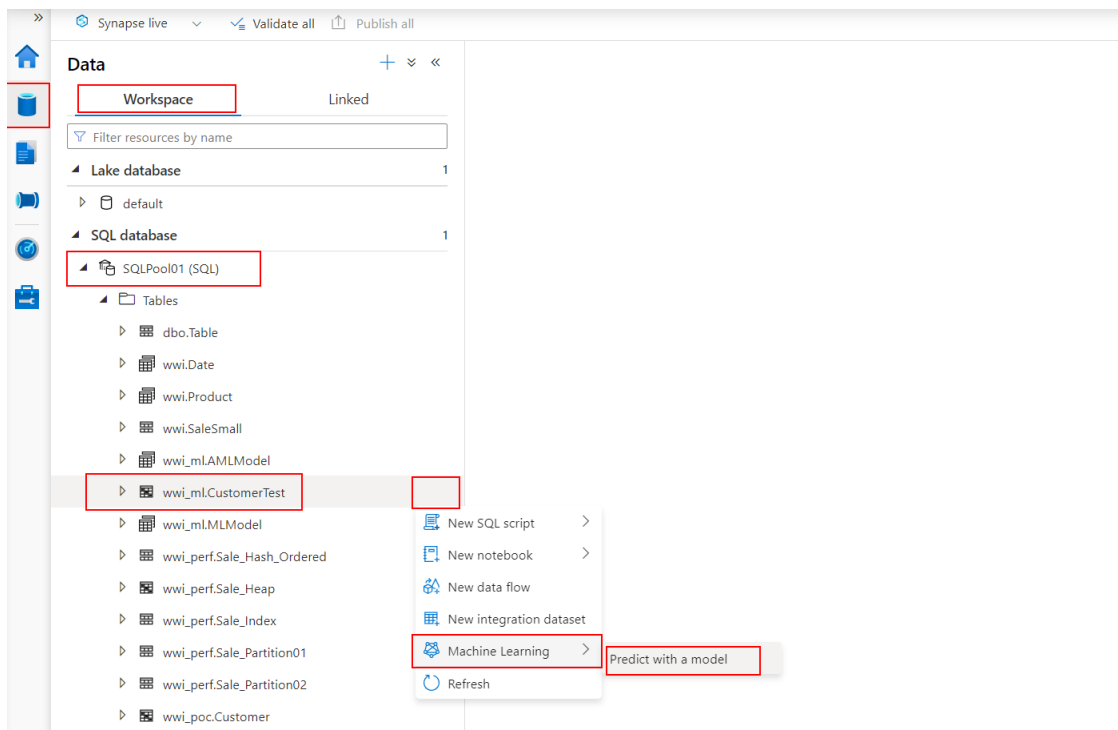
Exercise 3 - Using registered models in Synapse Analytics

NOTE:

Successfully completing Exercise 2 is a prerequisite for this exercise.

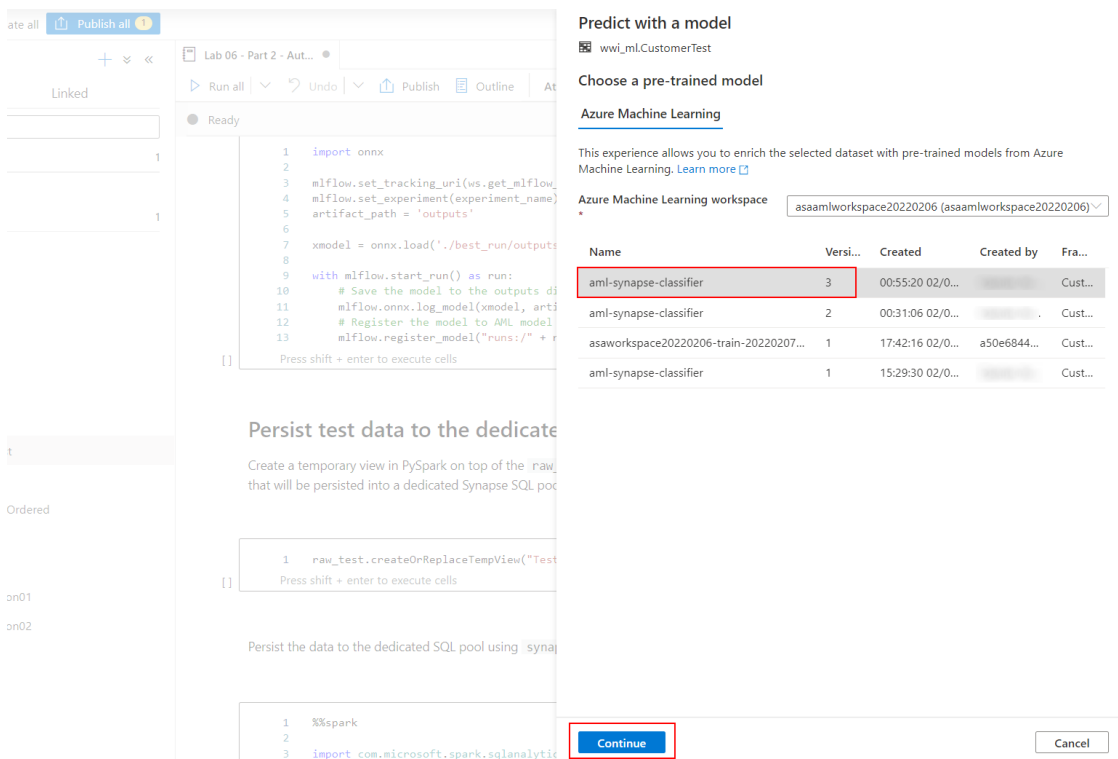
In this exercise you will use the model registered in Exercise 2 to perform predictions using the AML integration features of Synapse Studio.

1. In Synapse Studio, select the Data hub, Workspace section, SQLPool01 SQL database, and locate the `wwi_ml.CustomerTest` table (the one created at the end of Exercise 2).
2. Select the context menu of the table and then select Machine Learning -> Predict with a model.



Start prediction with AML model

3. In the Choose a pre-trained model dialog, select the highest version of the model named aml-synapse-classifier and then select Continue.



Select latest version of registered model

4. Leave the column mappings unchanged and select Continue.

NOTE:

The model schema generated with MLFlow and used to register the model enables Synapse Studio to suggest the mappings.

The screenshot shows the Synapse Studio interface. On the left, a Jupyter notebook is open with a Python script for MLFlow model registration. On the right, a 'Predict with a model' dialog is displayed for the model 'wwi_ml.CustomerTest'. The dialog includes a 'Map columns' section with a link to 'Learn more', an 'Input mapping' table, and an 'Output mapping' table. The 'Continue' button at the bottom of the dialog is highlighted with a red box.

Source column	Model input	Input type
Cost	Cost	real
Size	Size	real
Price	Price	real
PrimaryBrandId	PrimaryBrandId	bigint
GenderId	GenderId	bigint
MaritalStatus	MaritalStatus	bigint
LowerIncomeBound	LowerIncomeBound	real
UpperIncomeBound	UpperIncomeBound	real

Model output	Output type
label_out	bigint

Continue Back Cancel

Model column mappings

5. In the Store objects in the database dialog, select the following:

- Script type: View
- View name: enter wwi_ml.CustomerPrediction
- Database table: Existing table
- Existing target table: select the wwi_ml.AMLModel table

Select Deploy model + open script to continue. Synapse Studio will deploy the model into the AMLModel table and create SQL scoring script for you.

Lab 06 - Part 2 - Aut...

Run all Undo Publish Outline At

Ready

```

1 import onnx
2
3 mlflow.set_tracking_uri(ws.get_mlflow_
4 mlflow.set_experiment(experiment_name)
5 artifact_path = 'outputs'
6
7 xmodel = onnx.load('./best_run/outputs
8
9 with mlflow.start_run() as run:
10     # Save the model to the outputs di
11     mlflow.onnx.log_model(xmodel, arti
12     # Register the model to AML model
13     mlflow.register_model("runs:/" + r

```

Persist test data to the dedicate

Create a temporary view in PySpark on top of the raw, that will be persisted into a dedicated Synapse SQL pool

```

1 raw_test.createOrReplaceTempView("Test

```

Persist the data to the dedicated SQL pool using syna

```

1 %%spark
2
3 import com.microsoft.spark.sqlanalytic

```

Predict with a model

wwi_ml.CustomerTest

Store objects in the database

Save the prediction script as a stored procedure or a view so you can reuse it when needed. [Learn more](#)

Script type *

☐ Stored procedure ☒ View

A view is a virtual table extracted from a database. Views can be used in a number of different ways to improve the quality of your solution. [Learn more](#)

View name *

wwi_ml.CustomerPrediction

Load model into a database table

Create a new database table or use an existing table to store the machine learning model. [Learn more](#)

Database table *

☒ Existing table ☐ Create new

Existing target table *

wwi_ml.AMLModel

Deploy model + open script Back Cancel

Deploy ML model to database

6. Run the generated SQL script.

Synapse live Validate all Publish all

Lab 06 - Part 2 - Aut... SQL script 1

Run Undo Publish Query plan Connect to SQLPool01 Use database SQLPool01

```

1 -- Create a view to score machine learning models.
2 CREATE VIEW wwi_ml.CustomerPrediction
3 AS
4 -- Select input scoring data and assign aliases.
5 WITH InputData AS
6 (
7     SELECT
8         CAST([Cost] AS [real]) AS [Cost],
9         CAST([Size] AS [real]) AS [Size],
10        CAST([Price] AS [real]) AS [Price],
11        CAST([PrimaryBrandId] AS [bigint]) AS [PrimaryBrandId],
12        CAST([GenderId] AS [bigint]) AS [GenderId],
13        CAST([MaritalStatus] AS [bigint]) AS [MaritalStatus],
14        CAST([LowerIncomeBound] AS [real]) AS [LowerIncomeBound],
15        CAST([UpperIncomeBound] AS [real]) AS [UpperIncomeBound]
16    FROM [wwi_ml].[CustomerTest]
17 )
18 -- Using T-SQL Predict command to score machine learning models.
19 SELECT *
20 FROM PREDICT ([MODEL] = (SELECT [model] FROM wwi_ml.AMLModel WHERE [ID] = 'aml-synapse-cl
21    DATA = InputData,
22    RUNTIME = ONNX) WITH ([label_out] [bigint])
23 GO
24
25 -- Display the results.
26 SELECT * FROM wwi_ml.CustomerPrediction

```

Properties

General Related (0)

Name *

SQL script 1

Description

Type

.sql script

Size

977 bytes

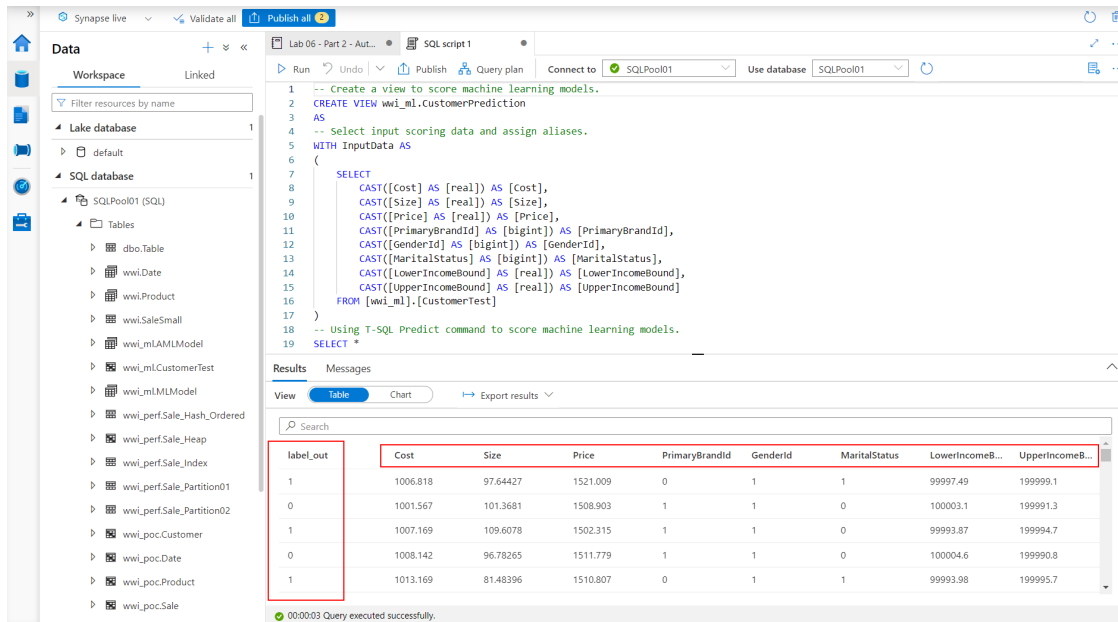
Results settings per query

☒ First 5000 rows (default)

☐ All rows

Run the PREDICT T-SQL statement

7. Observe the results of the prediction.



The screenshot displays the Synapse live interface. On the left, the 'Data' pane shows a workspace with a linked 'Lake database' and 'SQL database'. The 'SQL database' contains a table named 'wwi_ml.CustomerPrediction'. The main pane shows a SQL script with the following content:

```
1 -- Create a view to score machine learning models.
2 CREATE VIEW wwi_ml.CustomerPrediction
3 AS
4 -- Select input scoring data and assign aliases.
5 WITH InputData AS
6 (
7     SELECT
8         CAST([Cost] AS [real]) AS [Cost],
9         CAST([Size] AS [real]) AS [Size],
10        CAST([Price] AS [real]) AS [Price],
11        CAST([PrimaryBrandId] AS [bigint]) AS [PrimaryBrandId],
12        CAST([GenderId] AS [bigint]) AS [GenderId],
13        CAST([MaritalStatus] AS [bigint]) AS [MaritalStatus],
14        CAST([LowerIncomeBound] AS [real]) AS [LowerIncomeBound],
15        CAST([UpperIncomeBound] AS [real]) AS [UpperIncomeBound]
16    FROM [wwi_ml].[CustomerTest]
17 )
18 -- Using T-SQL Predict command to score machine learning models.
19 SELECT
```

The 'Results' pane shows the output of the query, which is a table with 9 columns: 'label_out', 'Cost', 'Size', 'Price', 'PrimaryBrandId', 'GenderId', 'MaritalStatus', 'LowerIncomeB...', and 'UpperIncomeB...'. The table contains 6 rows of data.

label_out	Cost	Size	Price	PrimaryBrandId	GenderId	MaritalStatus	LowerIncomeB...	UpperIncomeB...
1	1006.818	97.64427	1521.009	0	1	1	99997.49	199999.1
0	1001.567	101.3681	1508.903	1	1	0	100003.1	199991.3
1	1007.169	109.6078	1502.315	1	1	0	99993.87	199994.7
0	1008.142	96.78265	1511.779	1	1	0	100004.6	199990.8
1	1013.169	81.48396	1510.807	0	1	1	99993.98	199995.7

The status bar at the bottom indicates '00:00:03 Query executed successfully.'

View prediction results