Azure for Research - Azure ML Tutorial

Set up Azure ML 1

Experiment Overview 2

Create a model 3

Train the model 3

Score and test the model 3

Step 1: Get data 3

Step 2: Preprocess data 7

Step 3: Define features 10

Step 4: Choose and apply a learning algorithm 12

Step 5: Predict new automobile prices 16

Additional Learning– 19

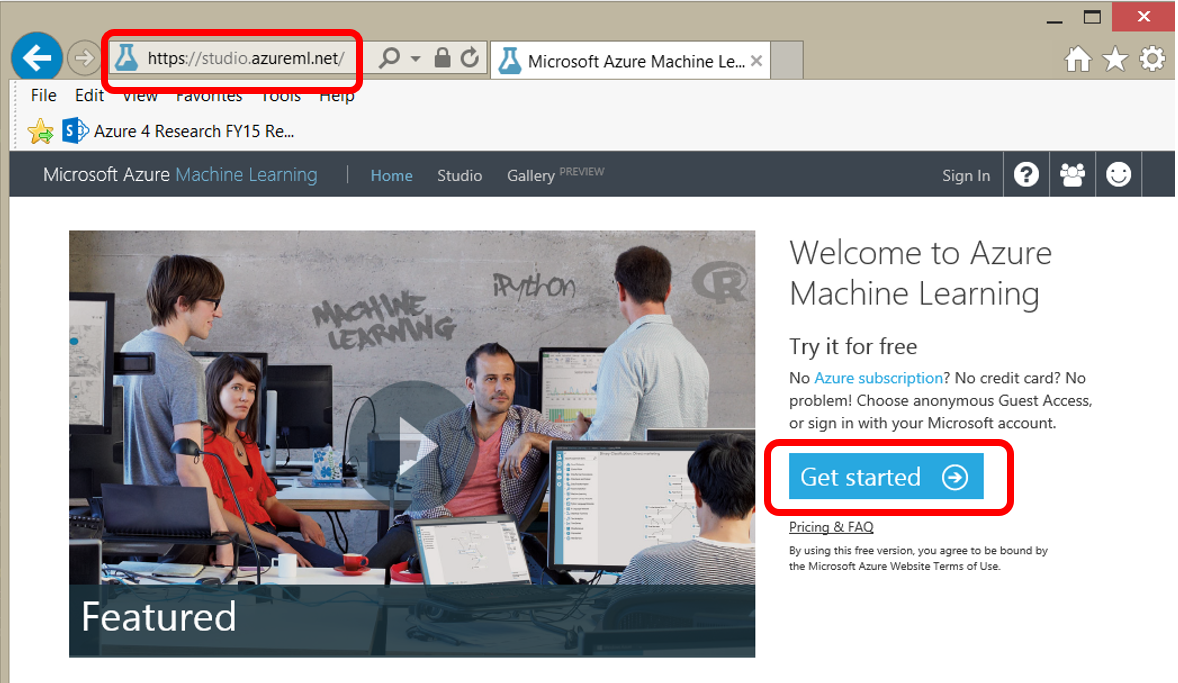
Compare two models 19

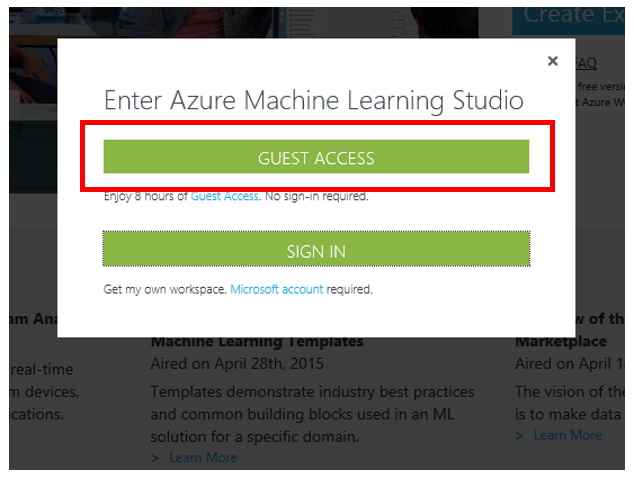
Publish the model as a web service 20

## Set up Azure ML

In the web browser, navigate to [**http://studio.azureml.net**](http://studio.azureml.net)and sign up with Guest Access

*Alternately sign in to your existing Azure account if you have one.*





## Experiment Overview

In this machine learning tutorial, you will build and evaluate a Machine Learning model to predict automobile prices. You will follow five basic steps to build this experiment in Machine Learning Studio in order to create, train, and score your model:

### Create a model

* + Step 1: Get data
  + Step 2: Preprocess data
  + Step 3: Define features

### Train the model

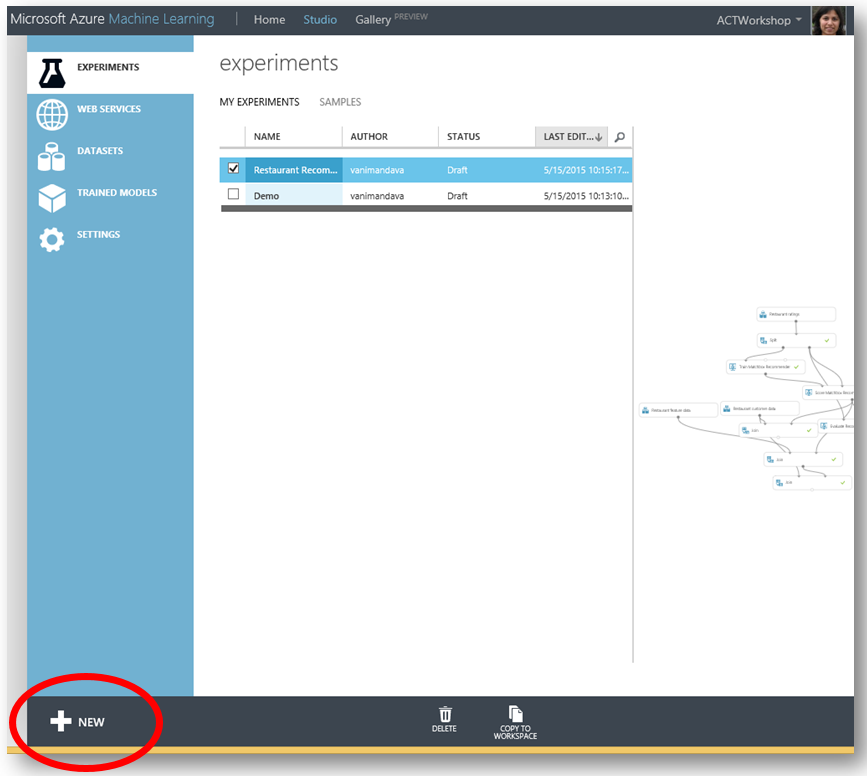
* + Step 4: Choose and apply a learning algorithm

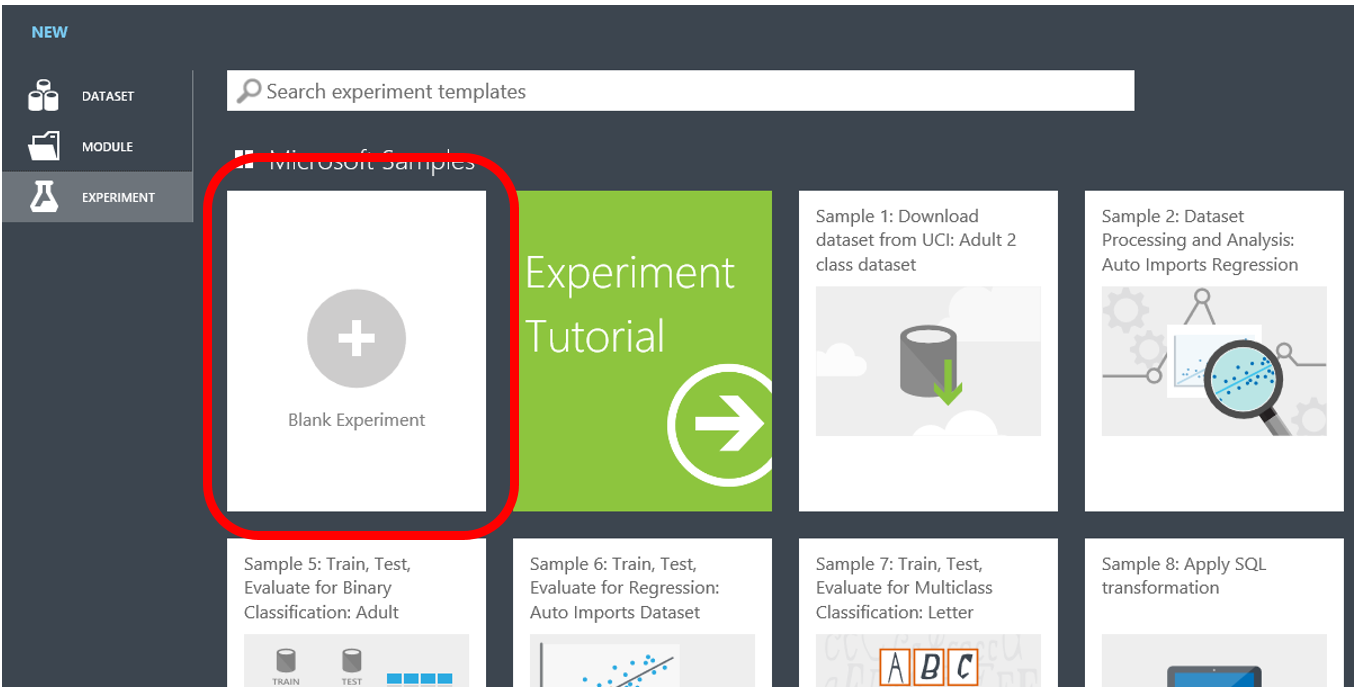
### Score and test the model

* + [Step 5: Predict new automobile prices](https://azure.microsoft.com/en-us/documentation/articles/machine-learning-create-experiment/#step-5-predict-new-automobile-prices)

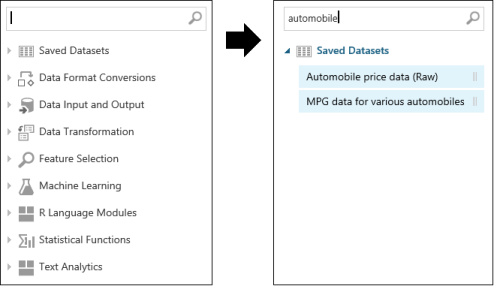
## Step 1: Get data

There are a number of sample datasets included with Machine Learning Studio, and you can import data from many sources. For this example, we will use the included sample dataset, **Automobile price data (Raw)**. This dataset includes entries for a number of individual automobiles, including information such as make, model, technical specifications, and price.

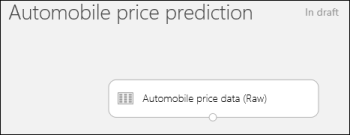
1. Start a new experiment by clicking **+NEW** at the bottom of the Machine Learning Studio window, select **EXPERIMENT**, and then select "Blank Experiment". 



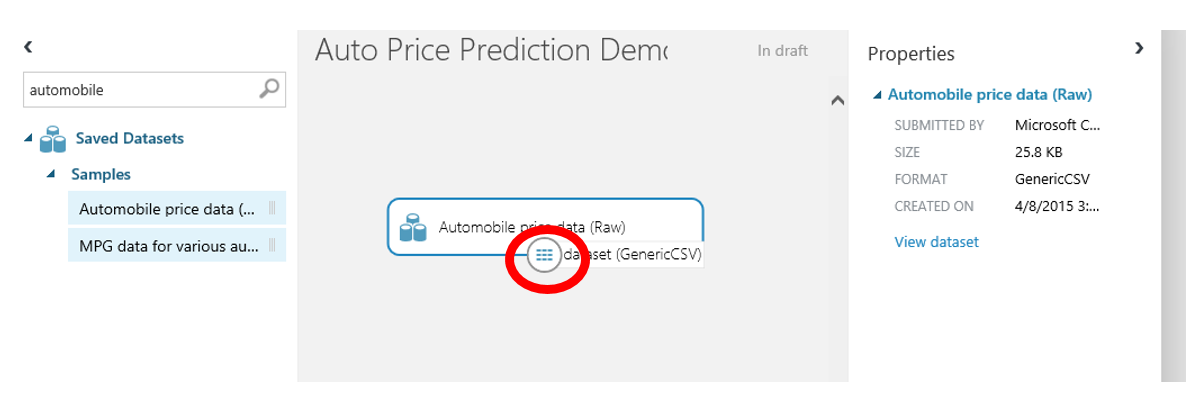
1. Select the default experiment name at the top of the canvas and rename it to something meaningful, for example, **Automobile price prediction**.
2. To the left of the experiment canvas is a palette of datasets and modules. Type **automobile** in the search box at the top of this palette to find the dataset labeled **Automobile price data (Raw)**.

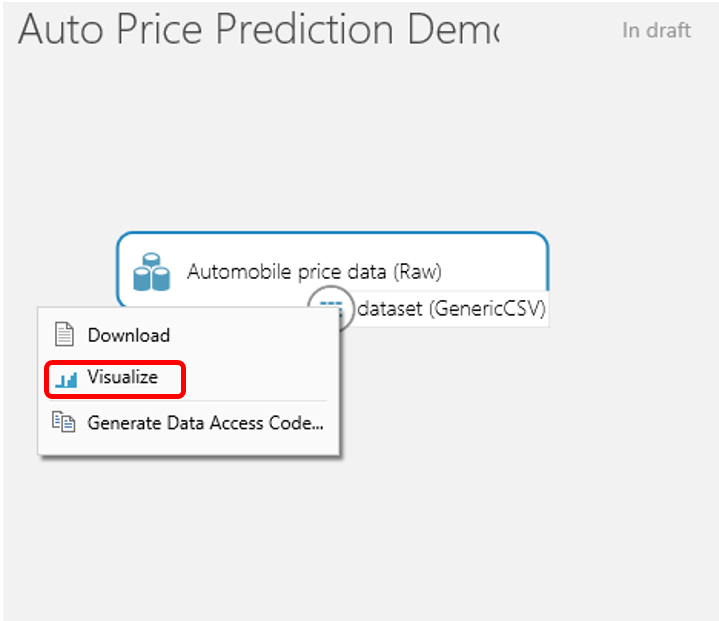


1. Drag the dataset to the experiment canvas.

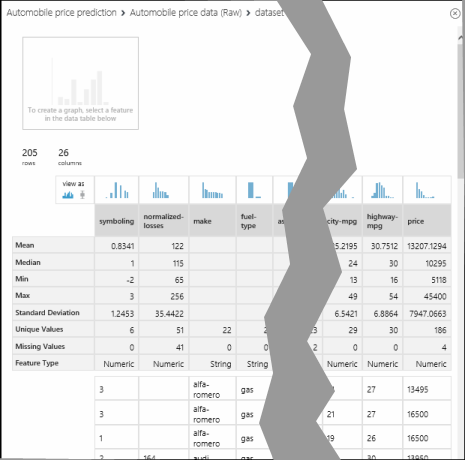


To see what this data looks like, click the output port at the bottom of the automobile dataset and select **Visualize**.





The variables in the dataset appear as columns, and each instance of an automobile appears as a row. The far-right column (column 26 and titled "price") is the target variable we're going to try to predict.



Close the visualization window by clicking the "**x**" in the upper-right corner.

## Step 2: Preprocess data

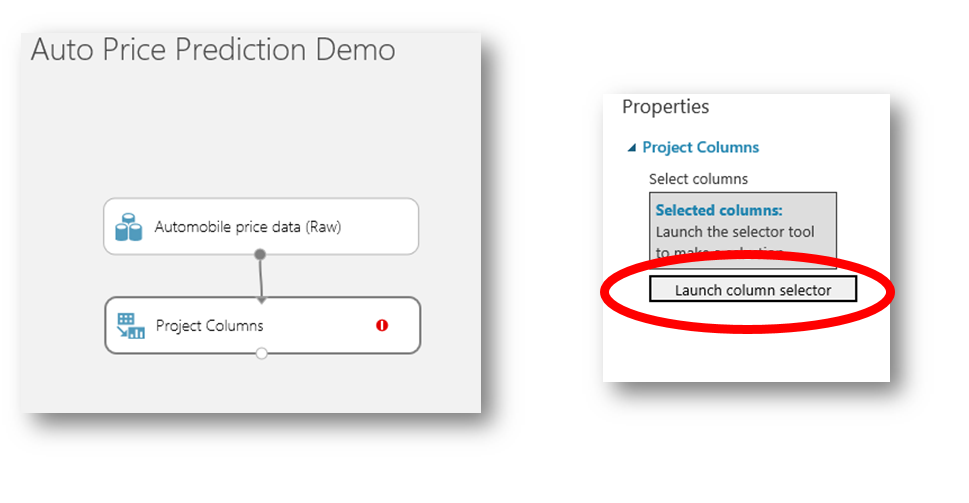
A dataset usually requires some preprocessing before it can be analyzed. You may have noticed the missing values present in the columns of various rows. These missing values need to be cleaned so the model can analyze the data properly. In our case, we'll remove any rows that have missing values. Also, the **normalized-losses** column has a large proportion of missing values, so we'll exclude that column from the model altogether.

TIP:

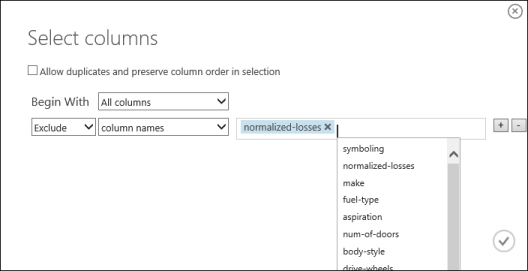
Cleaning the missing values from input data is a prerequisite for using most of the modules.

First we'll remove the **normalized-losses** column, and then we'll remove any row that has missing data.

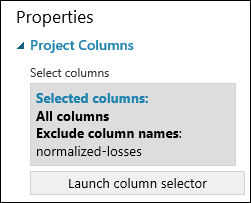
1. Type **project columns** in the search box at the top of the module palette to find the [Project Columns](https://msdn.microsoft.com/library/azure/1ec722fa-b623-4e26-a44e-a50c6d726223/) module, then drag it to the experiment canvas and connect it to the output port of the **Automobile price data (Raw)** dataset. This module allows us to select which columns of data we want to include or exclude in the model.
2. Select the **Project Columns** module and click **Launch column selector** in the properties pane.



* + Make sure **All columns** is selected in the filter drop-down list, **Begin With**. This directs **Project Columns** to pass through all the columns (except those we're about to exclude).
  + In the next row, select **Exclude** and **column names**, and then click inside the text box. A list of columns is displayed. Select **normalized-losses** and it will be added to the text box.
  + Click the check mark (OK) button to close the column selector.



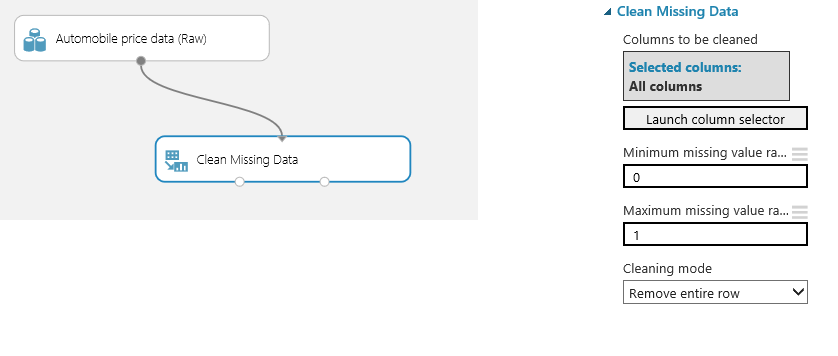
The properties pane for **Project Columns** indicates that it will pass through all columns from the dataset except **normalized-losses**.



TIP:

You can add a comment to a module by double-clicking the module and entering text. This can help you see at a glance what the module is doing in your experiment. In this case, double-click the [Project Columns](https://msdn.microsoft.com/library/azure/1ec722fa-b623-4e26-a44e-a50c6d726223/) module and type the comment "Exclude normalized-losses."

1. Drag the [Clean Missing Data](https://msdn.microsoft.com/library/azure/d2c5ca2f-7323-41a3-9b7e-da917c99f0c4/) module to the experiment canvas and connect it to the [Project Columns](https://msdn.microsoft.com/library/azure/1ec722fa-b623-4e26-a44e-a50c6d726223/) module. In the **Properties** pane, select **Remove entire row** under **Cleaning mode** to clean the data by removing rows that have missing values. Double-click the module and type the comment "Remove missing value rows."



1. Run the experiment by clicking **RUN** under the experiment canvas.

When the experiment finishes, all the modules will have a green check mark to indicate that they completed successfully. Notice also the **Finished running** status in the upper-right corner.

All we have done in the experiment to this point is clean the data. If you want to view the cleaned dataset, click the left output port of the [Clean Missing Data](https://msdn.microsoft.com/library/azure/d2c5ca2f-7323-41a3-9b7e-da917c99f0c4/) module ("Cleaned dataset") and select **Visualize**. Notice that the **normalized-losses** column is no longer included, and there are no missing values.

Now that the data is clean, we're ready to specify what features we're going to use in the predictive model.

## Step 3: Define features

In machine learning, *features* are individual measurable properties of something you’re interested in. In our dataset, each row represents one automobile, and each column is a feature of that automobile. Finding a good set of features for creating a predictive model requires experimentation and knowledge about the problem you want to solve. Some features are better for predicting the target than others. Also, some features have a strong correlation with other features (for example, city-mpg versus highway-mpg), so they will not add much new information to the model and they can be removed.

Let's build a model that uses a subset of the features in our dataset. You can come back and select different features, run the experiment again, and see if you get better results. As a first guess, we'll select the following features (columns) with the [Project Columns](https://msdn.microsoft.com/library/azure/1ec722fa-b623-4e26-a44e-a50c6d726223/) module. Note that for training the model, we need to include the *price* value that we're going to predict.

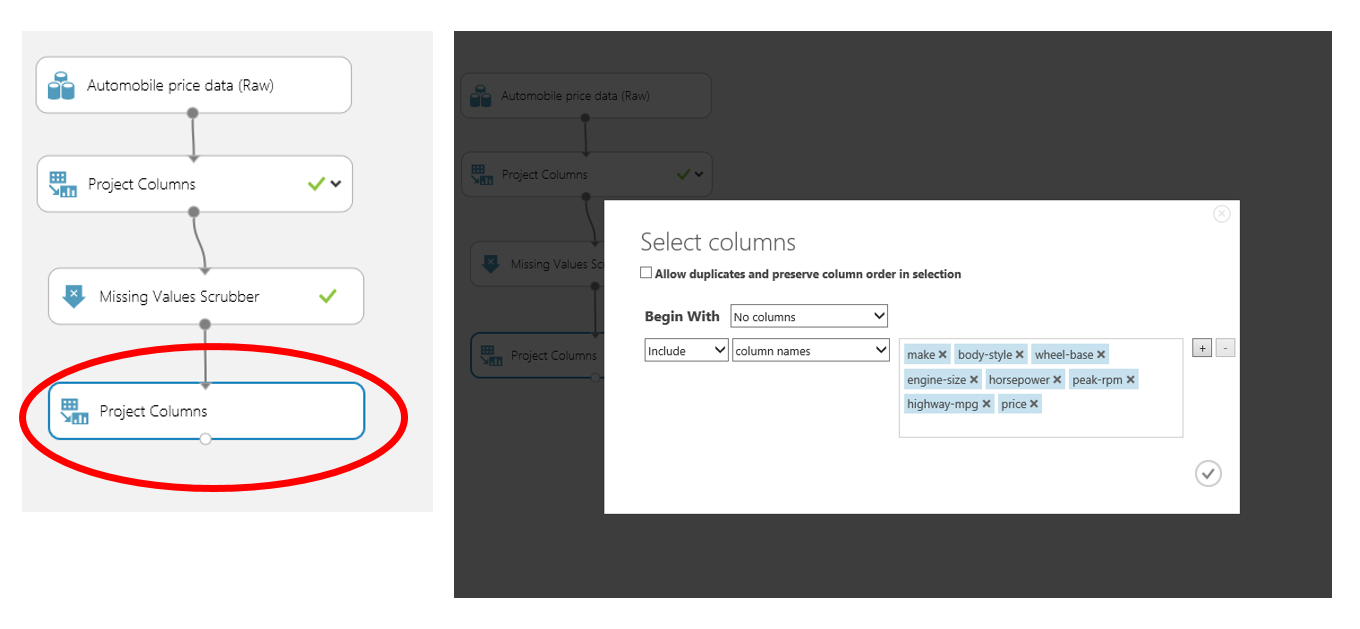
make, body-style, wheel-base, engine-size, horsepower, peak-rpm, highway-mpg, price

1. Drag another [Project Columns](https://msdn.microsoft.com/library/azure/1ec722fa-b623-4e26-a44e-a50c6d726223/) module to the experiment canvas and connect it to the left output port of the [Clean Missing Data](https://msdn.microsoft.com/library/azure/d2c5ca2f-7323-41a3-9b7e-da917c99f0c4/) module. Double-click the module and type "Select features for prediction."
2. Click **Launch column selector** in the **Properties** pane.
3. In the column selector, select **No columns** for **Begin With**, and then select **Include** and **column names** in the filter row. Enter our list of column names. This directs the module to pass through only columns that we specify.

TIP:

Because we've run the experiment, the column definitions for our data have passed from the original dataset through the [Clean Missing Data](https://msdn.microsoft.com/library/azure/d2c5ca2f-7323-41a3-9b7e-da917c99f0c4/) module. When you connect [Project Columns](https://msdn.microsoft.com/library/azure/1ec722fa-b623-4e26-a44e-a50c6d726223/) to [Clean Missing Data](https://msdn.microsoft.com/library/azure/d2c5ca2f-7323-41a3-9b7e-da917c99f0c4/), the [Project Columns](https://msdn.microsoft.com/library/azure/1ec722fa-b623-4e26-a44e-a50c6d726223/) module becomes aware of the column definitions in our data. When you click the **column names** box, a list of columns is displayed, and you can select the columns that you want to add to the list.

1. Click the check mark (OK) button.



This produces the dataset that will be used in the learning algorithm in the next steps. Later, you can return and try again with a different selection of features.

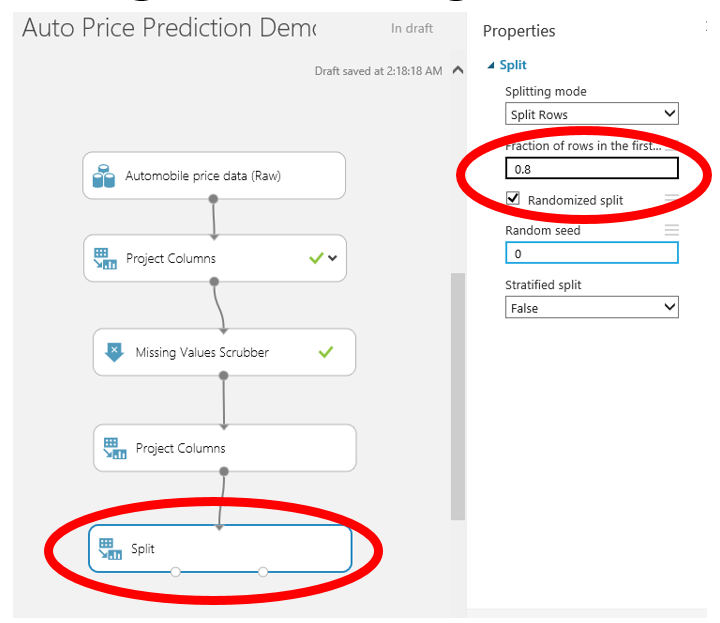
## Step 4: Choose and apply a learning algorithm

Now that the data is ready, constructing a predictive model consists of training and testing. We'll use our data to train the model and then test the model to see how close it's able to predict prices.

*Classification* and *regression* are two types of supervised machine learning techniques. Classification is used to make a prediction from a defined set of values, such as a color (red, blue, or green). Regression is used to make a prediction from a continuous set of values, such as a person's age.

We want to predict the price of an automobile, which can be any value, so we'll use a regression model. For this example, we'll train a simple *linear regression* model, and in the next step, we'll test it.

1. We can use our data for both training and testing by splitting it into separate training and testing sets. Select and drag the [Split](https://msdn.microsoft.com/library/azure/70530644-c97a-4ab6-85f7-88bf30a8be5f/) module to the experiment canvas and connect it to the output of the last [Project Columns](https://msdn.microsoft.com/library/azure/1ec722fa-b623-4e26-a44e-a50c6d726223/) module. Set **Fraction of rows in the first output dataset** to 0.8. This way, we'll use 80% of the data to train the model, and hold back 20% for testing.

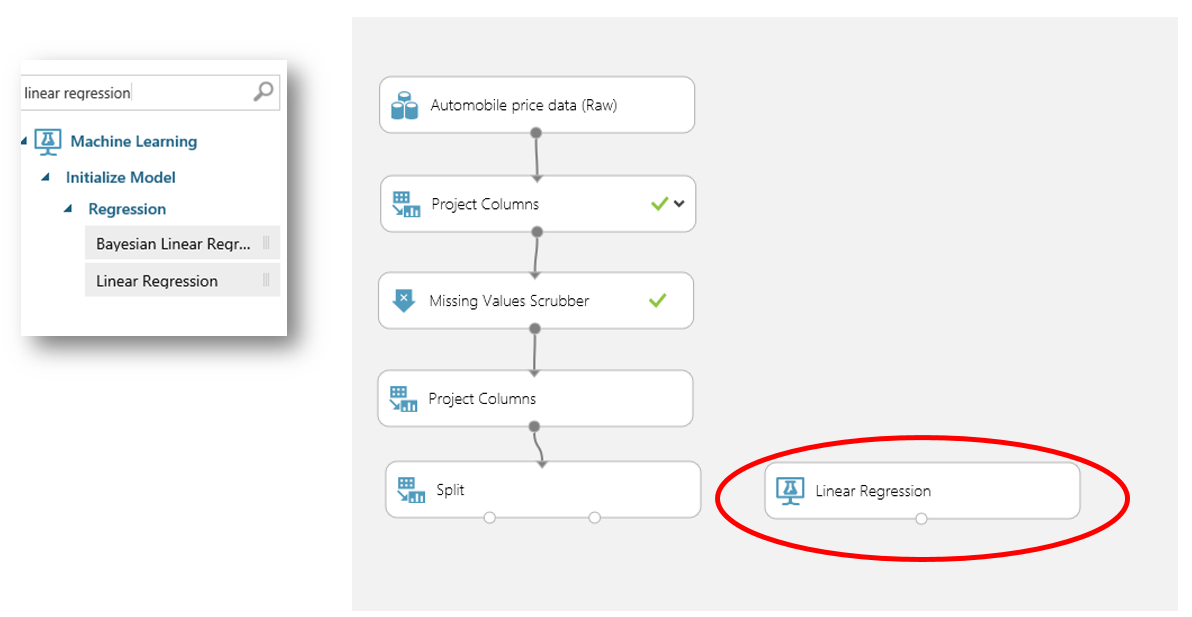


TIP:

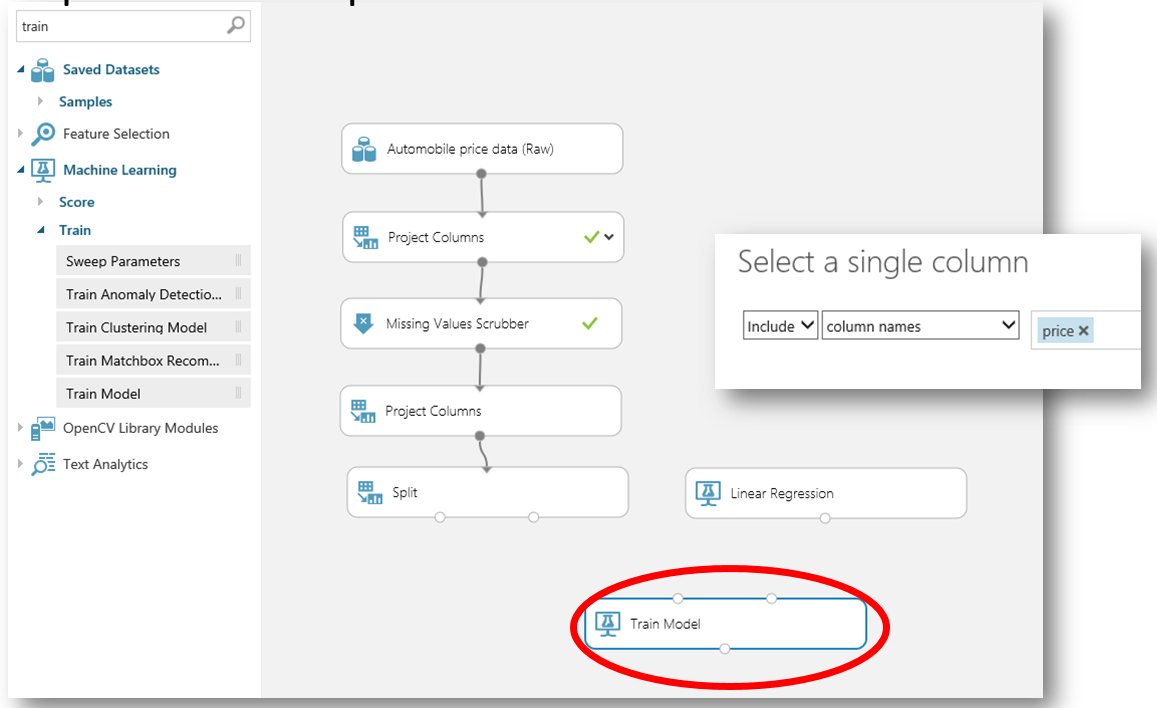
By changing the **Random seed** parameter, you can produce different random samples for training and testing. This parameter controls the seeding of the pseudo-random number generator.

1. Run the experiment. This allows the [Project Columns](https://msdn.microsoft.com/library/azure/1ec722fa-b623-4e26-a44e-a50c6d726223/) and [Split](https://msdn.microsoft.com/library/azure/70530644-c97a-4ab6-85f7-88bf30a8be5f/) modules to pass column definitions to the modules we'll be adding next.
2. To select the learning algorithm, expand the **Machine Learning** category in the module palette to the left of the canvas, and then expand **Initialize Model**. This displays several categories of modules that can be used to initialize machine learning algorithms.

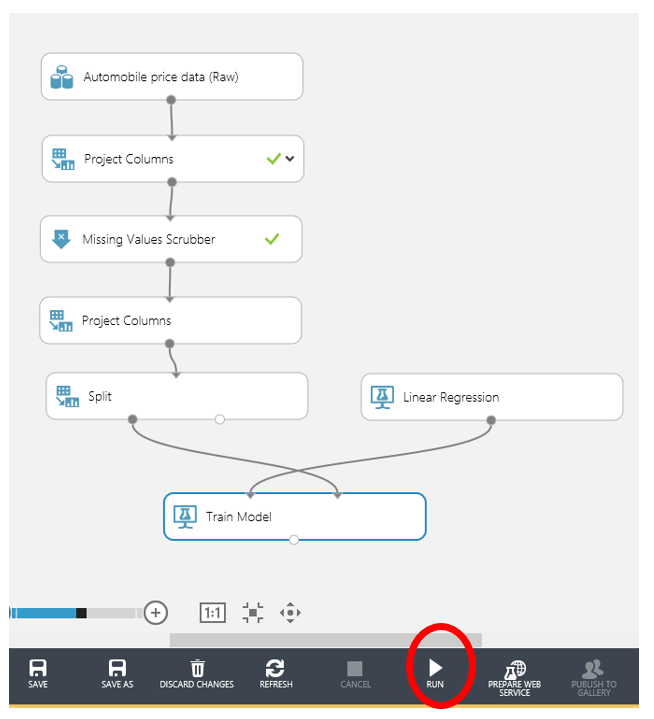
For this experiment, select the [Linear Regression](https://msdn.microsoft.com/library/azure/31960a6f-789b-4cf7-88d6-2e1152c0bd1a/) module under the **Regression** category (you can also find the module by typing "linear regression" in the palette Search box), and drag it to the experiment canvas.



1. Find and drag the [Train Model](https://msdn.microsoft.com/library/azure/5cc7053e-aa30-450d-96c0-dae4be720977/) module to the experiment. Select the [Train Model](https://msdn.microsoft.com/library/azure/5cc7053e-aa30-450d-96c0-dae4be720977/) module, click **Launch column selector** in the **Properties** pane, and select the **price** column. This is the value that our model is going to predict.
2. Connect the left input port to the output of the [Linear Regression](https://msdn.microsoft.com/library/azure/31960a6f-789b-4cf7-88d6-2e1152c0bd1a/) module. Connect the right input port to the training data output (left port) of the [Split](https://msdn.microsoft.com/library/azure/70530644-c97a-4ab6-85f7-88bf30a8be5f/) module.



1. Run the experiment.

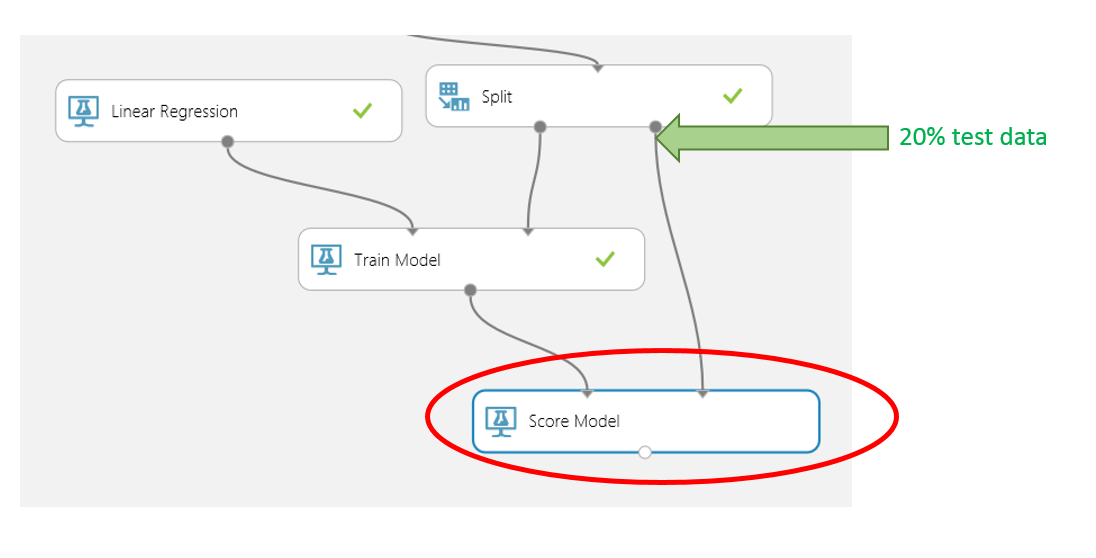


The result is a trained regression model that can be used to score new samples to make predictions.

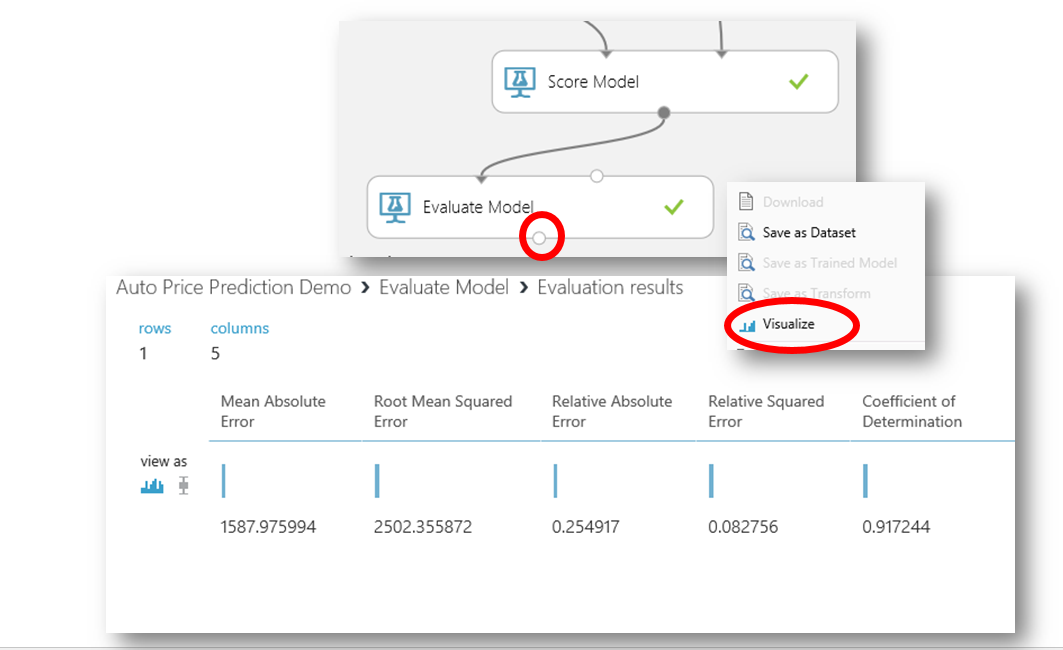
## Step 5: Predict new automobile prices

Now that we've trained the model using 80% of our data, we can use it to score the other 20% of the data to see how well our model functions.

1. Find and drag the [Score Model](https://msdn.microsoft.com/library/azure/401b4f92-e724-4d5a-be81-d5b0ff9bdb33/) module to the experiment canvas and connect the left input port to the output of the [Train Model](https://msdn.microsoft.com/library/azure/5cc7053e-aa30-450d-96c0-dae4be720977/) module. Connect the right input port to the test data output (right port) of the [Split](https://msdn.microsoft.com/library/azure/70530644-c97a-4ab6-85f7-88bf30a8be5f/) module.
2. To run the experiment and view the output from the [Score Model](https://msdn.microsoft.com/library/azure/401b4f92-e724-4d5a-be81-d5b0ff9bdb33/) module, click the output port and select **Visualize**. The output shows the predicted values for price and the known values from the test data.



1. Finally, to test the quality of the results, select and drag the [Evaluate Model](https://msdn.microsoft.com/library/azure/927d65ac-3b50-4694-9903-20f6c1672089/) module to the experiment canvas, and connect the left input port to the output of the [Score Model](https://msdn.microsoft.com/library/azure/401b4f92-e724-4d5a-be81-d5b0ff9bdb33/) module. (There are two input ports because the [Evaluate Model](https://msdn.microsoft.com/library/azure/927d65ac-3b50-4694-9903-20f6c1672089/) module can be used to compare two models.)



1. Run the experiment.

To view the output from the [Evaluate Model](https://msdn.microsoft.com/library/azure/927d65ac-3b50-4694-9903-20f6c1672089/) module, click the output port and select **Visualize**. The following statistics are shown for our model:

* **Mean Absolute Error** (MAE): The average of absolute errors (an *error* is the difference between the predicted value and the actual value).
* **Root Mean Squared Error** (RMSE): The square root of the average of squared errors of predictions made on the test dataset.
* **Relative Absolute Error**: The average of absolute errors relative to the absolute difference between actual values and the average of all actual values.
* **Relative Squared Error**: The average of squared errors relative to the squared difference between the actual values and the average of all actual values.
* **Coefficient of Determination**: Also known as the **R squared value**, this is a statistical metric indicating how well a model fits the data.

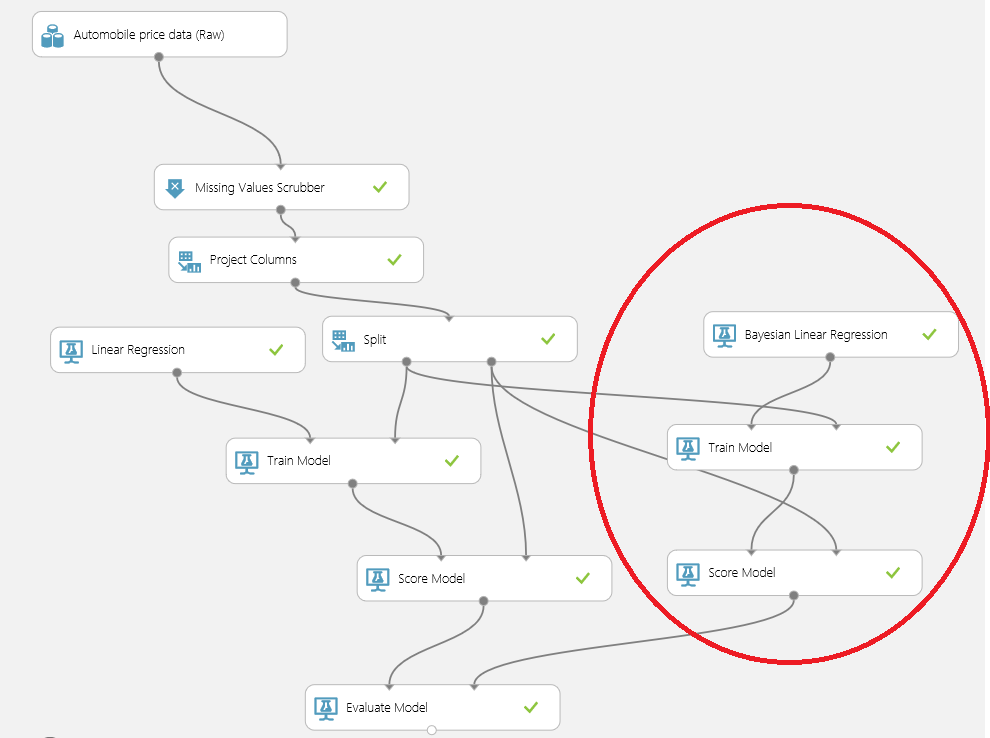
For each of the error statistics, smaller is better. A smaller value indicates that the predictions more closely match the actual values. For **Coefficient of Determination**, the closer its value is to one (1.0), the better the predictions.

# Additional Learning

### Compare two models

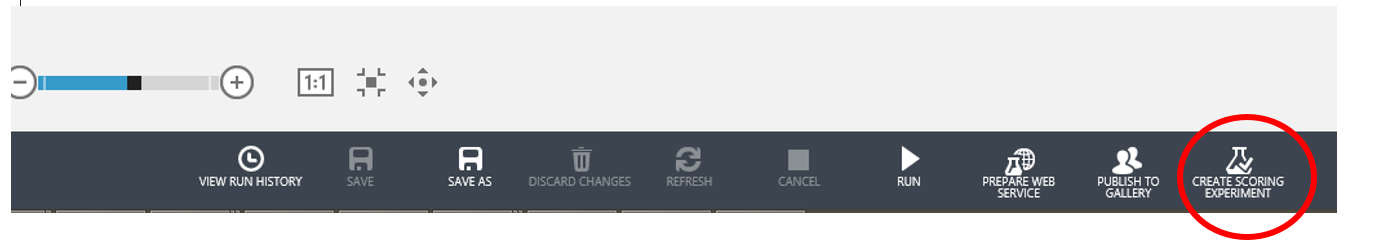
The Evaluate module is very effective in comparing the metrics between two different algorithms.

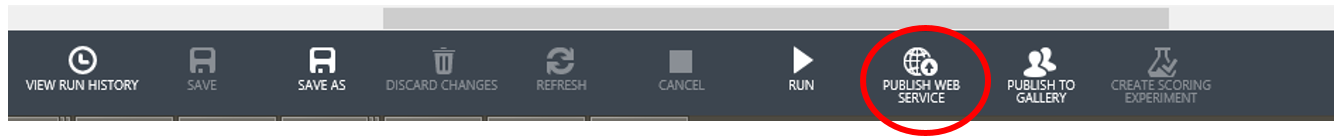
Compare the results of the linear regression based model with a different model. Which model performs better?

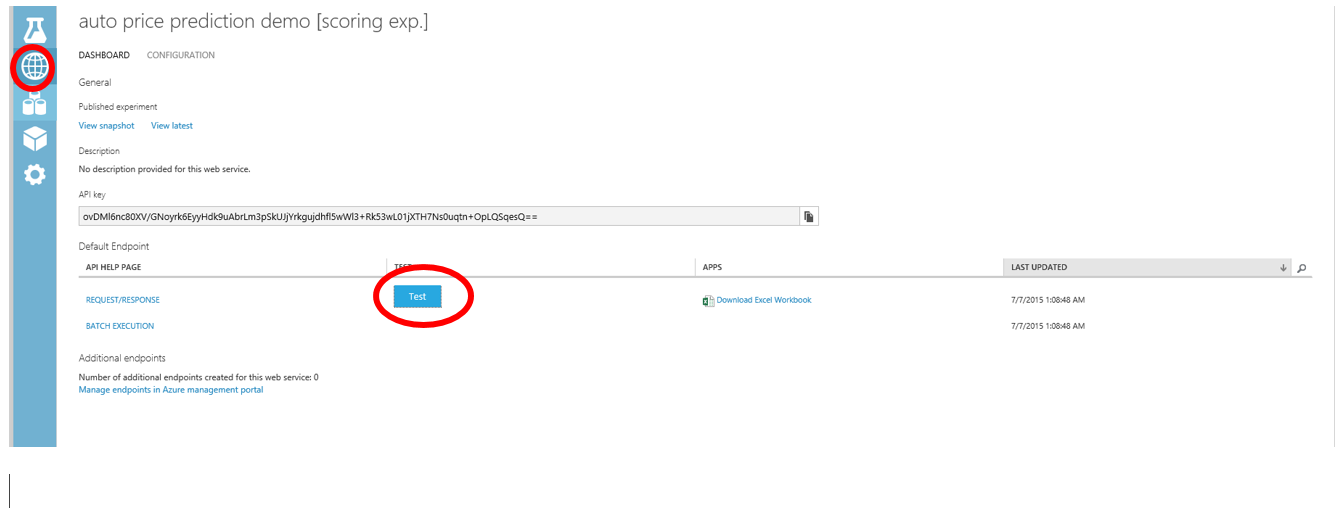


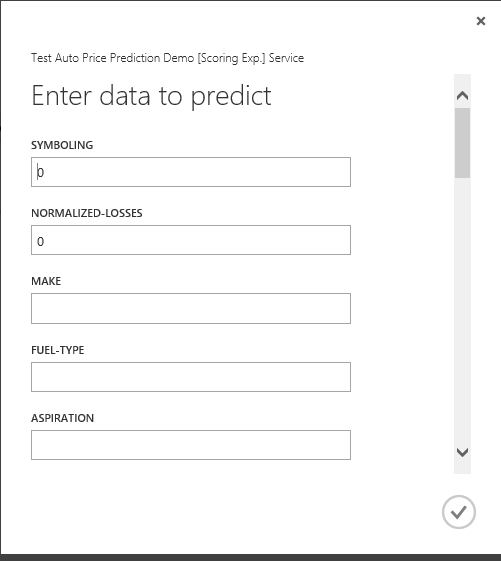
### Publish the model as a web service

Convert the experiment to a Scoring Experiment and publish the model as a web service.





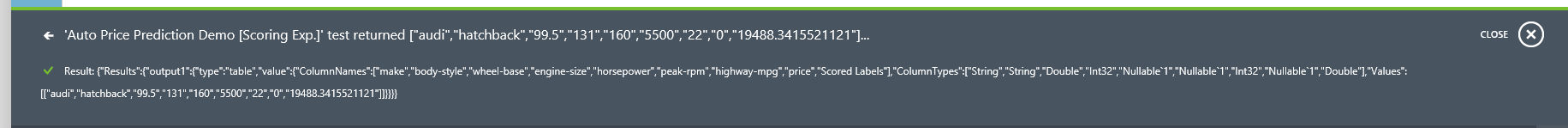




Enter Values –

* 1. audi gas turbo two hatchback 4wd front 99.5 178.2 67.9 52 3053 ohc five 131 mpfi 3.13 3.4 7 160 5500 16 22

Prediction Result -



Errata-

Some screenshots in this lab incorrectly refer to the “Clean Missing Data” module as the “Missing Values Scrubber” – which is a deprecated data cleaning module.