**Lab for Chapter 3:**

**Regression Techniques Part I: Linear Regression**

Contents

[Exercise 1: Create a Model **4**](#_bookmark0)

[Exercise 2: Train the Model **9**](#_bookmark1)

[Exercise 3: Test the Model **11**](#_bookmark2)

**Exercise 0: Overview**

Identify what kind of machine learning problem this is. **“Predict the price of an automobile”**

This exercise will create a Regression model to predict automobile price.

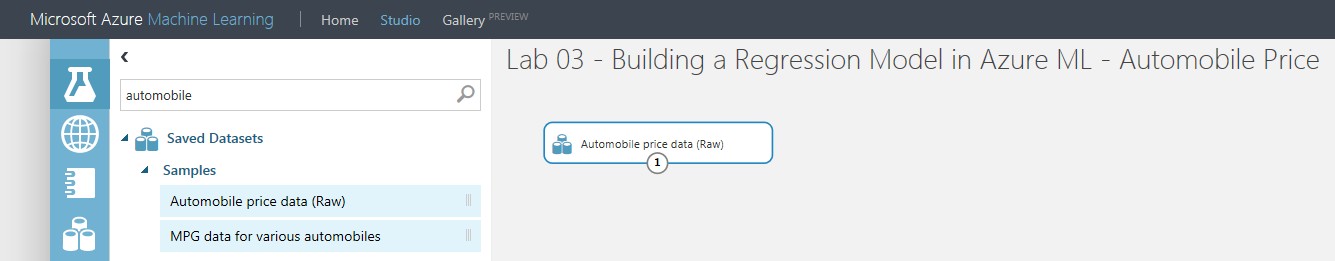
The simplest process to create any predictive model is **Create a Model**

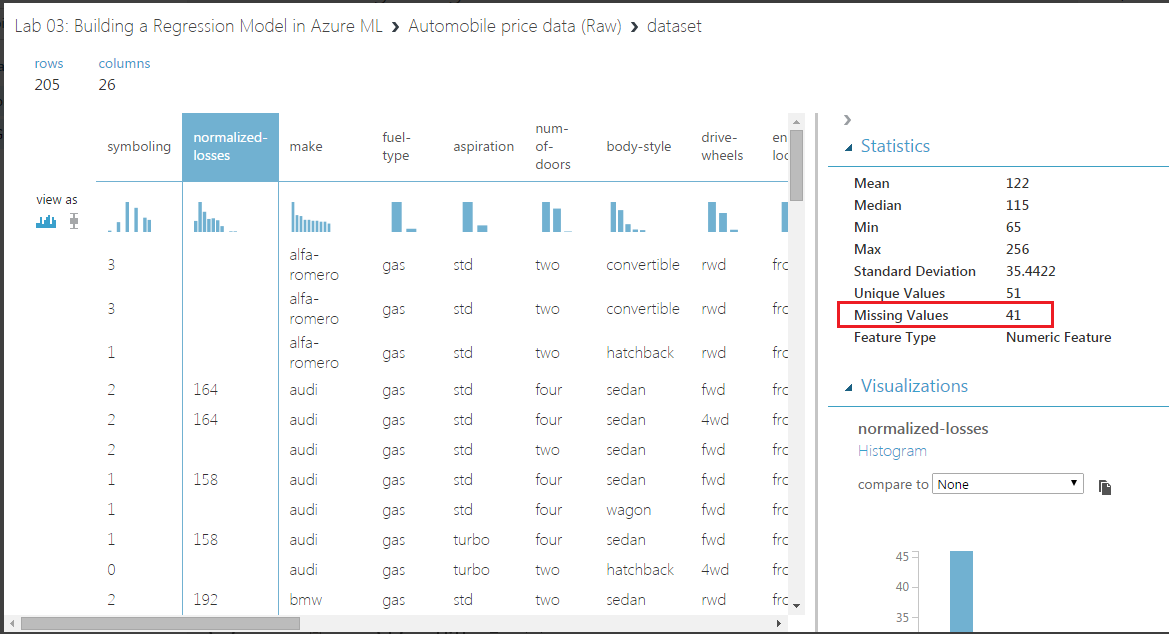
1. Get data
2. Preprocess data
3. Define features   
     
   **Train the Model**
4. Choose and apply a Machine Learning algorithm   
     
   **Test the Model**
5. Predict over new data

**Exercise 1: Create Model**

Step 1: Get Data

* 1. Create a new experiment and rename appropriately (Lab 04 - Building a Regression Model in Azure ML – Automobile Price).
  2. **Automobile price (raw)** data will be used for this exercise and this is already loaded in Azure ML by default.
  3. Search for it and drag the dataset into the experiment.



* 1. By clicking the output port of the dataset, you can select Visualize, which will allow you to explore the data and understand the key statistics of each of the columns.
  2. Click on column ‘normalized-losses’ and under Statistics look out for "Missing Values”

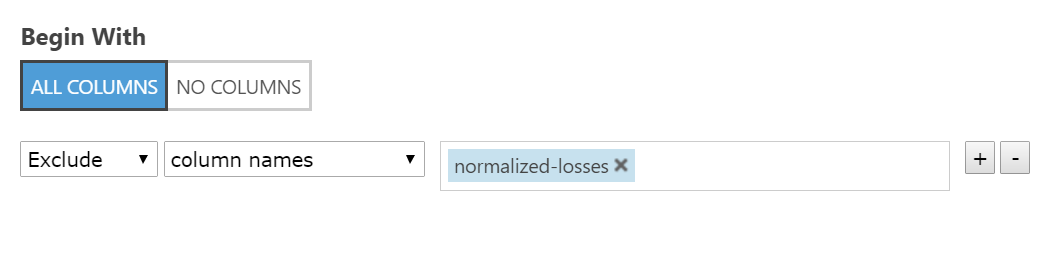
**Step 2: Preprocess Data**

In many cases raw data needs to be processed before being sent as input to train a model. **Missing values** need to be cleaned and unwanted columns have to be removed.

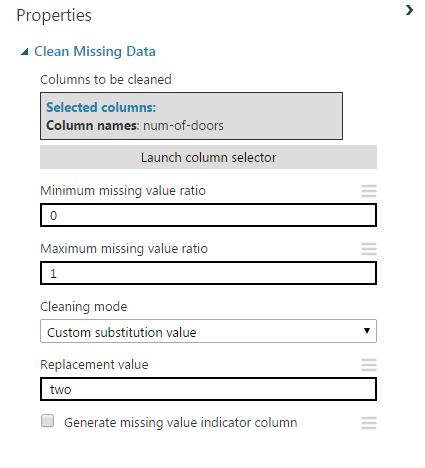
*As a rule of thumb, if a variable has 40% or more missing values, it can be dropped from the analysis, unless it is business critical.*

Column ‘normalized-losses’ is having too many missing values, so we can ignore/delete it.

1. Drag the **“Select Columns”** module, and connect it to the output port of the “Automobile price data (Raw)” dataset. This module allows you to select which columns of data you want to include or exclude in the model.
2. Select the ‘Select Columns’ module and click ‘Launch column selector’ in the properties pane (i.e. the right pane).
3. Make sure ‘All columns’ is selected in the filter dropdown called ‘Begin With’. This directs Select Columns to pass all columns through (except for the ones you are about to exclude).
4. In the next row, select ‘Exclude’ and ‘column names’, and select “**normalized-losses**”. With this selection, all columns will pass through, except for the column ‘normalized-losses’.
5. Click the check mark OK button to close the column selector.



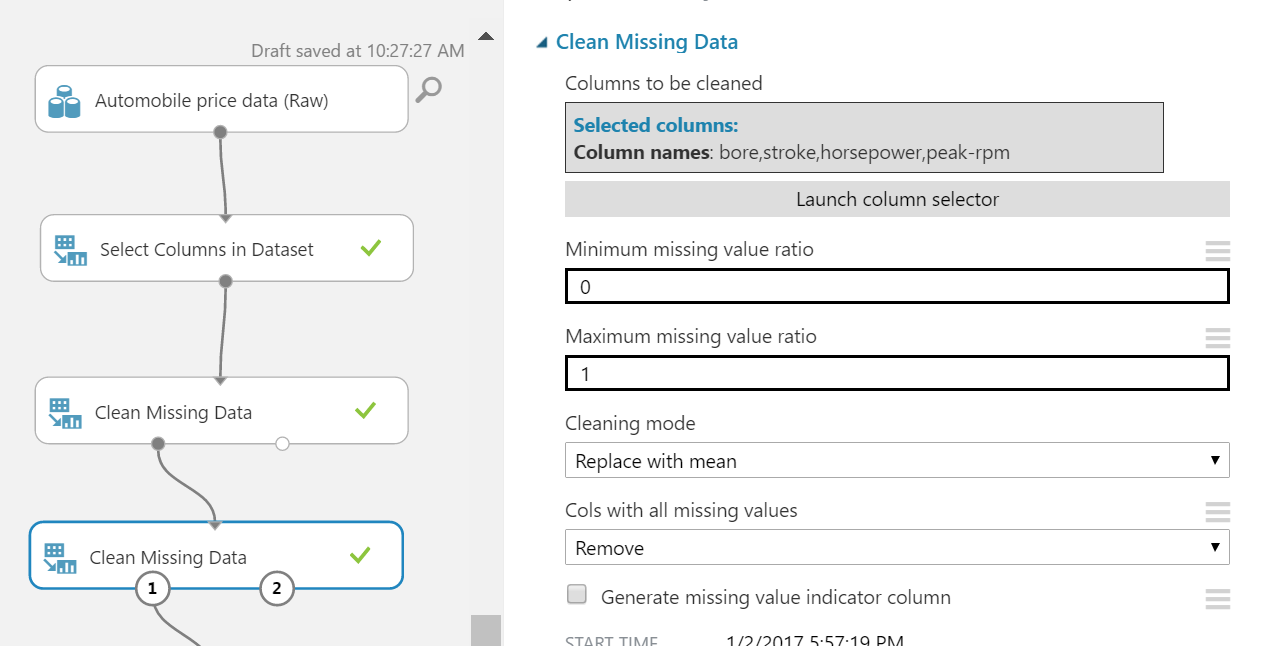
1. Label the ‘Select Columns’ module for good style.
2. Drag the “**Clean Missing Data**” module to the experiment canvas and connect it with the output of Select Columns module.
3. Clean Missing Data module replaces the missing value with custom or other calculated value.
4. Click ‘Launch column selector’ in the properties pane, Make sure ‘No columns’ is selected in the filter dropdown called ‘Begin With’.
5. In the next row, select ‘Include’ and ‘column names’, and select “**num-of-doors**”.
6. Click the check mark OK button to close the column selector.
7. In the properties pane of “Clean Missing Data” module, configure it to set the ‘Replacement value’ as “**two**”.



1. Drag another “**Clean Missing Data**” module to the experiment canvas and connect it to the previous “Clean Missing Data” module.
2. Now clean the **numerical** missing data with the same procedure as shown below.
3. Label both the Clean Missing Data modules for good style.

There are missing values in the following columns

* + bore
  + stroke
  + horsepower
  + peak-rpm



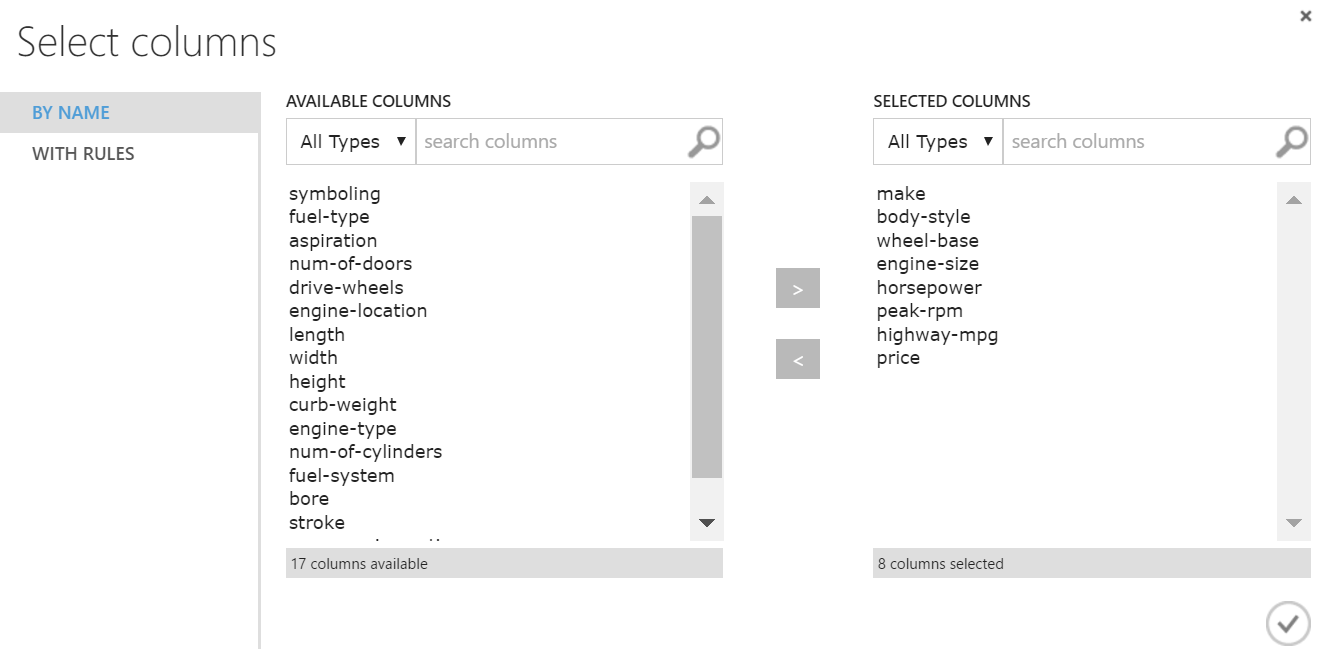
**Step 3: Define Features**

In machine learning, features are individual measurable properties created from the raw data to help the algorithms to learn the task at hand. Understanding the role played by each feature is super important. For example, some features are better at predicting the target than others. In addition, some features can have a strong correlation with other features (e.g. city-mpg vs. highway-mpg). Adding highly correlated features as inputs might not be useful, since they contain similar information.

For this exercise, you will build a predictive model that uses a subset of the features of the “Automobile price data (Raw)” dataset to predict the price for new automobiles.

Each row represents an automobile. Each column is a feature of that automobile. It is important to identify a good set of features that can be used to create the predictive model. Often, this requires experimentation and knowledge about the problem domain. For illustration purpose, you will use the Select Columns module to select the following features: **make, body-style, wheel-base, engine- size, horsepower, peak-rpm, highway-mpg and price**

1. Drag another “**Select Columns**” module to the experiment canvas. Connect it to the second “Clean Missing Data” module.
2. Click ‘Launch column selector’ in the properties pane.
3. In the column selector, select ‘No columns’ for ‘Begin With’, then select ‘Include’ and ‘column names’ in the filter row. Enter the following column names: **make, body-style, wheel-base, engine-size, horsepower, peak-rpm, highway-mpg, price**. This directs the module to pass through only these columns.
4. Click OK.



**Exercise 2: Train the Model**

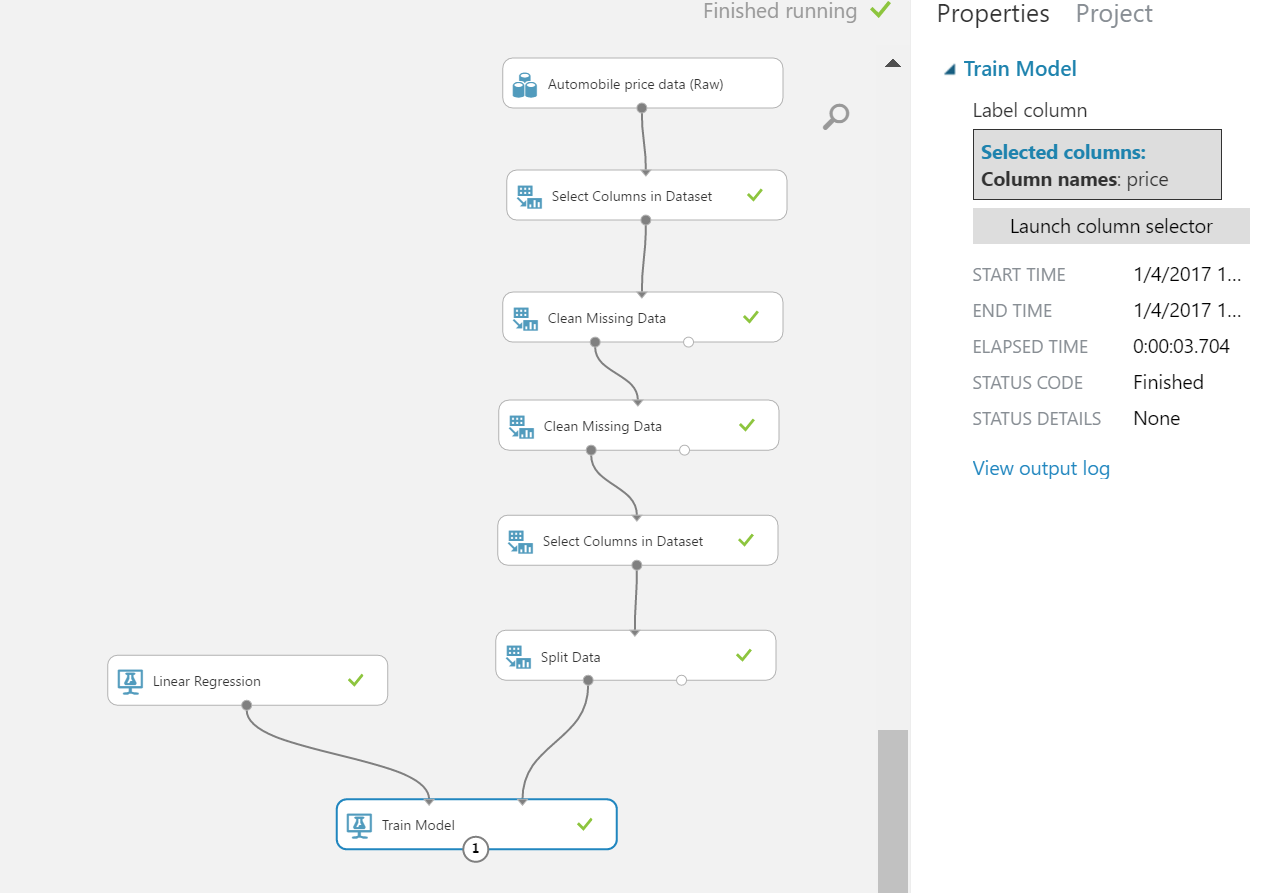
**Step 1: Choose and Apply Machine Learning Algorithms**

When constructing a predictive model, you first need to train the model and then validate that the model is effective. In this experiment, you will build a Regression model.

In this experiment, you will train a regression model and use it to predict the price of an automobile.

Specifically, you will train a simple **Linear Regression** model. After the model has been trained, you will use some of the modules available in Machine Learning Studio to validate the model.

1. Split the data into training and testing sets: Select and drag the “**Split**” module to the experiment canvas and connect it to the output of the last “Select Columns” module.
2. In the properties pane, select ‘Splitting mode’ to ‘Split Rows’. Next, set ‘Fraction of rows in the first output dataset’ to **0.8**. This way, you will use 80% of the data to train the model and hold back 20% for testing.
3. Run the experiment. This allows the Select Columns and Split modules to pass along column definitions to the modules you will be adding next.
4. 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 a learning algorithm.
5. For this example experiment, select the “**Linear Regression**” module under the ‘Regression’ category and drag it to the experiment canvas.
6. Also, find and drag the “**Train Model**” module to the experiment. Click ‘Launch column selector’ and select the ‘**price’** column. This is the feature that your model is going to predict.
7. Connect the output of the “Linear Regression” module to the left input port of the “Train Model” module.
8. Also, connect the training data output (i.e. the left port) of the “Split” module to the right input port of the “Train Model” module.
9. Run the experiment.

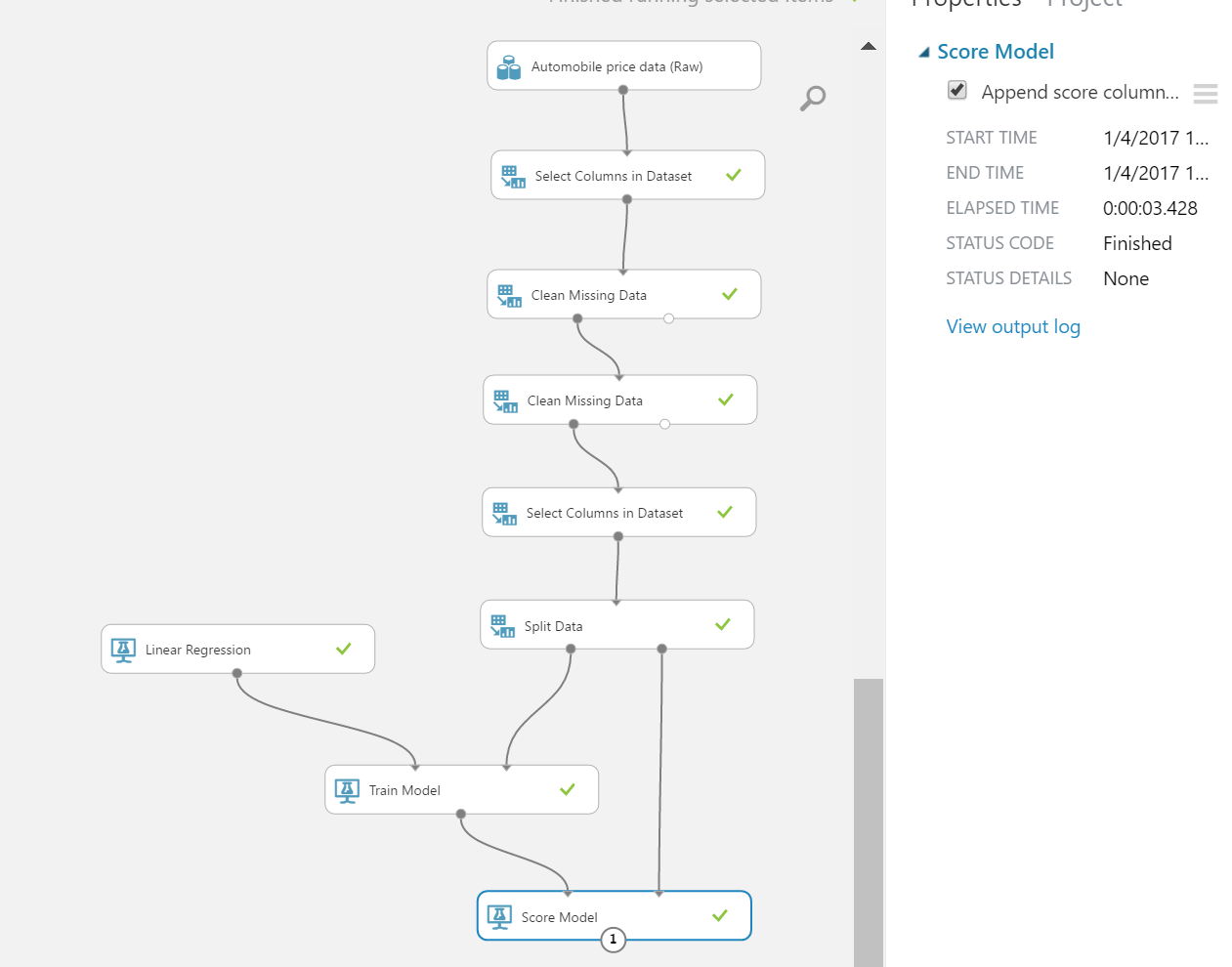


**Exercise 3: Test the Model**

**Step 1: Predict Over New Data**

Now that you’ve trained the model, you can use it to score the other 20% of your data and see how well your model predicts on unseen data.

1. Find and drag the “**Score Model**” module to the experiment canvas. Connect the output of the “Train Model” module to the left input port and test data output (right port) of the “Split” module to the right input port.



1. Run the experiment and view the output from the “Score Model” module (by clicking the output port and selecting Visualize). The output will show the predicted values for **price** along with the known values from the test data.

Remember, the standard output of any regression is **Scored Labels**.

