Exercise Part 4: Train a Machine Learning Model

Microsoft Azure Machine Learning includes an *automated machine learning*capability that leverages the scalability of cloud compute to automatically try multiple pre-processing techniques and model-training algorithms in parallel to find the best performing supervised machine learning model for your data.

Note

The automated machine learning capability in Azure Machine Learning supports *supervised* machine learning models - in other words, models for which the training data includes known label values. You can use automated machine learning to train models for:

- **Classification** (predicting categories or *classes*)
- **Regression** (predicting numeric values)
- **Time series forecasting** (regression with a time-series element, enabling you to predict numeric values at a future point in time)

Run an automated machine learning experiment

In Azure Machine Learning, operations that you run are called *experiments*. Follow the steps below to run an experiment that uses automated machine learning to train a regression model that predicts bicycle rentals.

1. In Azure Machine Learning studio, view the **Automated ML** page (under **Author**).

2. Create a new Automated ML run with the following settings:

Select dataset:

- **Dataset**: bike-rentals

Configure run:

- **New experiment name**: mslearn-bike-rental
- **Target column**: rentals (*this is the label the model will be trained to predict)*
- **Training compute target**: *the compute cluster you created previously*

Task type and settings:

- **Task type**: Regression *(the model will predict a numeric value)*

Additional configuration settings:

- **Primary metric**: Select **Normalized root mean square error** *(more about this metric later!)*
- **Explain best model**: Selected - *this option causes automated machine learning to calculate feature importance for the best model; making it possible to determine the influence of each feature on the predicted label.*
- **Blocked algorithms**: *Block**all** other than **RandomForest** and **LightGBM** - normally you'd want to try as many as possible, but doing so can take a long time!*

Exit criterion:

- **Training job time (hours)**: 0.25 - *this causes the experiment to end after a maximum of 15 minutes.*
- **Metric score threshold**: 0.08 - *this causes the experiment to end if a model achieves a normalized root mean square error metric score of 0.08 or less.*

Featurization settings:

- **Enable featurization**: Selected - *this causes Azure Machine Learning to automatically preprocess the features before training.*

3. When you finish submitting the automated ML run details, it will start automatically. Wait for the run status to change from *Preparing*to *Running*.

4. When the run status changes to *Running*, view the **Models** tab and observe as each possible combination of training algorithm and pre-processing steps is tried and the performance of the resulting model is evaluated. The page will automatically refresh periodically, but you can also select ↻ **Refresh**. It may take ten minutes or so before models start to appear, as the cluster nodes need to be initialized before training can begin.

5. Wait for the experiment to finish. It may take a while - now might be a good time for a coffee break!

Review the best model

After the experiment has finished; you can review the best performing model that was generated (note that in this case, we used exit criteria to stop the experiment - so the "best" model found by the experiment may not be the best possible model, just the best one found within the time allowed for this exercise!).

1. On the **Details** tab of the automated machine learning run, note the best model summary.

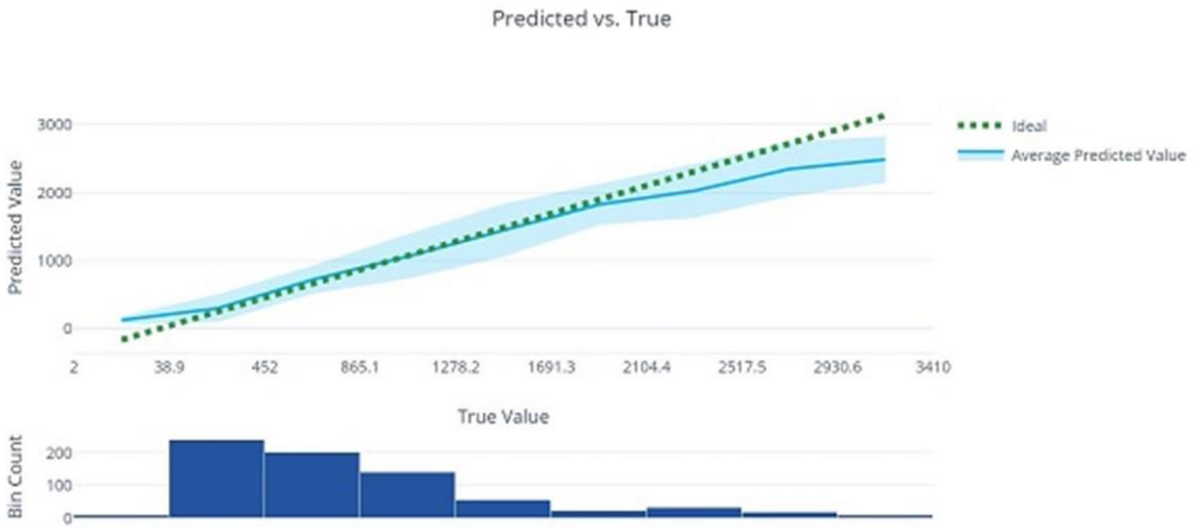
2. Select the **Algorithm name** for the best model to view its details. The best model is identified based on the evaluation metric you specified (*Normalized root mean square error*). To calculate this metric, the training process used some of the data to train the model, and applied a technique called *cross-validation*to iteratively test the trained model with data it wasn't trained with and compare the predicted value with the actual known value.

The difference between the predicted and actual value (known as the *residuals*) indicates the amount of *error*in the model, and this particular performance metric is calculated by squaring the errors across all of the test cases, finding the mean of these squares, and then taking the square root. What all of this means is that smaller this value is, the more accurately the model is predicting.

3. Next to the *Normalized root mean square error*value, select **View all other metrics** to see values of other possible evaluation metrics for a regression model.

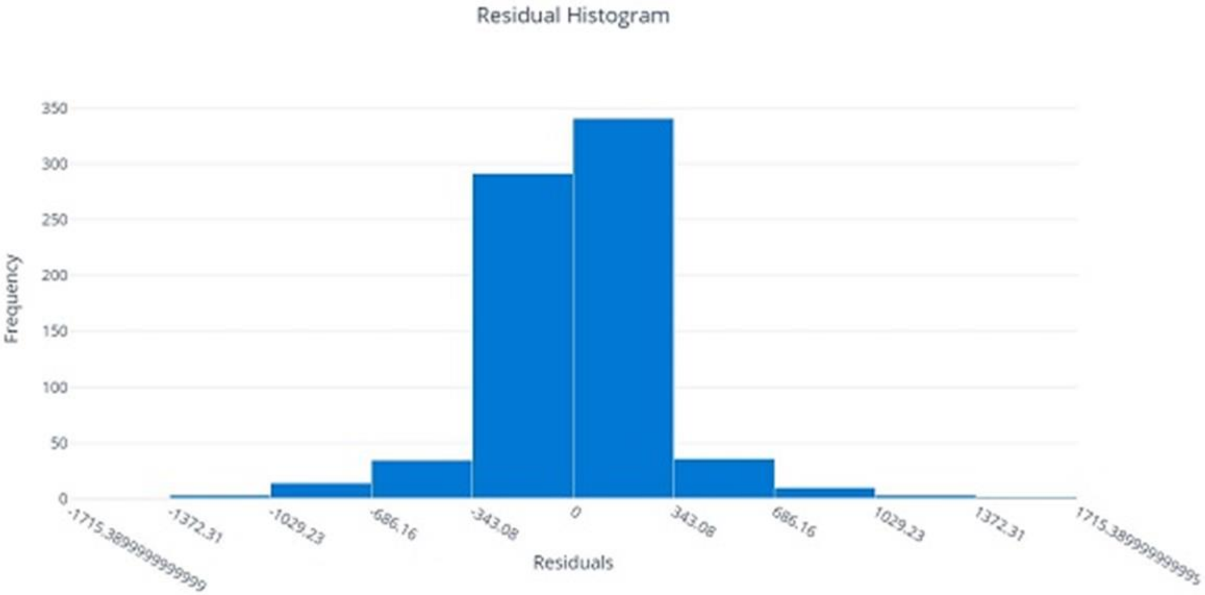
4. Select the **Metrics** tab and select the **residuals** and **predicted_true** charts if they are not already selected. Then review the charts, which show the performance of the model by comparing the predicted values against the true values, and by showing the *residuals* (differences between predicted and actual values) as a histogram.

The **Predicted vs. True** chart should show a diagonal trend in which the predicted value correlates closely to the true value. A dotted line shows how a perfect model should perform, and the closer the line for your model's average predicted value is to this, the better its performance. A histogram below the line chart shows the distribution of true values.



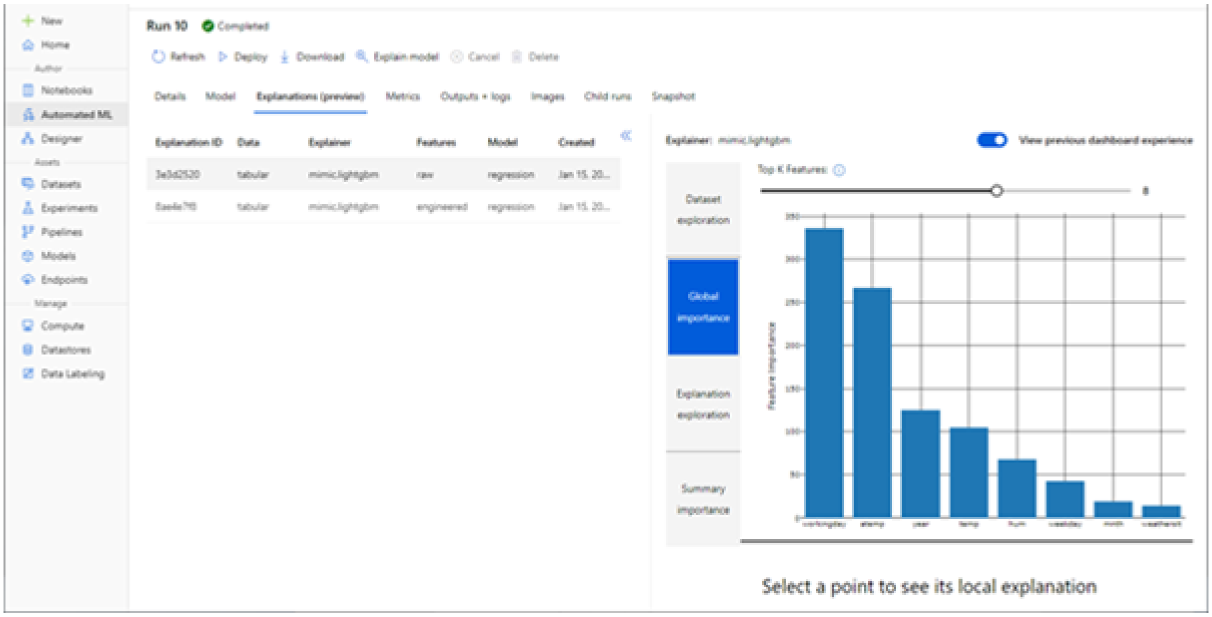
Column and line chart showing Predicted values versus True values

The **Residual Histogram** shows the frequency of residual value ranges. Residuals represent variance between predicted and true values that can't be explained by the model - in other words, errors; so what you should hope to see is that the most frequently occurring residual values are clustered around 0 (in other words, most of the errors are small), with fewer errors at the extreme ends of the scale.



Histogram showing Residuals on the X-Axis and Frequency on the Y-Axis

5. Select the **Explanations** tab. Click on the arrows >> next to **Explanation ID** to expand the explanations list. Select an explanation ID, select **View previous dashboard experience** on the right-hand side. Then select **Global Importance**. This chart shows how much each feature in the dataset influences the label prediction, like this:



Explanation ID	Data	Explainer	Features	Model	Created
3e3d2520	tabular	mimic.lightgbm	raw	regression	Jan 15, 20...
8ae4e7f8	tabular	mimic.lightgbm	engineered	regression	Jan 15, 20...

Explainer: mimic.lightgbm

Select a point to see its local explanation