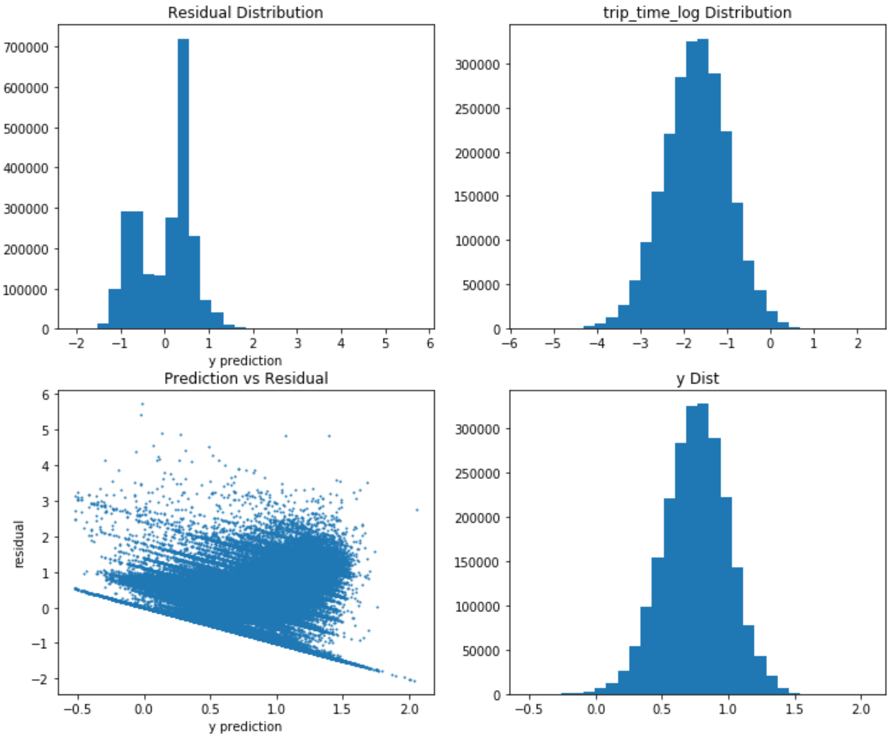
Capstone Project 1 Machine Learning

First let’s take some final steps to prep our data for our machine learning process. From our previous data exploration we know that many of our variables are right skew. In particular, the trip time, trip distance, fare amount, and average speed are all right skew. Let’s perform a log transform on these variables to make them more normal.

The next step to prepare our data is to find a way to handle our date objects. We have the various date attributes in their individual columns (ie month, date, hour, weekday), but they are represented in a linear fashion. For example hour 7 is before hour 9, and hour 13 is before hour 23. This doesn’t account for the cyclical nature of time and says that hour 15 is closer to hour 23 than hour 1 is. To fix this we can represent each datetime attribute as a combination of sin and cos coordinates. For example, we take hour 15 and represent it as a combinations of sin(2*pi(15/24)) and cos(2*pi(15/24)). We can do this for all of the datetime attributes so that their cyclical nature can be accounted for.

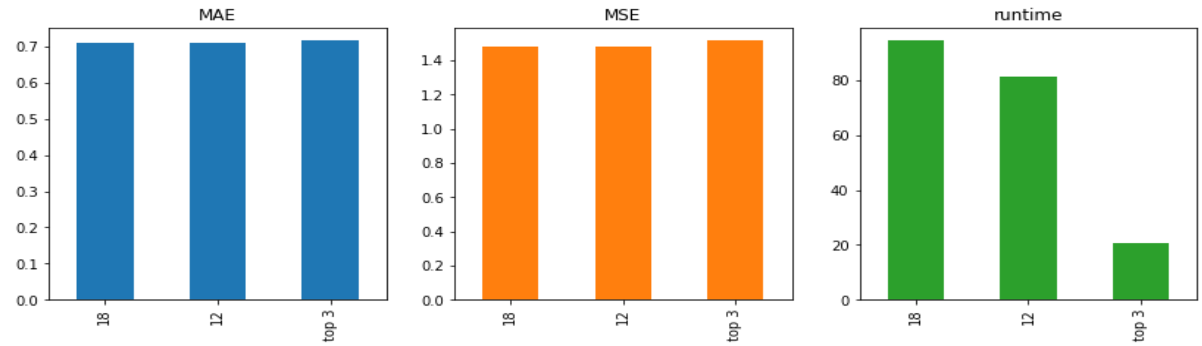
FIrst we try out a very simple linear regression model using only 1 feature variable, the distribution of predicted values mirrors that of the feature values and appears to be normal. We see that the residuals distribution appears to be bimodal, and the predicted y vs residuals plot shows some serious heteroscedasticity. Since the model was already performed in the log of the tip amount, let’s try and perform the linear regression on the data but excluding samples with a tip amount of 0 (which we suspect may be causing the line at the bottom of the plot)



In an effort to try and reduce the heteroscedasticity, let's try a few more simple linear regression models trained on a data set without samples with a tip amount of 0. A linear regression model using the trip time and the average speed are created, but unfortunately did not solve the problem with our residuals. This may suggest that we may be missing some important variables or factors in order to create a valid linear regression model. Let's try using a random forest regressor.

To start out, let's create some simple random forest regression models using a single feature variable. We can look at each model's mean absolute error and mean squared error numbers, and use these metrics to select our model. From our preliminary trials, the one that performed the best is the one that uses all of our available feature variables. This is very time and resource consuming however. Let's see if we can streamline and improve it.

From the model that uses all of our available feature variables, we look at our feature importance list and see that the fare amount has by far the highest importance, followed by the average speed and the trip time. Our original hypothesis that the weather would have an impact on the tip amounts proved incorrect. Using this list of feature importance, let's try some more random forest regressor models using different combinations of the top features. We try one using features with importance greater than 0, one with importance greater than 0.1, and one with importance greater than 0.3. Our 2nd model give us the lowest overall MAE. However, the model using only the top 3 important features had a significantly lower run time. Since the MAE doesn't differ by much, we chose to go with the model using the top 3 important features.



Next let's try and fine tune some of the parameters to see if we can improve upon our model. We can test out different estimator numbers, as well as different values for max_features using GridSearchCV using the neg_mean_absolute_error scoring metric. This tells us that our best parameters are n_estimators = 100 and max_features = log2.

Using our new best parameters, we now train the model again and see that our MAE has improved to 0.712 from 0.716, and our runtime is shorter. However our MSE has increased from 1.517 to 1.522. Since there doesn't seem to be an obvious reason why large errors should be penalized more, we should lean towards using MAE as our scoring metric and choose the model with the lowest MAE, which is our most recent one. Looking at our mean percentage errors, we can also see that this model frequently under estimates the tip amount by 43%. Another comparison we can do is to see if a model predicting the tip percentage will perform better than one predicting the tip amount. From this model we get a MAE of 0.05 and a MSE of 1.007. At first glance these seem like great error numbers. Once we take a closer look at the error percentages however, we can see that both models perform very similarly. Since both models are similar, let's choose to go with more straightforward one predicting the tip amount.

In conclusion, after attempting to fit a linear regression model, heteroscedasticity problems persisted and we decided to use a random forest regressor instead. After ranking the features by importance we decided to use the top 3 most important features. This gave us a model with a MAE similar to the others, but with a much quicker run time. Then using GridSearchCV for various parameters, we chose the set that gives us the best results. The end result is a random forest regressor that assuming a tip is given, predicts the tip amount based on the fare amount, trip time, and average speed of the taxi ride.