Capstone Project 1 - Predicting Taxi Tip Amounts

Proposal

Taxi cab drivers generally have to work long hours throughout the week and usually do so through a variety of weather conditions. It would be helpful to taxi drivers and potentially ride sharing companies like Uber and Lyft to figure out what factors affect the amount of tips a ride generates. We would like to explore the relationship between tip amounts given during a taxi ride and the inherent attributes of the ride. We would also like to explore the impact that the weather has on the amount of tips received. It does not seem outrageous that maybe customers tip more when the weather is nicer. Or maybe they tip more it's raining out of gratitude for the taxi driver stopping to pick them up. These are all relationships we would like to explore as we try to build a model to predict the amount of tips a taxi ride received.

We will be mainly working with data from the following sources:

2016 NYC Yellow Taxi Cab
https://data.cityofnewyork.us/Transportation/2016-Yellow-Taxi-Trip-Data/k67s-dv2t

2016 Hourly Weather in New York
https://www.kaggle.com/meinertsen/new-york-city-taxi-trip-hourly-weather-data/download

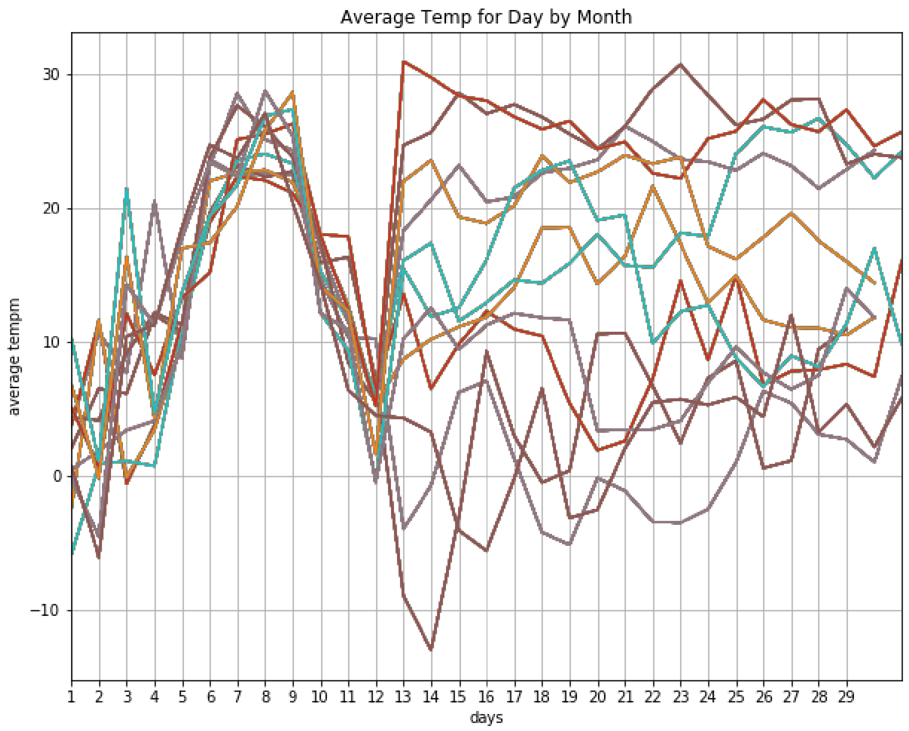
Weather Data Cleaning

The 2016 NYC weather data was originally acquired from the Wunderkind API, and provided here.
https://www.kaggle.com/meinertsen/new-york-city-taxi-trip-hourly-weather-data

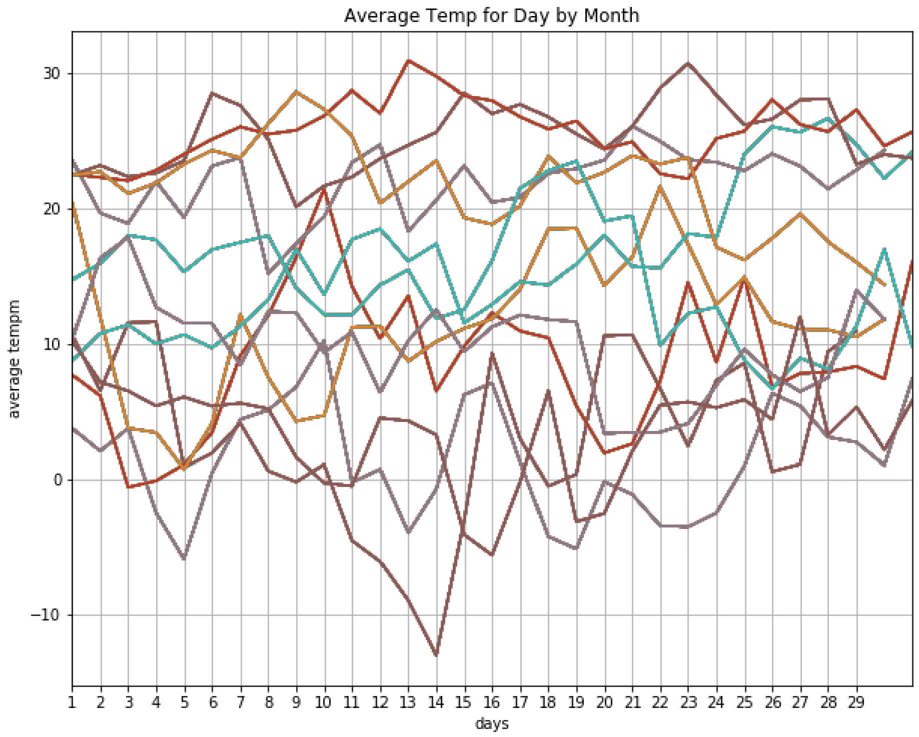
First we dropped all the columns that we do not think that we'll use.

As we check the data, we notice that it's fairly well organized. The rain and snow columns are are separate boolean columns. The data would be easier to read if it was a single column with a categorical values, so we combine the two columns into one precip_type column with values 'rain', 'snow', or an empty string for clear weather.

From examining the weather data, we notice something is off with the temperatures for certain days. Plotting the daily average temperatures for every month, we quickly notice that the for the days before the 12th of the month, the average temperatures rise and fall as you would expect throughout the year, and after the 12th of the month the average temperatures spread out and differ similar to how you expect them to differ between the various months. From this we can conclude that for the days that are before the 12, the month and day have been accidentally reversed in the data.



After correcting for this, we plot the average temperatures again and we see that it now somewhat behaves how we would expect.



Merging Weather and Taxi Data

Now that the dates in the weather data are correct, we can merge the weather and the taxi data on their pickup times using merge_asof, matching each taxi ride pickup time with its nearest weather recording.

Sampling

Since our data set is fairly large with over 60 million rows, due to hardware limitations we would need to take a sample to work with. With the goal of analyzing if the weather affects the tip amounts that taxi riders give, we should take a stratified sample of taxi rides in the rain, in the snow, and in clear weather. Using 10% of the original data would still give us 6 million samples, which seems reasonable.

Data Cleaning

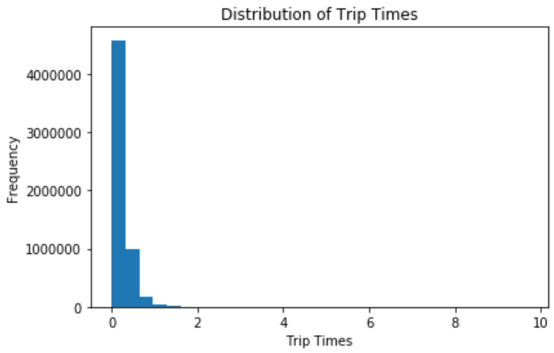
With a merged data set, now we can start to clean it up. The first thing we should do is calculate the time of each taxi trip using the pick up and drop off datetimes. Once we do this we no longer need the drop off pickup time columns, so we can drop that. Now with the trip time info, we can calculate the average speed of each trip using the time and the distance. Using the average speed, we can also eliminate speeds that are either very low or very high and that do not make sense. Lastly we should drop the rest of the columns that don't appear useful. The thunder, hail, and tornado columns have only 0 values so we should drop those. Now our data is ready to be explored.

Data Story

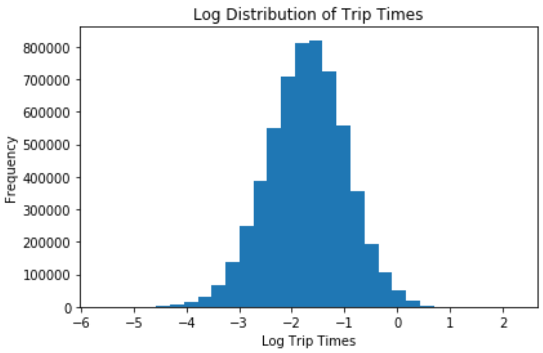
We would like to explore the relationships between various attributes of a taxi ride and the tip amount that it receives. This includes features like the trip time, trip distance, average speed, temperature, and visibility.

The first thing we do is create a correlation heat map to see if there are maybe some areas we should focus on and explore. From this heatmap we can see that there is a strong correlation between the trip time and the tip amount. It doesn't seem apparent that there is a strong relationship between any of the weather attributes and the tip amount though.

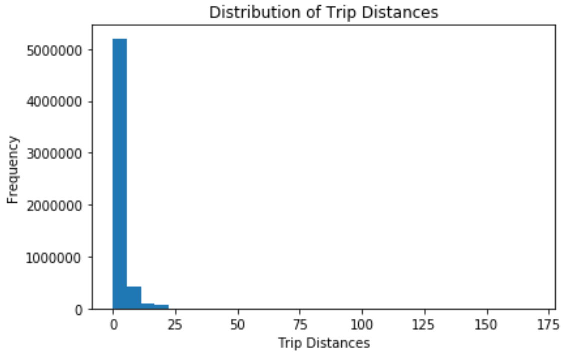
Looking at the trip times and distances, we can see that they are both right skewed in their distributions. Using a log transformation makes them look close to normal.



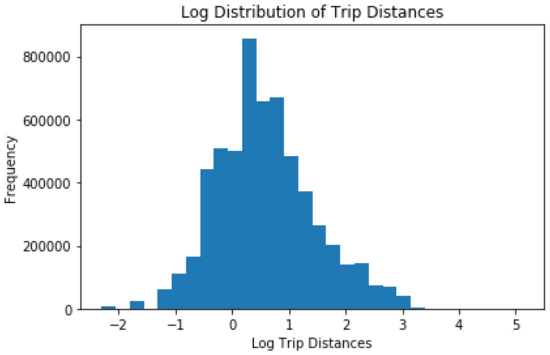
Distribution of Trip Times



Log Distribution of Trip Times

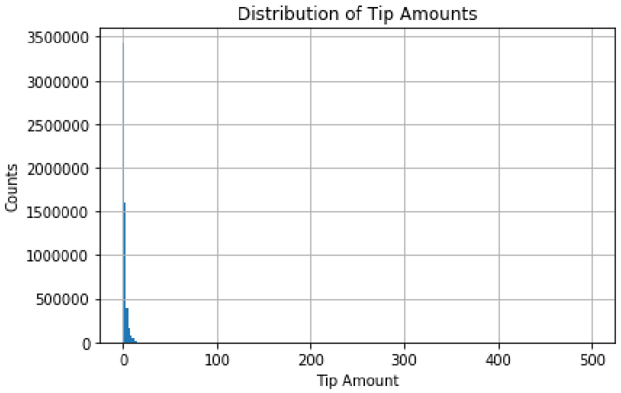


Distribution of Trip Distances

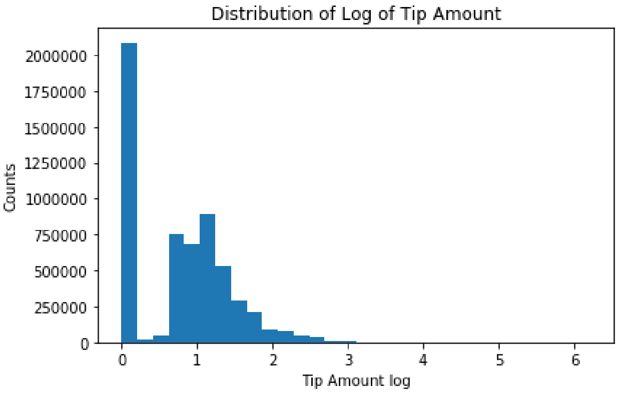


Log Distribution of Trip Distances

The trip amounts appear to be bimodal, with a large spike at 0 and a right skewed distributions. This intuitively makes sense since there would be a lot of trips with 0 tips, and the trips that do tip would have a range of amounts.

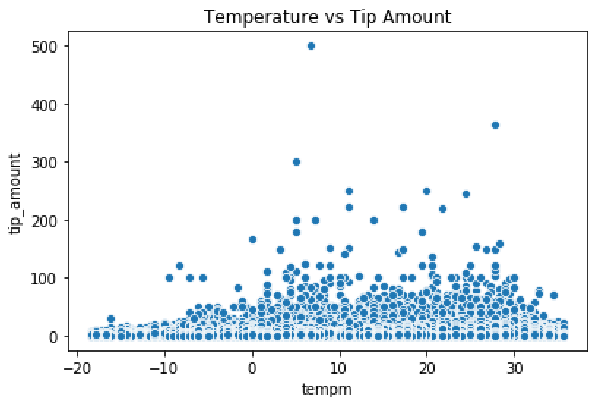


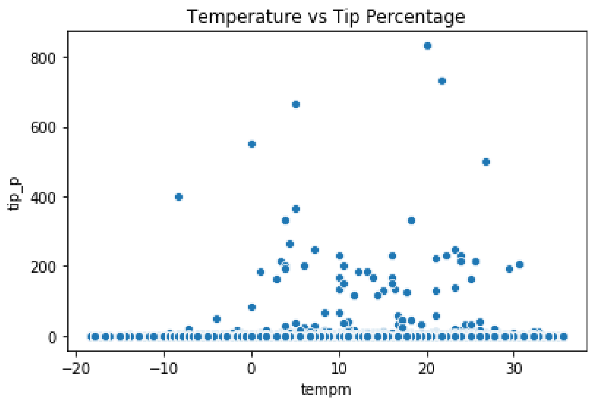
Distribution of Tip Amounts

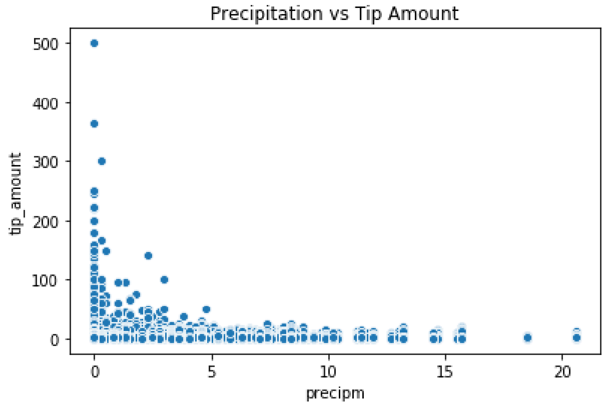


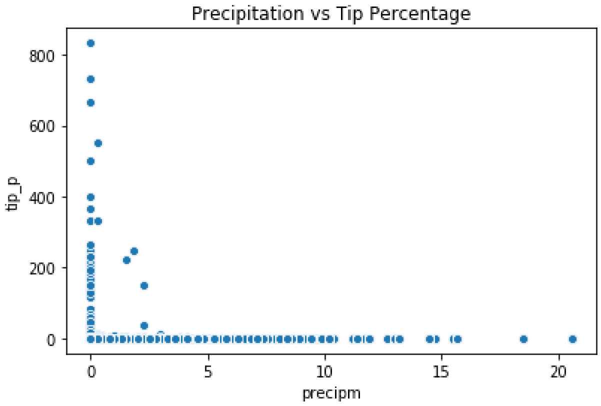
Distribution of Log of Tip Amount

Next we can explore the various weather attributes in relation to the tip amount. For example we can plot the tip amounts vs the temperature and see that it higher tip amounts tend to happen during the warmer days.

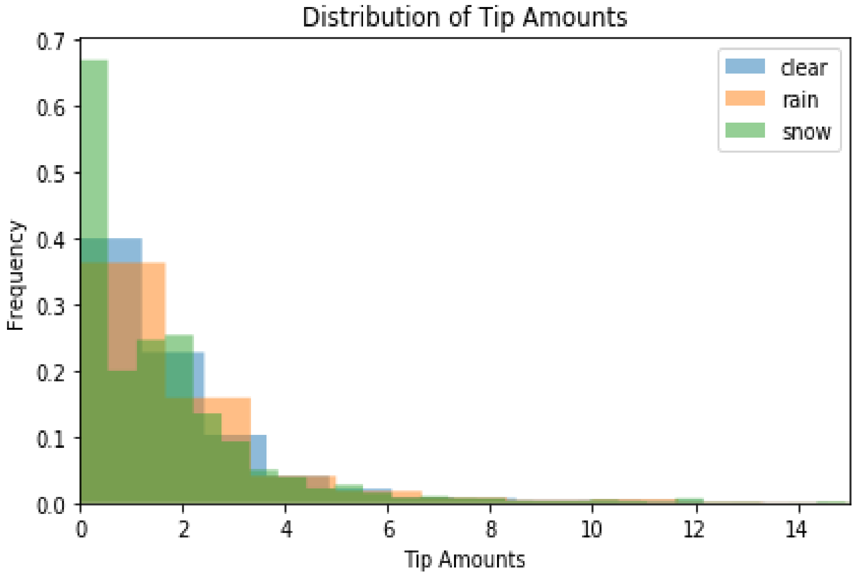




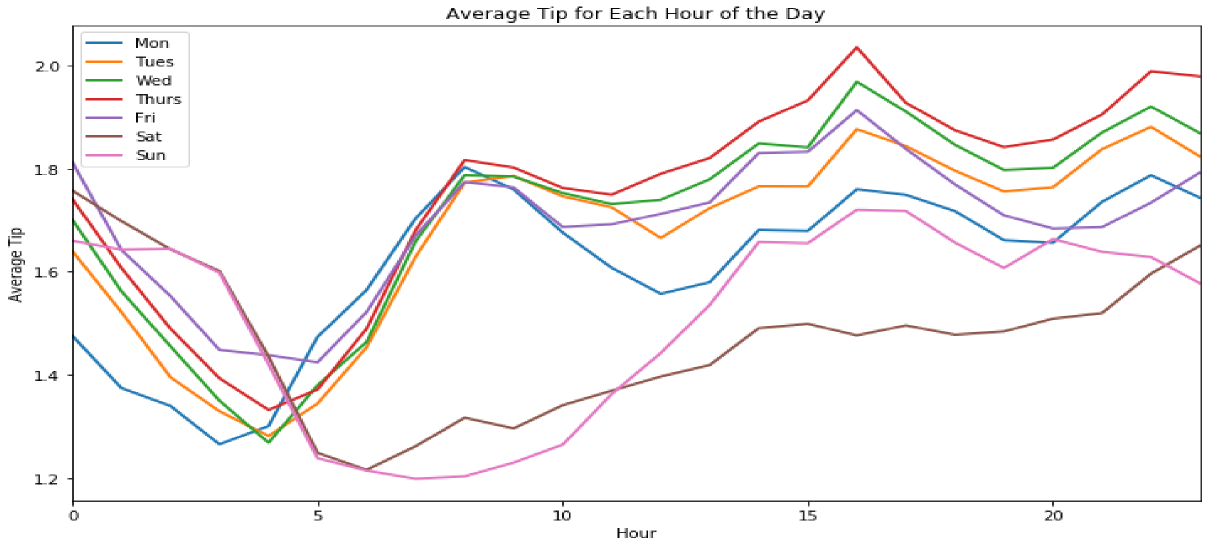




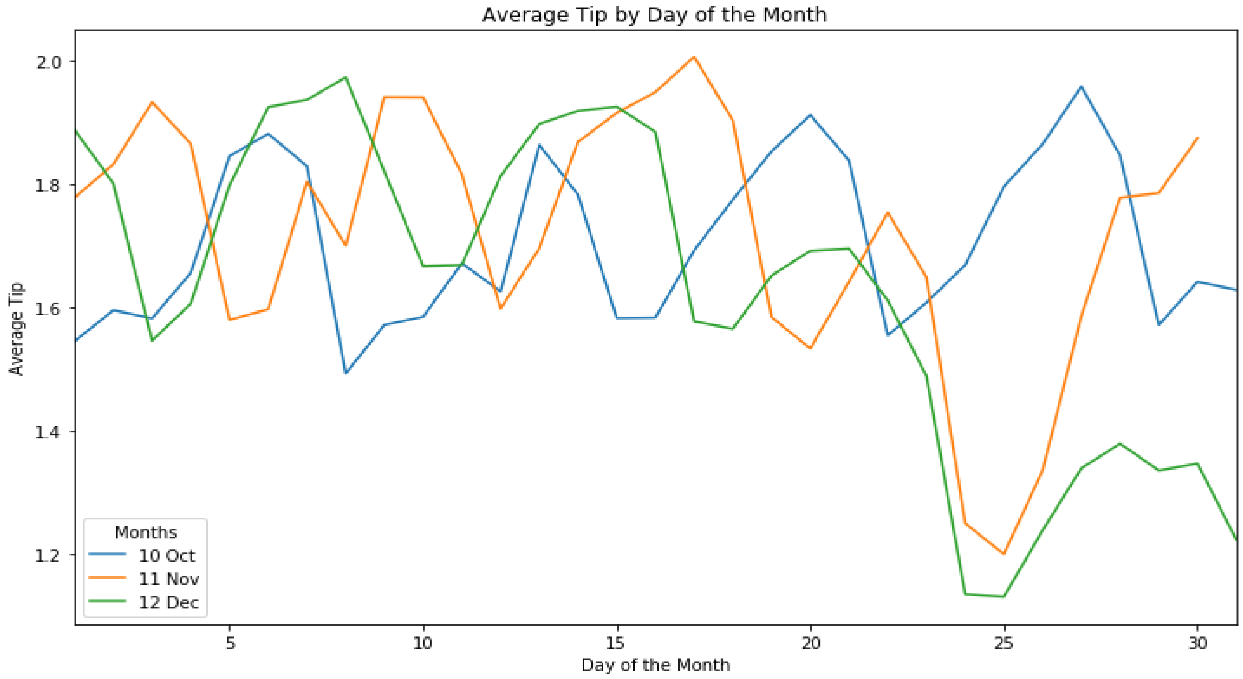
Plotting the distributions of tip amounts for rain, snow, and clear days suggests that the weather does change the distributions. The snow days seem to have the lowest mean tip amounts, and the clear days the highest.



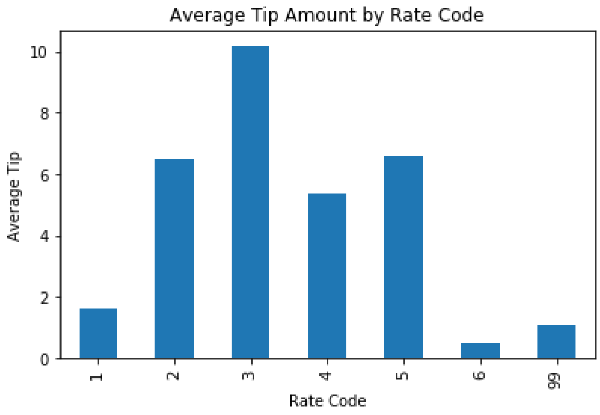
The day of the week seems to also have a clear effect on the amount of tips a taxi ride will get. From this plot we can see that Saturdays and Sundays generate a lower average tip, especially early in the morning.

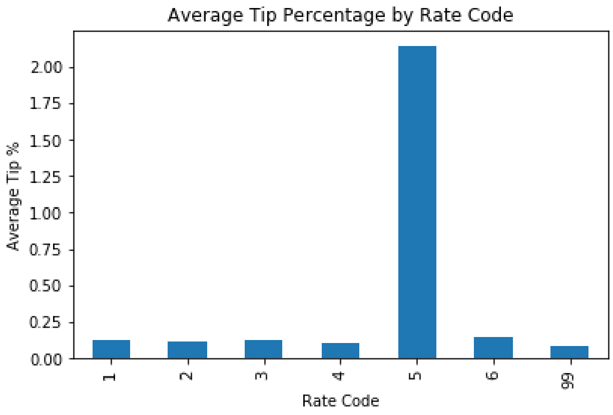


The day of the month seems to affect the average tip amounts as well. From the previous plot we can see that the weekends generate the lowest average tips. The weekend dips can be seen in the plots of the average tip by the day of the month. This also shows us that holidays affect the tip amounts as well. For example in the following plot, we can see that during thanksgiving (around nov 24) and christmas (dec 25) the tip amounts see a significant drop.

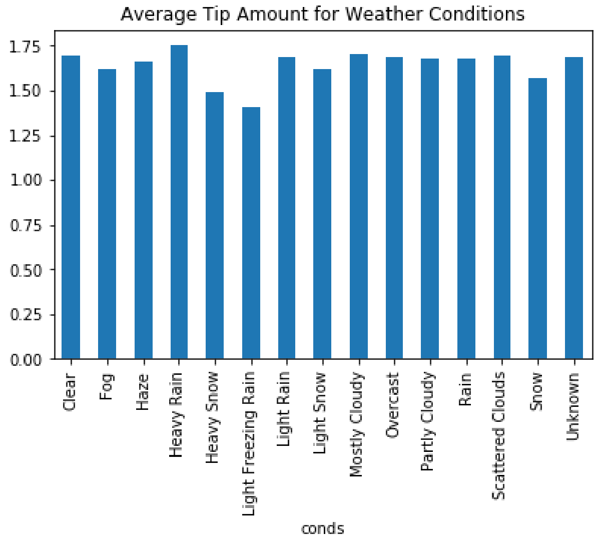


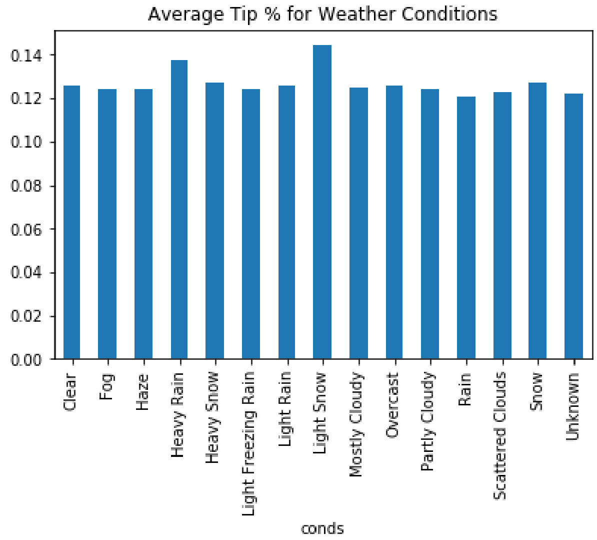
Lastly, we can see what the average tip amounts are across various categorical variables. For example, we can see what the average tip amount is for the different rate codes, as well as across all the weather conditions.





We can see that in general rate code 1 offers the lowest average tips, and rate code 3 the most. It’s interesting to note that rate code 5 offers the highest tip by percentage.





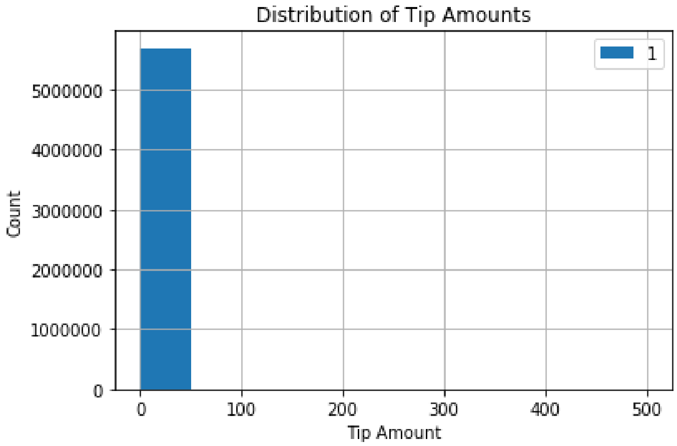
Here we can see that rides during light freezing rain have the lowest average tip, followed by heavy snow.

From exploring this data, it’s not entirely obvious which features have a large impact on the tip amounts. The weather does seem to play a role though, as seen in the temperature vs tips scatter plot, as well as the average tips across weather conditions. It’s also seems clear that the date as well as the hour of the day has an impact. We can see from our exploration that the weekends and holidays have a lower average tip amount. This seems counterintuitive, but one possible theory could be that maybe people tip more when they are in a hurry during the work day and don't think about the amount too much.

Statistical Analysis

From our exploratory data analysis, we can see that there are several categorical variables across which we can perform some statistical analysis.

Looking at the different Rate Codes, there are 6 categories of codes. After plotting the distributions of the tip amount across the different rate codes, we can see that they are right skewed and not normal (as shown below), so we can't use a typical t test.



We can however use the Kruskal-Wallis H test to compare the distributions of various samples. Using an alpha of 0.05 and a null hypothesis that the distributions are equal, we can see from the kruskal test that the distribution of tip amounts for rate codes 3 and 4 are similar, and the distribution of tip amounts for rate codes 6 and 99 are also similar.

We can repeat the kruskal test for the various payment types, as well as the rain/snow conditions. From these tests we can see that the payment types 1 and 2 have a p value below the alpha of 0.05 and satisfy the null hypothesis. Payment types 3 and 4 have a $p>0.05$ and so we can conclude that their distributions are similar. The kruskal tests also shows us that the distributions for the rain/snow/clear conditions are not the same.

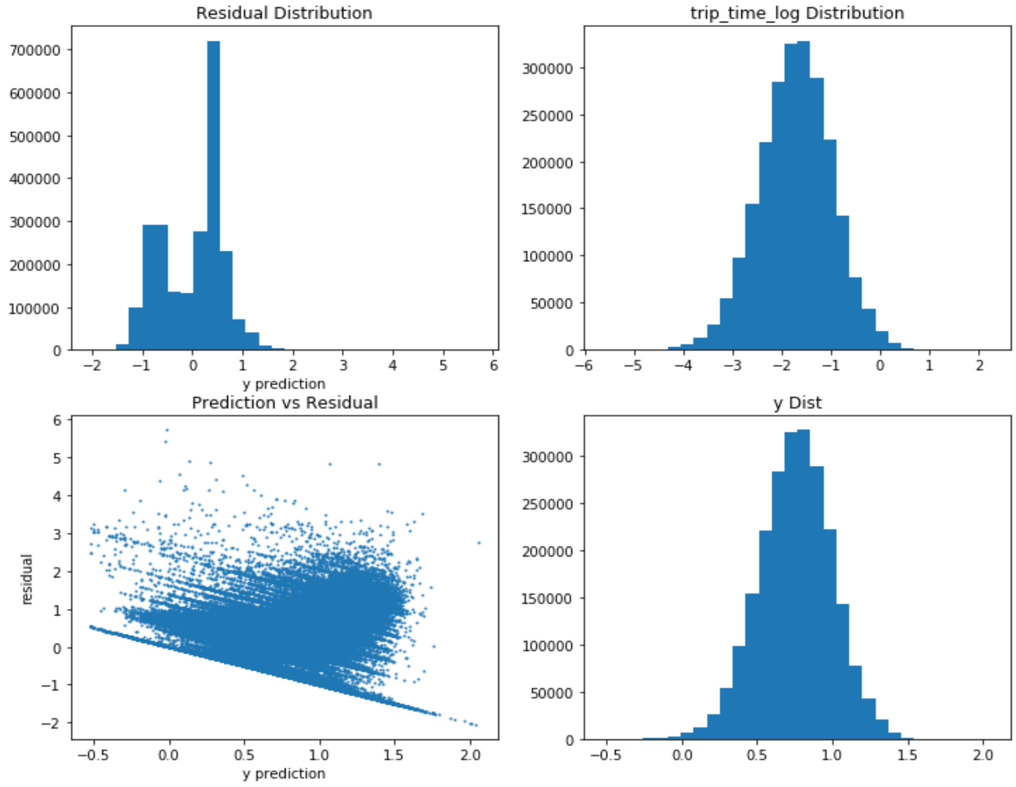
Looking at our categorical variables, we should perform a chi-square test to see if any of them are highly correlated. We should check for multicollinearity between our variables to help us in the following regression phase. Performing a chi2_contingency test between our categorical variables returns a $p < 0.05$ between all 3 pairs, suggesting that there is a statistically significant relationship between all 3 of our categorical variables. We should avoid using all of them in our model at the same time going forward.

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.

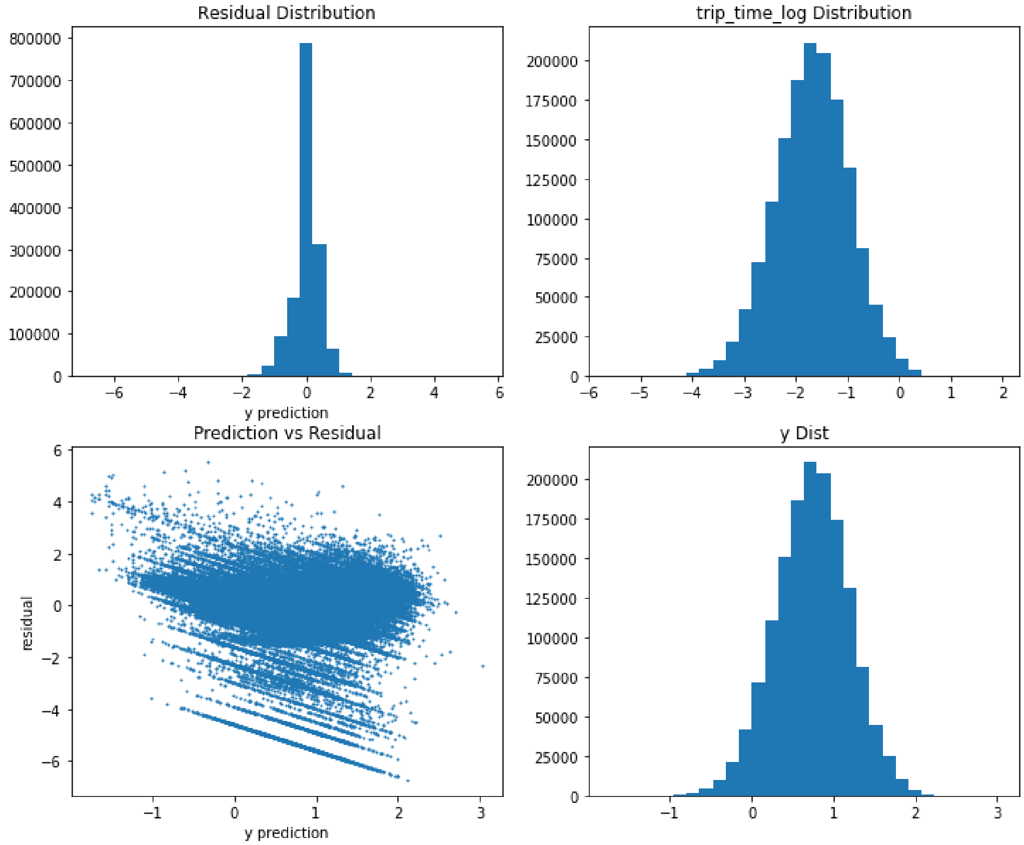
The next step to prepare our data is to find a way to handle our date objects. We have the various date attributes in their individual columns (ie month, date, hour, weekday), but they are represented in a linear fashion. For example hour 7 is before hour 9, and hour 13 is before hour 23. This doesn’t 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).

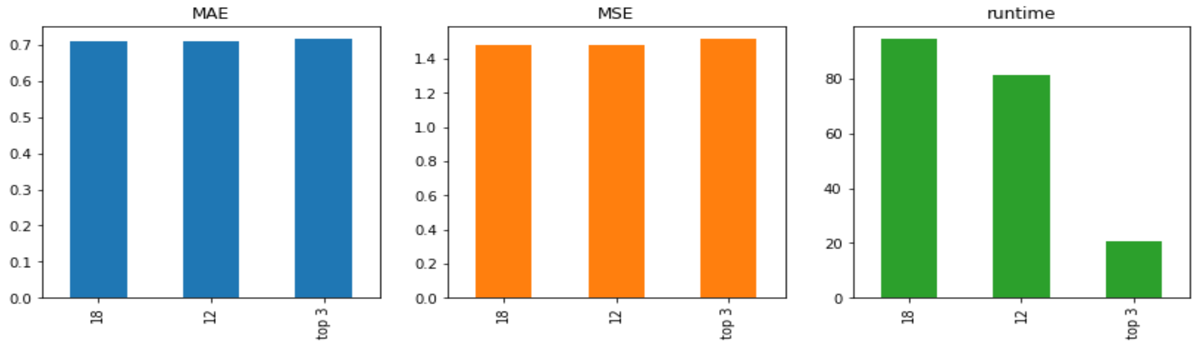
In an effort to try and reduce the heteroscedasticity, let's try a few more simple linear regression models trained on a data set without samples with a tip amount of 0. A linear regression model using the trip time and the average speed are created, but unfortunately did not solve the problem with our 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.