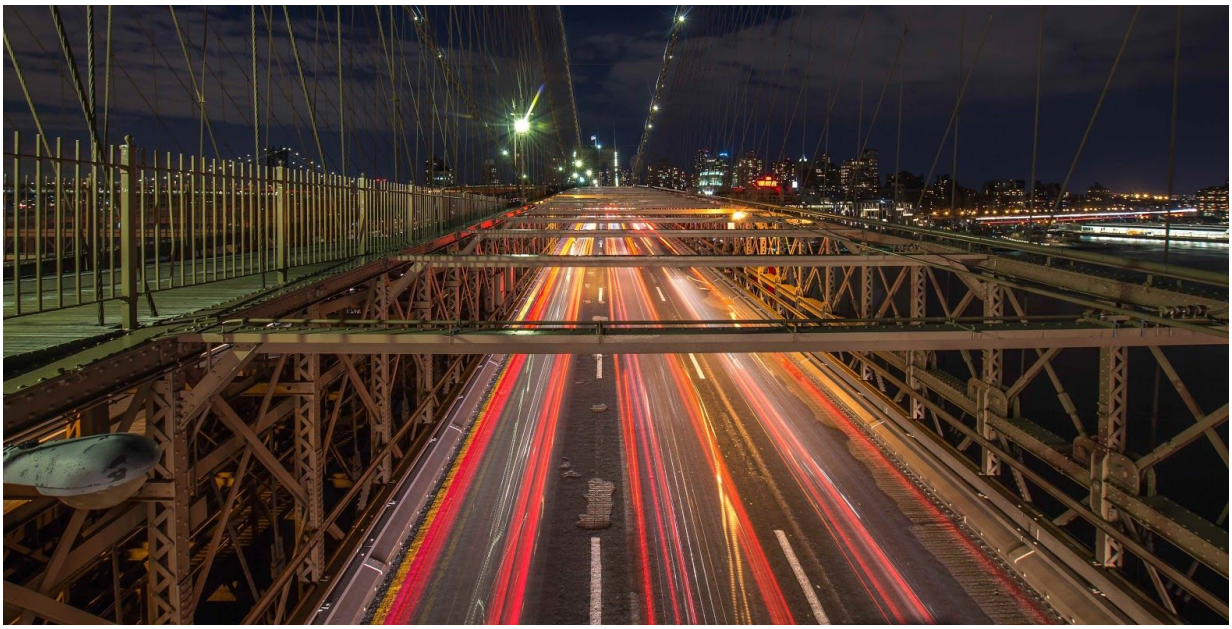
Predicting New York Traffic Accidents

Springboard Capstone 1 Final Report

Gene Hopping

December 2019



Problem Statement

New York City is the largest city in the United States, and the 28th largest in the world with an estimated population of 8.6 million people.[1] This population combined with over 6,000 miles of roads[2] and a large number of tourists every year ensures a challenging traffic situation. As a driver in New York city, it would be informative to know when is the most likely time you could get in a car accident, so you can try to avoid driving at that time. Is this time during morning, or afternoon rush hour? Late at night, or in the early hours of the morning? Or, is there no real correlation between the time of day and traffic accidents?

Data Wrangling

To answer these questions, this project utilizes two datasets, both available from the New York City Data Repository. The first is the NYPD Motor Vehicle Collisions - Crashes dataset, a dataset consisting of 1.58 million rows of data detailing the date, time, location (street name, cross street and lat and long coordinates), contributing factors, types and numbers of vehicles involved in the accident and the number of injuries.

https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions-Crashes/h9gi-nx95

The second dataset is the Traffic Volume Counts (2014-2018) dataset. This dataset contains the location, direction of traffic and number of vehicles recorded per hour.

https://data.cityofnewyork.us/Transportation/Traffic-Volume-Counts-2014-2018-/ertz-hr4r

The questions we're asking relate broadly to all accidents across all New York, therefore the specific locations of accidents, or vehicle counts is not needed. We're going to transform and aggregate the accidents (and counts), to give counts per hour. Each entry in the accidents dataset pertains to an accident, therefore we can sort by date and time and aggregate them to get the count. The volume data is already separated by hour and day, we just need to sort on these features and count to aggregate the counts observed for all the different roads.

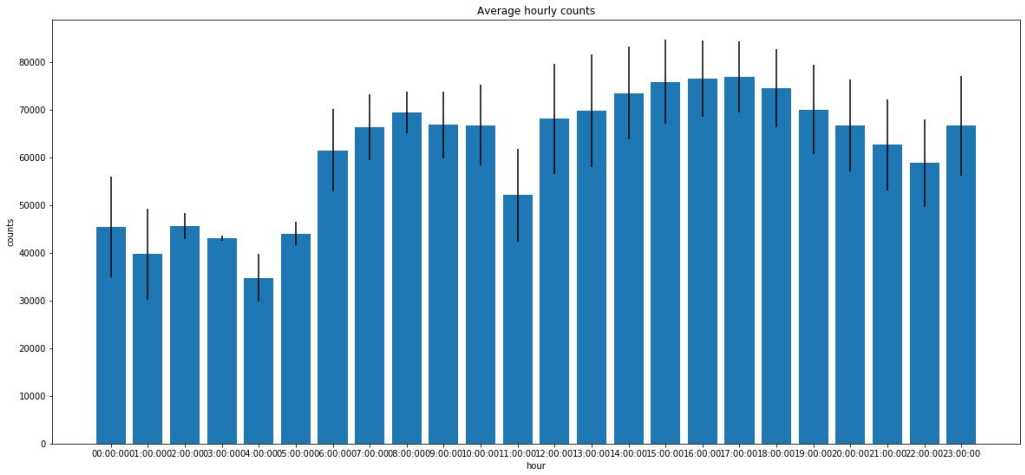
Volume data

The volume data were imported into pandas dataframes from csv files downloaded via the links above. Column titles were converted to lowercase, and any spaces were replaced with underscores to facilitate chaining. The data were provided in an untidy format - each hour period is a separate column. The separate hour columns were melted into a single time column, with each time span being reduced to a single hour representing the beginning of the hour window (for example 1:00 - 2:00 pm became 1:00 pm). From over 514,000 observations, there were 44 missing values. These entries were removed, and all dates and times were converted to type

[1] https://www.worldatlas.com/articles/the-10-largest-cities-in-the-world.html
[2] http://www.nyc.gov/html/dot/downloads/pdf/2013-dot-sustainable-streets-5-infrastructure.pdf

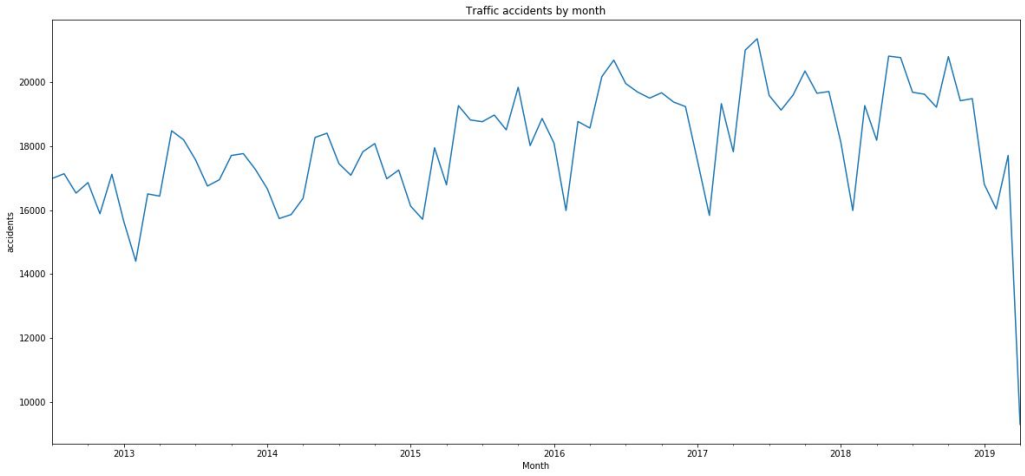
'datetime64'. A new datetime column was created and set as the index for easy indexing using the pandas .loc accessor.

Because we were interested in citywide traffic, each day and hour was aggregated to provide a new dataframe containing the datetime index, the date, the time and the vehicle count. This is the dataset that was used going forwards.

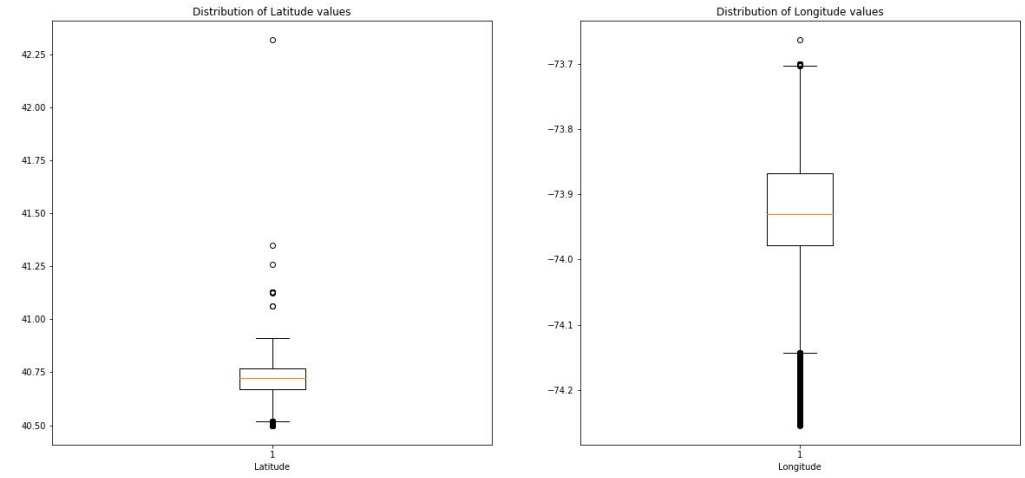


Accident data

The accident data was imported and the column titles converted in a similar fashion to the volume data. Again a datetime index was set. Each line in the dataset represented the time and location of an accident reported to the NYPD. Upon initial inspection of the data, the number of rows of datetime data did not match the number of rows of longitude and latitude. Closer inspection revealed over 185 thousand NaN instances, and over 900 values of 0 for longitude and latitude. 0 longitude and latitude is on the equator, south of Ghana. Since the date and location of an accident is of paramount importance to this work, all NaN values were first converted to 0 and dropped together with the existing 0 values. The removal of these entries resulted in the same number of datetime values as longitude and latitude.



Creating boxplots of the remaining latitude and longitude values showed that there was a large spread of data, larger than expected for the geography of New York City, with many outliers. Max and min values of latitude and longitude that would define the four corners of New York City were defined, and the latitude and longitude values were constrained to those criteria. From the initial 1.48MM date values and 1.29MM latitude values we obtained 1.13MM datetime, latitude and longitude values of cleaned data.

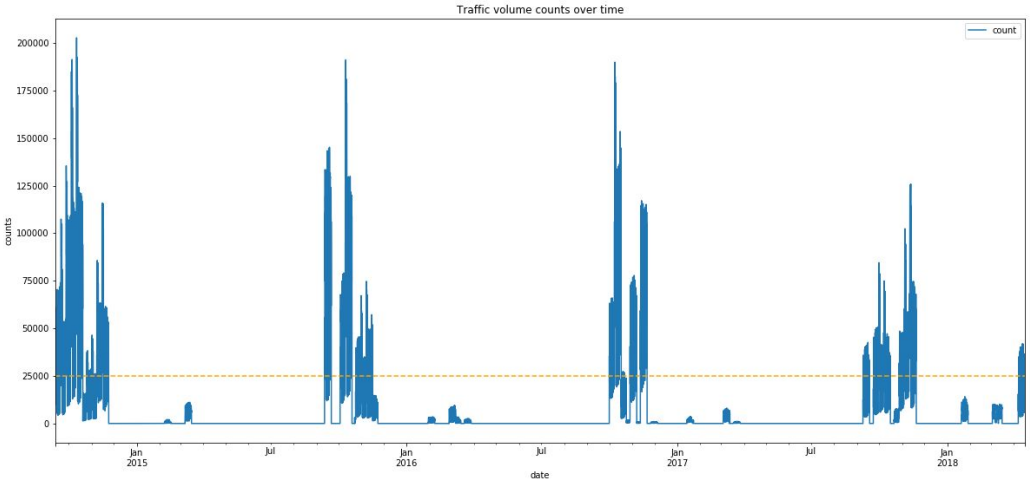


A new dataframe was generated by grouping by date and hour and counting the resulting rows, containing a datetime index, the date, hour, and accident count.

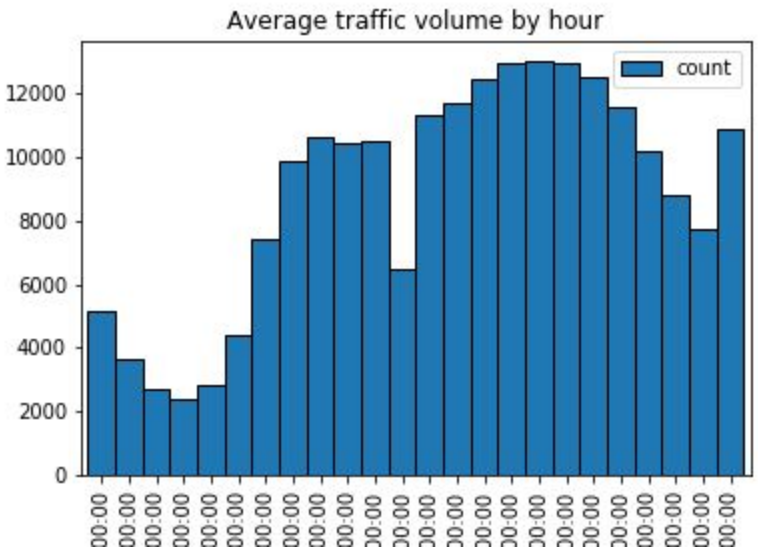
EDA

Volume data

Initial exploration of the volume data showed that counts with values > 0 were extremely sparse. Approximately 73% of the entries were 0 values. When all 0 values were excluded, a histogram of the counts shows that the smallest bin contained the most counts. The start date and end dates of each cluster was determined by looping over the dates and checking if counts exist. Once the date regions were defined, the values above an arbitrary threshold of 25,000 counts per day were copied to a new dataframe. This provided 7,944 traffic volume data points to use to combine with the traffic accident data.

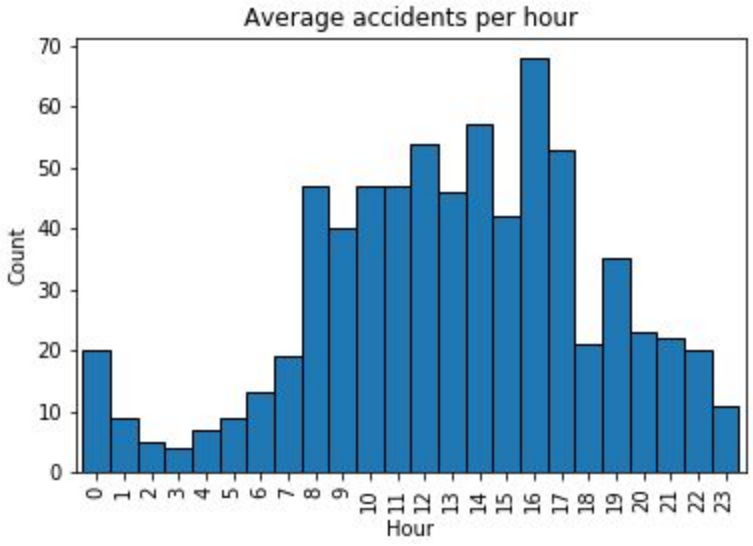


Plotting the hourly counts per day shows a bimodal distribution, which is to be expected as traffic flow generally peaks twice a day with the morning and evening rush hours.



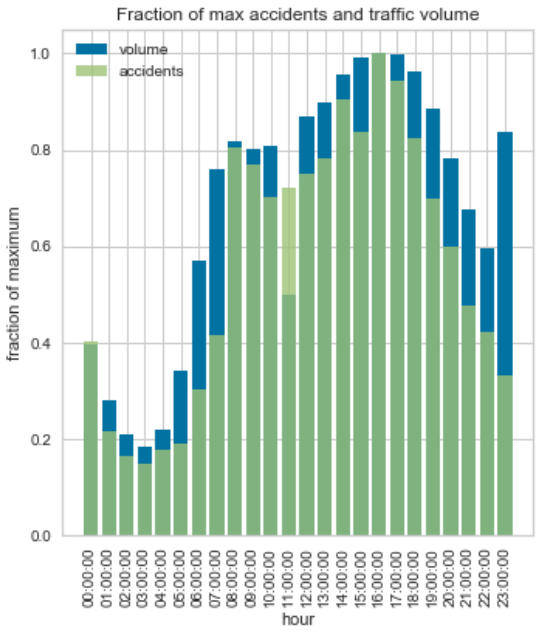
Accident data

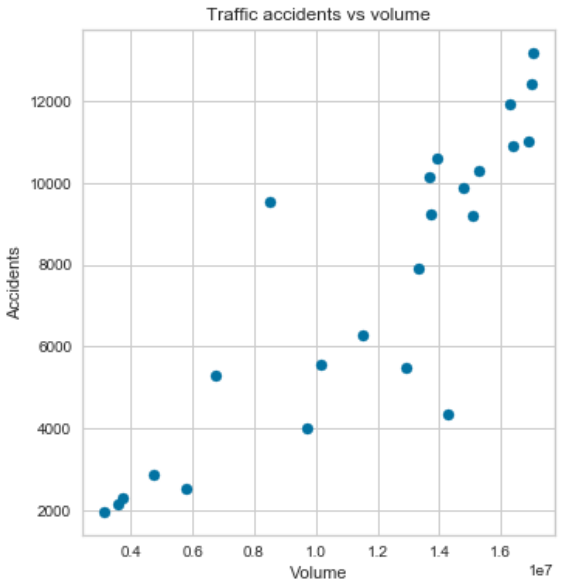
The accident data was also grouped by date and hour, and a new dataframe created keeping the date time, and including the number of accidents recorded. Plotting these data results in a somewhat similar bimodal distribution as the traffic volume data.



Now that we have both data sets in a form that we want, we can use all dates and times where we have count data for both. The volume data is the limiting data, and was selected based on a threshold to provide 7,944 data points. To obtain accident data for the corresponding hours of volume data, we performed an inside merge of the volume data with the accident data, keeping only rows that contained matching dates and time. A quick check showed that our final datatable contained the date, time, volume and accident columns, and 7,944 rows. Grouping the

accident data by hour in this manner resulted in the aggregation of 179,026 separate accidents into the 7,944 hourly time points.





Due to the disparity in counts between traffic volume and traffic accidents (which is a good thing!) the fraction of maximum was calculated per hour, by dividing each hourly count by the maximum hourly count. This allowed a direct comparison between the data sets. The traffic and accident data show a similar pattern when overlayed. When we plot these untransformed data against each other, we observe a roughly linear relationship. Upon observing this, we posed the question; can we represent the relationship between traffic accidents and volume, for all new york, using a linear regression model. A linear regression is arguably one of the most simple models available

We built a linear regression model and see how effective it was at reproducing the data. This model also allowed regularization to determine important coefficients. These coefficients, while important for defining the regression line, could also indicate the periods of the day where there is increased or decreased risk of accident. Considering that this is count data, we can also build a Poisson regression model for comparison.

Model Building

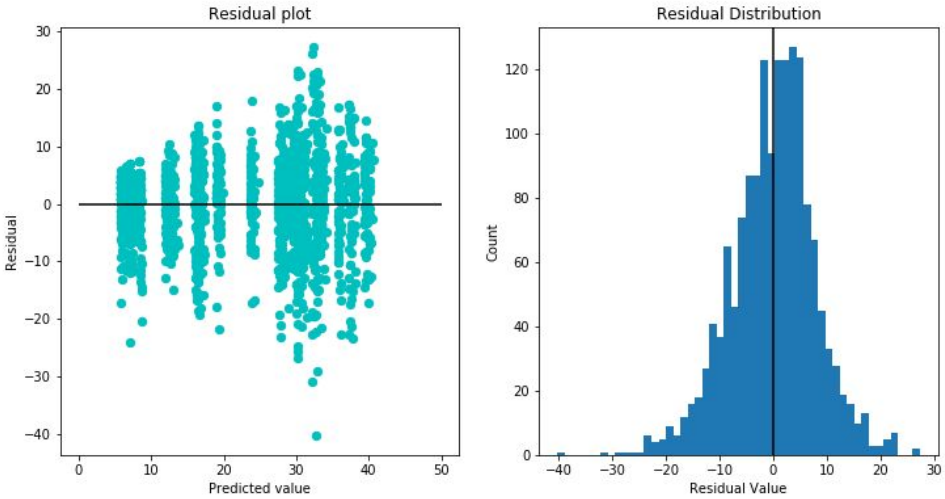
In this work, we're taking a rich dataset and distilling it down to simple counts over time. Linear regression is arguably one of the most simple machine learning algorithms, so we're going to apply this to our problem, to see how predictive a simple model can be.

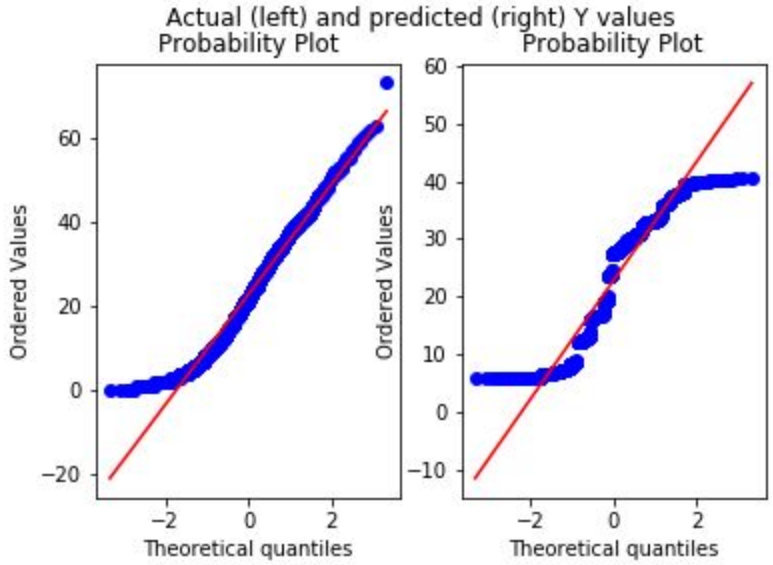
1. Linear Regression

Now that we have a dataframe containing the date, time, volume count and accident count, we can begin to build our models. To answer our first question, is the accident number simply a function of the traffic volume, we ignored the date and time and fit the model to the traffic volume alone. We used an ordinary least-squares (OLS) linear regression model implemented in scikit-learn. Results from 5-fold cross validation (CV) gave a mean R^2 value of 0.07, indicating that the number of accidents are not very correlated with traffic volume alone.

The second model we tried was another OLS linear regression, this time with the data aggregated by hour, and each hour represented by dummy variables. The date was also represented by an integer corresponding to days after the first date in the dataset. This model performed much better, with the mean 5-fold CV R^2 value being 0.63. This is quite remarkable considering the data is taken from accidents across all of New York City, with traffic volume inferred from only a subset of streets that were counted.

A residual plot shows they are centered around 0, but the pattern should be completely random. These values fan out toward the larger X values, indicating there is some heteroskedasticity about them. A histogram of the predicted values shows them again centered around zero, with some left skew.





The image on the left shows the distribution of the *actual* hourly accidents versus a normal distribution. For the most part the data follows a normal distribution, and the left skew can be seen by the data deviating from the diagonal on the left-hand side. The image on the right shows the *predicted* distribution of accidents versus a normal distribution. The deviation away from the diagonal for the upper values may indicate under-dispersion of the model.

3. Poisson Regression

Poisson regression is another generalized linear model, and is used to model count data. Traffic counts and accidents are examples of count data - data that can only be non-negative integer values. For example, it does not make sense to have -3, or 1.5 accidents. An assumption of the Poisson regression is that the response variable, Y follows a Poisson distribution. We again use the generalized linear model within the statsmodel module. Applying Poisson regression to our data, we immediately see the chi2 value is 1.86e4. If the data followed a Poisson distribution, the variance would equal the mean giving a chi2 value closer to 1. Clearly this is not the best model for this data. We can try negative binomial regression, which is a generalization of the Poisson model, with a relaxed requirement that the variance does not have to equal the mean.

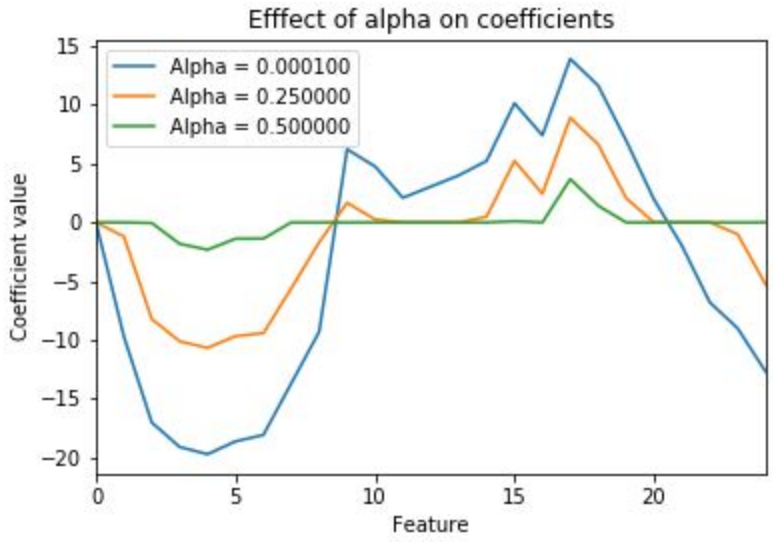
4. Negative Binomial Regression

The negative binomial model is similar to the Poisson with an additional variable, alpha, to lessen the strict variance requirement. The lambda values found during the Poisson regression were used to determine the optimal alpha value (for a tutorial see:

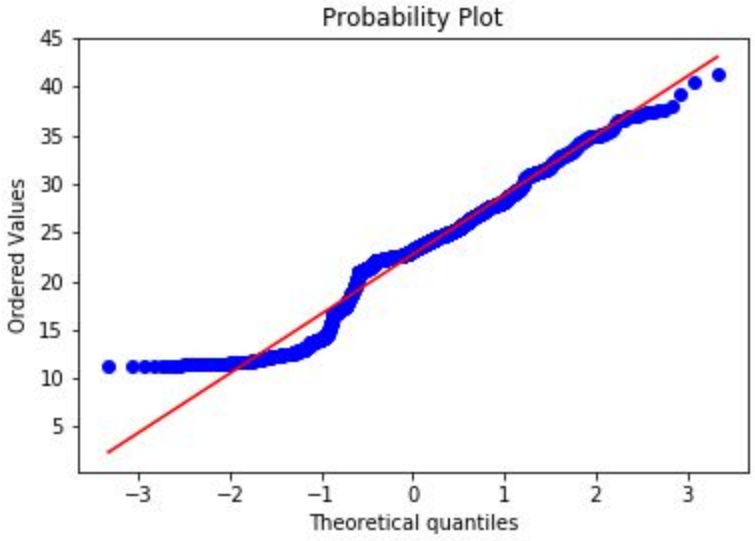
https://towardsdatascience.com/negative-binomial-regression-f99031bb25b4). An alpha value of 0.071 was found, and when negative binomial regression was performed using this value, the resulting model was shown to be statistically significant, which improved the model over the Poisson regression.

2. Lasso Regression

The second question that interested us was whether there was a particular time of day where there was a higher risk of being involved in an accident. One possible way of testing this would be to perform feature selection using regularization. To do this, we applied L1 regularization to the linear regression. Lasso or L1 regularization shrinks unimportant coefficients to 0, thereby selecting the most important features for a more parsimonious model. By applying lasso regularization to our linear model, the features that are slowest to approach 0 are the most important for the fit of the model. We chose alpha values of 0.001, 0.25, and 0.5. Applying higher alpha values reduced more coefficients to 0. Applying an alpha value of 0.25, the R^2 value was reduced to 0.52. Seven of the 24 hourly coefficients were reduced to 0, indicating these were the least important for the fit of the model. Of the remaining coefficients, the highest was at 2-6pm with a peak at 4pm, and to a lesser extent 8am.

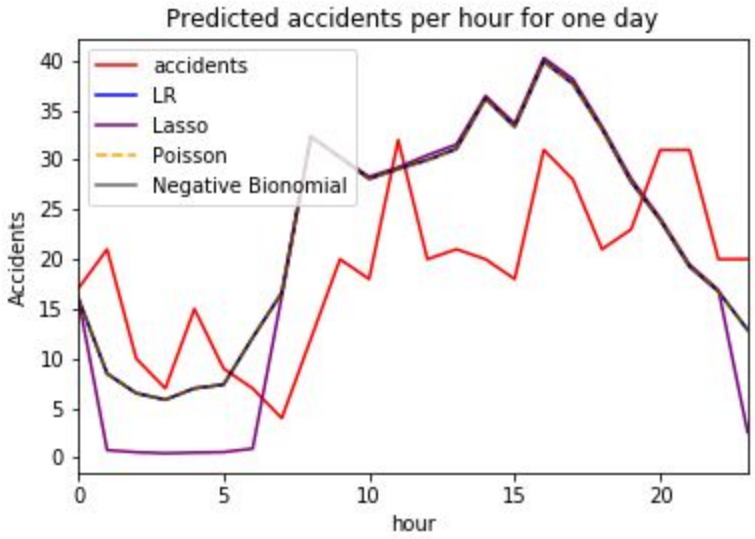


These coefficients correspond well to times of morning and evening rush hours. Unsurprisingly, the coefficient for the traffic volume is the smallest value at $5.6e^{-5}$, which is in agreement with the poor R^2 value from the model when fitting volume alone. If we look at the probability plot for the lasso regression, we see that the data has a more normal distribution despite having a reduced R^2 value.

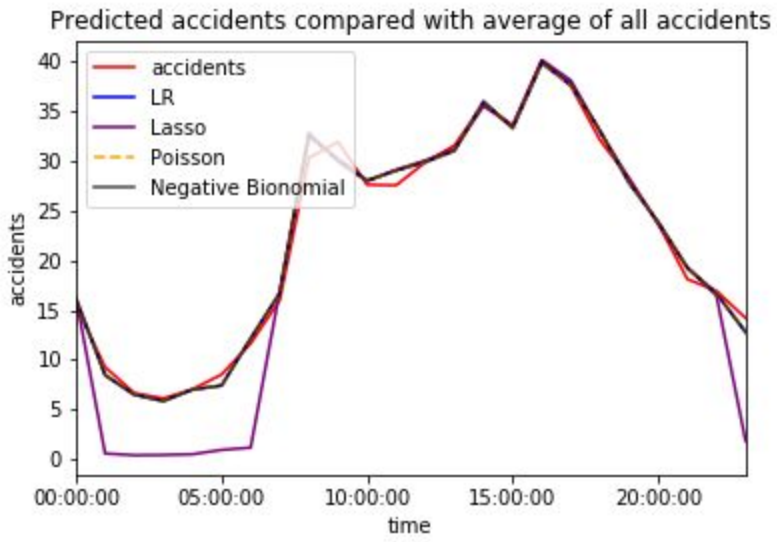


Assessing the models

Comparing predictions of the test set for each of the regressions against a single day of accident data reveals that all models show remarkably consistent predictions. The models seem to reproduce the overall trend, but generally overestimate the number of accidents. The deviation in the predictions, at least for the linear and lasso regression, is reflected in their modest R^2 values. It is also apparent that the lasso regression has a lower R2 value because it loses predictive ability in specific regions, not overall.



When we average all the hourly accident counts in the test set and compare the models against that, we of course see excellent agreement, indicating the models are capturing the average properties of the data.



Conclusion

179,026 accidents across 3.5 years were aggregated by hour and matched with traffic volume data to provide a final data set containing 7,944 hourly data points.
Linear regression revealed that traffic volume alone is a poor predictor of traffic accidents in New York city. The coefficients of the regression, emphasized by further L1 regularization revealed that 4pm was the most likely time to get in an accident.

Limitations of this model include inferring the traffic volume from possibly a subset of roads in New York. The variability of the counts observed in the traffic volume data suggest non uniform measurement, and therefore could result in inaccurate counts. To increase the accuracy of the model, the streets that contributed to the counts should be matched with the accident data, and not generalized across the entire 302 square miles of the city.