BigQuery-Geotab Intersection Congestion

Can you predict wait times at major city intersections?

Prepared by: Doyel Barman

Date: 03/17/2021

1. **Introduction**

The dataset used in this report includes aggregate stopped vehicle information and intersection wait times. The main purpose of this report is to predict congestion, based on an aggregate measure of stopping distance and waiting times, at intersection in 4 major US cities: Atlanta, Boston, Chicago and Philadelphia.

The data have been grouped by intersection, month, hour of day, direction driven through the intersection, and whether the day was on a weekend or not.

For each grouping in the dataset, we need to make predictions for three different quantiles of two different metrics covering how long it took the group of vehicles to drive through the intersection. Specifically, the 20th, 50th and 80th percentiles for the total time stopped as an intersection and the distance between the intersection and the first place a vehicle stopped while waiting. The goal can be thought as summarizing the distribution of wait times and stop distances at each intersection.

The future sections of this report describe the data preprocessing, a few methodologies and analyzing and understanding the results gathered from using the methods along with a discussion and conclusion.

1. **Data preprocessing**

Data preprocessing is done using Jupyter notebook, which is included with the submission of this report. In addition, all the work from now on related to models and metrics evaluation, is also done in the same Jupyter notebook file. As mentioned earlier, the original source contained two different files, one for training, and the other file is for testing. In this report, we only used the training dataset. This is simply to demonstrate that we do not necessarily need two different files, the dataset itself can be split into training and testing datasets.

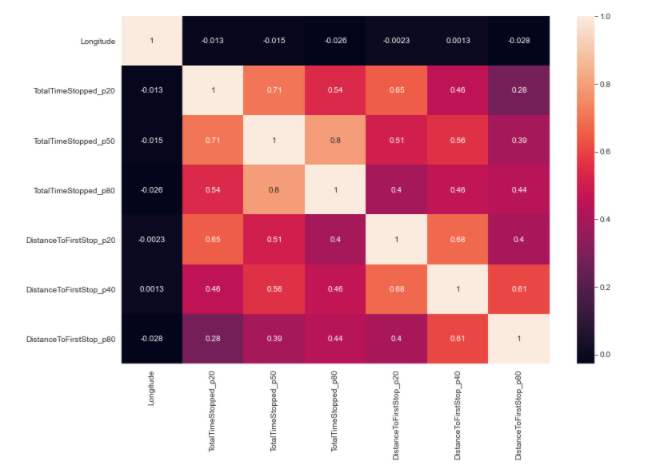
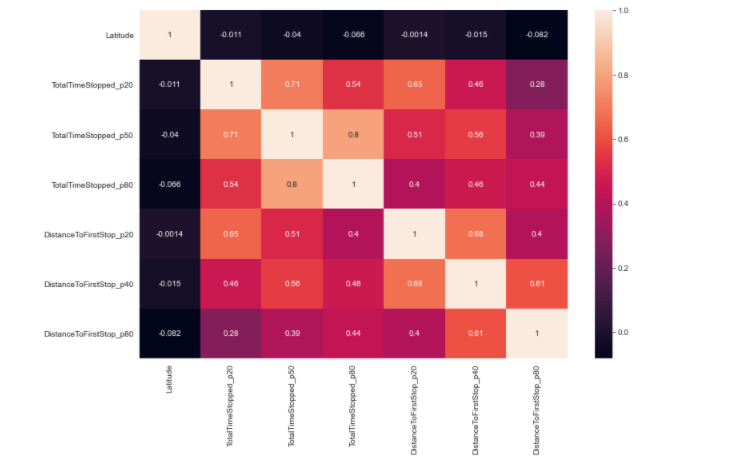
1. **Dropping NA/NULLs from the dataset**

we see that there are few NULLs in the dataset, so we dropped those NULL values.

1. **Feature Selection:**

This step is extremely important because we want features that are highly correlated with target class. To achieve this, we analyzed the dataset per feature:

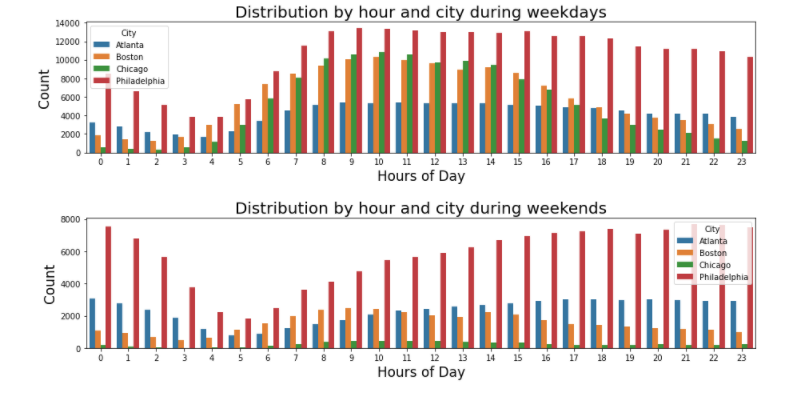
1. **Latitude and Longitude:** By analyzing the correlation matrix, we see that Latitude and Longitude have low correlations with the target variables.



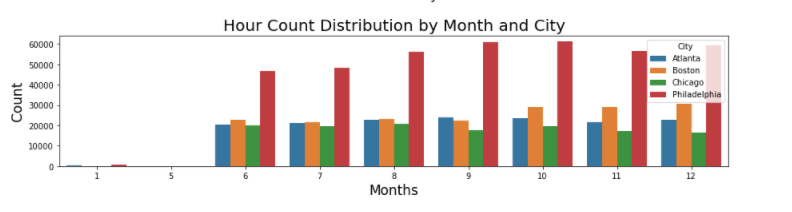
1. **Weekends:** Philadelphia have most of the total entries followed by Boston, Atlanta, Chicago.

During the weekdays, 08:00 to 17:00 is the rush hour in all cities, but for Philadelphia, it is 08:00 to 24:00.

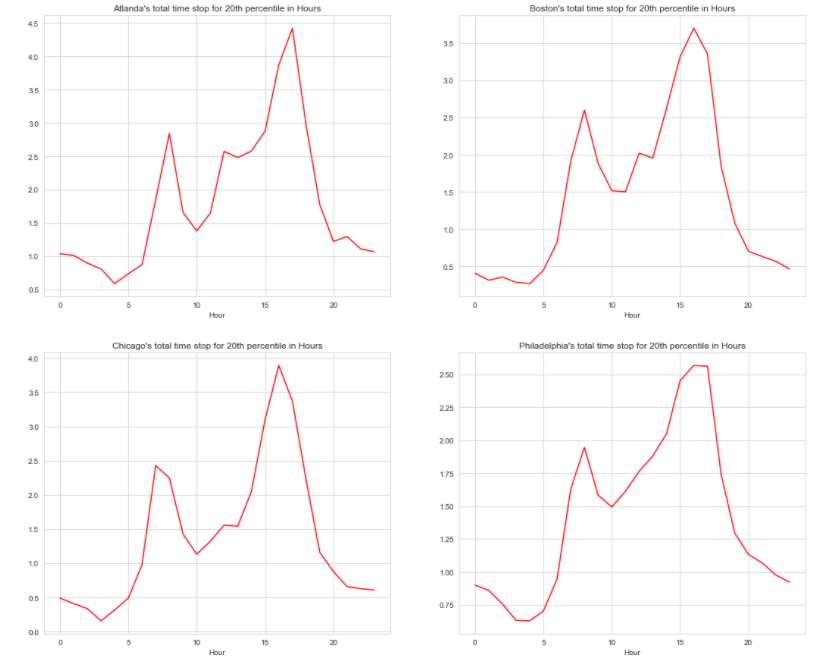
During the Weekend days: as expected, we see a peak on late night and throughout the day except in the early morning. So, weekend is an important feature in prediction.



1. **Month:** Month seems to be a good predictor as well: Here is an hour count distribution, although we only seem to have 6 months of data.

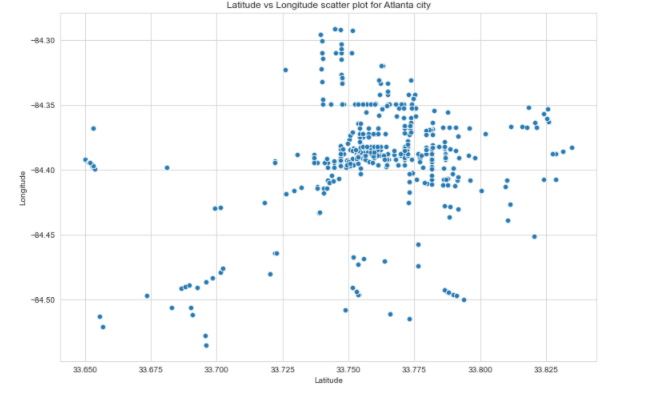


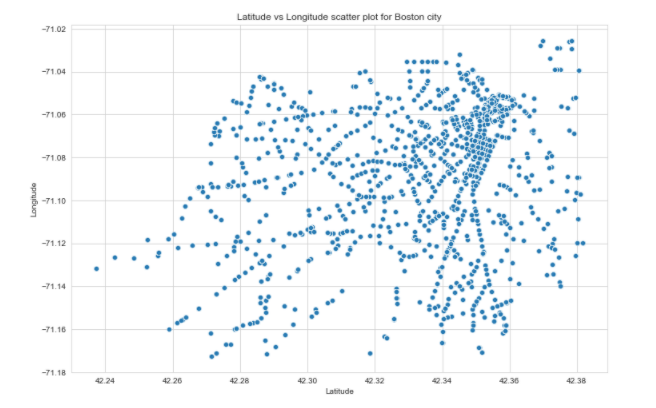
1. **Hour:** seems to be a good predictor as well

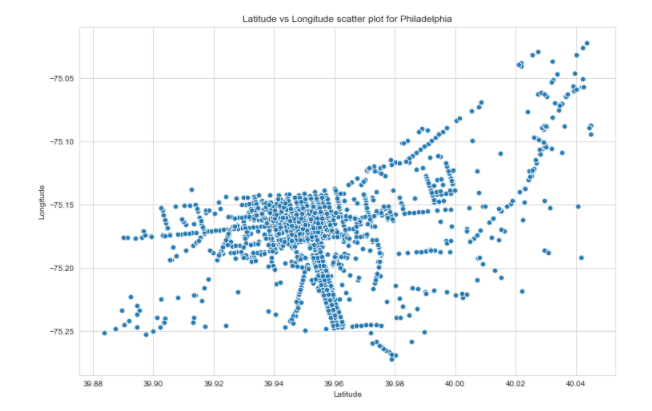
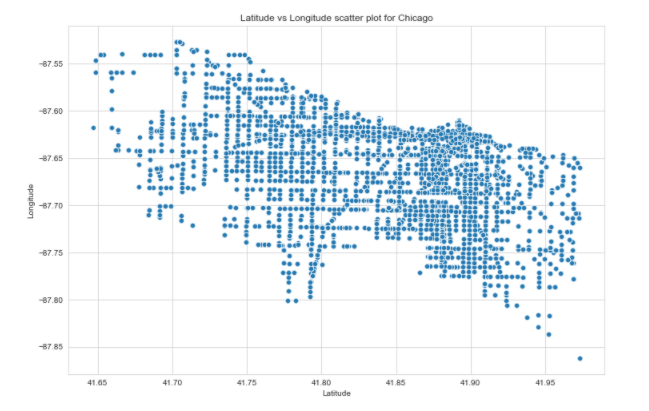


For all cities peak hours looks like from 15:00-18:00. Average time stopped for first 20% of the cars is significantly low but it shows the same peak hours as before for all 4 cities.

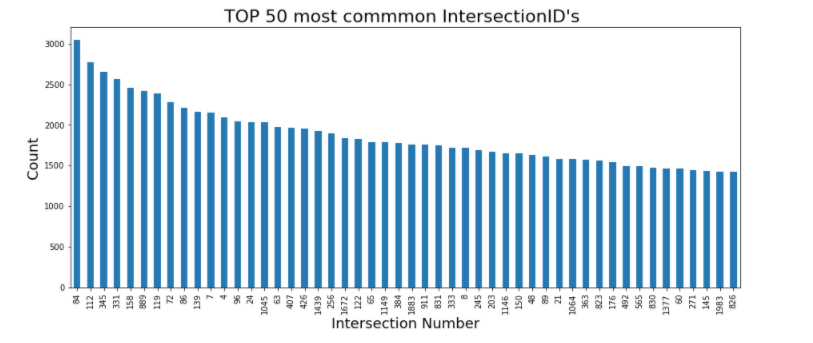
1. **Detecting outliers:** To detect any outliers, we chose scatter plots and saw that there are no outliers in this dataset.







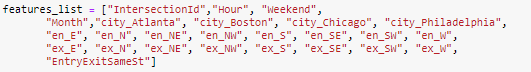
1. **IntersectionIds:** This is also a very important feature as we can see in the following chart.



1. **Feature Engineering**

Columns “City”, “EntryHeading” and “ExitHeading” are converted to numerical values using pandas get\_dummies method.

Finally, we end up with the following features for our models:



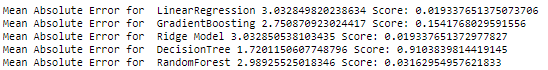
With the following target variables:



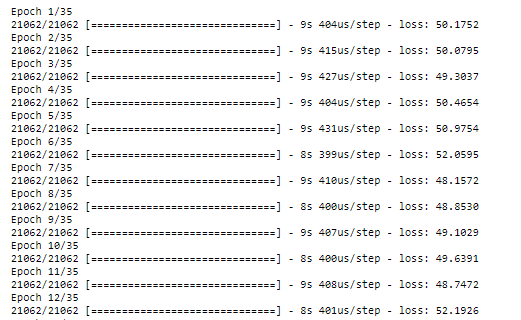
1. **Model selection:**

For this part, we have gone through multiple models with the first target variable to find a better performing model, and we decided to use Gradient Boosting Regressor model.

Below are the performances of multiple models used:

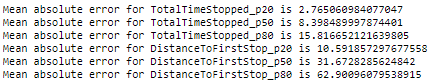


We have also used Keras Sequential model, adding a dense layer and calculated the Mean Absolute Error:



1. **Gradient Boosting Regressor Model**

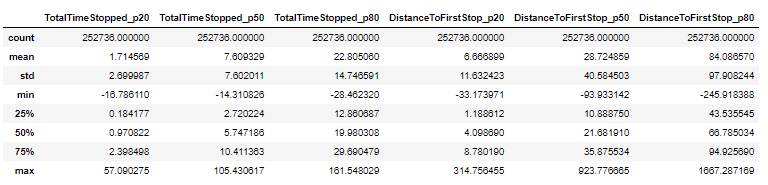
We used Gradient Boosting Regressor model to predict all the six target variables in a loop. Following is the Mean Absolute Error for each target variable:



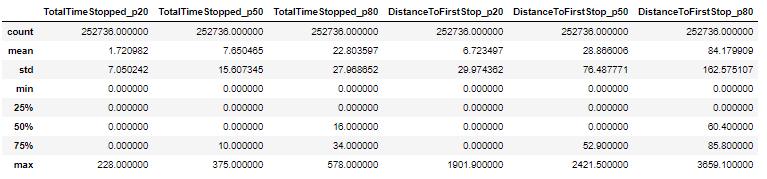
1. **Analysis of the predicted outcome**

After we get our predicted outcome, we did comparison between the actual values and the predicted outcome and noticed that the predicted outcome behaved very closely to the actual outcome.

Predicted outcome:



Actual:



1. **Output is attached with this project as Submission.csv**