BigQuery - Geotab Intersection Congestion

Rohit Marathe, Viswanathan Sankar Raman, Shreyas Surendra

Abstract

There are many instances where traffic congestion causes delay to the commuters. Traffic congestion is a common problem that is faced in various parts of the world. To address this problem, we try to predict the traffic, wait times at major cities. We use the data that was captured using telematic devices. Data pre-processing was performed to handle the missing values. Exploratory data analysis was performed on the dataset and features that had better weightage was selected. The features that we selected to predict the labels are month, intersection-id, city and direction towards and away from the intersection. The labels that we predict are total-time-stopped and Distance-to-first stop. The analysis of the results will help us to determine the major hotspots and to identify the bottleneck so that we can work on solutions to manage the flow of traffic in a better and efficient manner.

1 Introduction

Due to the increasing traffic congestion in various cities, we wanted to explore and understand the reason behind this and so we try to predict the wait times at major city intersections. Traffic has been a rising concern at many cities around the world. Due to traffic, lot of energy and time is wasted on the roads. These can be minimised if we can identify the important places where traffic causes major congestion. Modern telematic devices are used to capture real time data on the traffic from major cities. The dataset comprises of aggregate stopped vehicle information and intersection wait times based on an aggregate measure of stopping distance and waiting times, at intersections in 4 major US cities: Atlanta, Boston, Chicago & Philadelphia. We try to predict the wait times and thereby find the intersections that cause major congestion.

Various factors are considered when building a model. The features such as time of the day play a very important role in the amount of traffic that is present at that time. Also, one of the factors is if it is a weekday or weekend that is important in predicting the amount of traffic that is to be encountered in that particular time. Time of the day also plays a major role in the

amount of traffic that is expected. During the peak hours and weekdays, we can expect more traffic than at a time like early morning or in the afternoon. This feature is also considered for the model training.

2 Problem Statement

Our objective is to assist the aid city designers and governments to foresee traffic hot spots beforehand so that it reduces the stop-and-go stress of commuters. We achieve this by predicting the stop time at various intersections using the data that is obtained.

3 Related Work

A hackathon was conducted on similar topic by SMDL for predicting the traffic congestion. Geotab has the collection of many datasets that are aggregated over the time. They have been collected by using telematic devices from various places.

4 Proposed Approach

Perform Exploratory data analysis and select the important features that impact the outcome of the model. Use the features that gives better results. For this experiment we are focusing on three models to get the result and we have decided to run these models based upon their application in the regression. The goal of our experiment is to get the maximum score possible. Once we finalized the labels that we are going to predict, after the exploratory data analysis we tried to use methods such as XGboost, Random Forrest and Light Gradient boosting algorithm on the data to find out the best possible approach

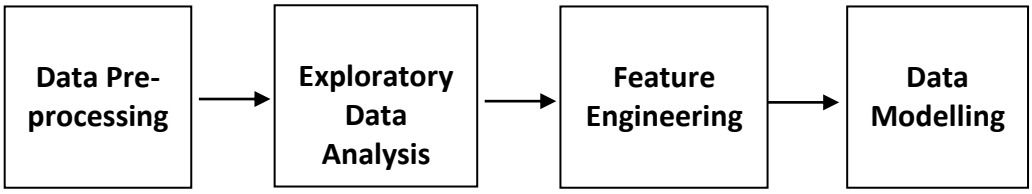


Fig 1: Pipeline

5 Technical details of proposed approach

5.1 Data Description

The data set contains the following information the intersection ID, month, hour of day, direction driven through the intersection, and whether the day was on a weekend or not.

Intersection ID: This column consists of unique IDs of an intersection that are unique to city.

Month: This column represents the month in which the vehicle information was recorded.

Hour of day: This column represents the hour of the day that vehicle information was captured.

Direction through intersection: This information is represented using two columns EntryHeading and ExitHeading. Entry Heading is the direction of the vehicle towards the intersection. Exit Heading is the direction of the vehicle away from the intersection.

Street Names: This information is represented using two columns EntryStreetName and ExitStreetName. Entry street name is the name of the street where the traffic or vehicle enters the intersection. Exit street name is the street where traffic or vehicle exits the intersection.

Path: This column is a concatenation of EntryStreetName and ExitStreetName. It represents the entire path of the traffic towards and away from the intersection.

Weekend/Weekday: Represents whether the traffic information was captured on a weekday or weekend. It is indicated with 1 or 0 where 1 is weekday and 0 is weekend.

TotalTimeStopped: It represents the total time that the vehicle stopped at the intersection. This is recorded in terms of twenty, forty, fifty, sixty and eighty percentiles.

DistanceToFirstStop: It represents how far before the intersection the vehicle stopped for the first time. This is recorded in terms of twenty, forty, fifty, sixty and eighty percentiles.

TimeFromFirstStop: It represents how long it took from that point to cross the intersection. This is recorded in terms of twenty, forty, fifty, sixty and eighty percentiles.

The train and test dataset are in the following shape 856387 rows, 28 columns and 1921357 rows, 13 columns, respectively. The dataset contains six categorical features namely EntryStreetName, ExitStreetName, EntryHeading, ExitHeading, City, Path.

Four numerical features namely Intersection ID, Hour, Weekend/Weekday, Month are also treated as categorical. The only numerical values used from this dataset is Latitude and Longitude.Two features RowId and Path, which possess no useful information with respect to the problem statement were eliminated straightaway.

Below is a sample of the dataset and the attributes that are present in them.

Feature names	Data Type	Null Values	Unique Values
RowId	Numeric	0	856387
IntersectionId	Numeric	0	2559
Latitude	Numeric	0	4799
Longitude	Numeric	0	4804
EntryStreetName	Categorical	8148	1723
ExitStreetName	Categorical	6287	1703
EntryHeading	Categorical	0	8
ExitHeading	Categorical	0	8
Hour	Numeric	0	24
Weekend	Numeric	0	2
Month	Numeric	0	9
City	Categorical	0	4
Path	Categorical	0	15075
TotalTimeStopped_p20	Numeric	0	171
TotalTimeStopped_p40	Numeric	0	238
TotalTimeStopped_p50	Numeric	0	262
TotalTimeStopped_p60	Numeric	0	306
TotalTimeStopped_p80	Numeric	0	403
TimeFromFirstStop_p20	Numeric	0	244
TimeFromFirstStop_p40	Numeric	0	316
TimeFromFirstStop_p50	Numeric	0	336
TimeFromFirstStop_p60	Numeric	0	353
TimeFromFirstStop_p80	Numeric	0	355
DistanceToFirstStop_p20	Numeric	0	3631
DistanceToFirstStop_p40	Numeric	0	6415
DistanceToFirstStop_p50	Numeric	0	7751
DistanceToFirstStop_p60	Numeric	0	9826
DistanceToFirstStop_p80	Numeric	0	13689

Table 1: Train Data Description

5.2 Data Exploration

Exploratory data analysis was performed to get insights about the data, and to decide upon the feature set to feed into the model. The analysis was directed towards exploring features that affect the traffic congestion at the intersection. Since the congestion, in our problem statement, is defined using three labels TotalTimeStopped, DistanceToFirstStop and TimeFromFirstStop, the analysis was focused on determining the effect of all the features on these three labels.

5.2.1 How direction of traffic affects congestion

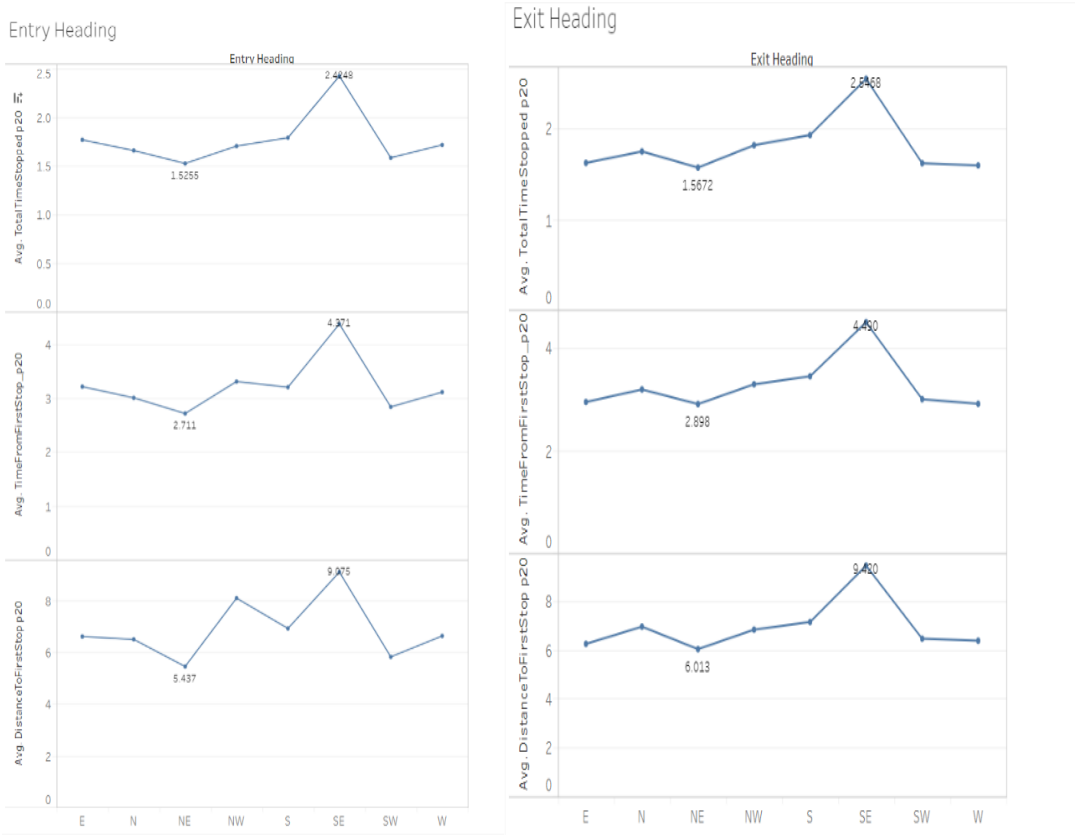


Fig 2: Direction vs Congestion

From the above visualization it can be observed how the congestion pattern varies with respect to the direction of the traffic flow. The traffic heading towards the intersection from South East direction is more prone to longer wait times than the traffic heading from North East. Same pattern can be observed with Exit direction, traffic exiting the intersection towards South East faces longer wait time than the traffic exiting towards North East. Due to the dependency of congestion on traffic direction, entry and exit direction was included in the feature set.

5.2.2 How hour of the day affects congestion

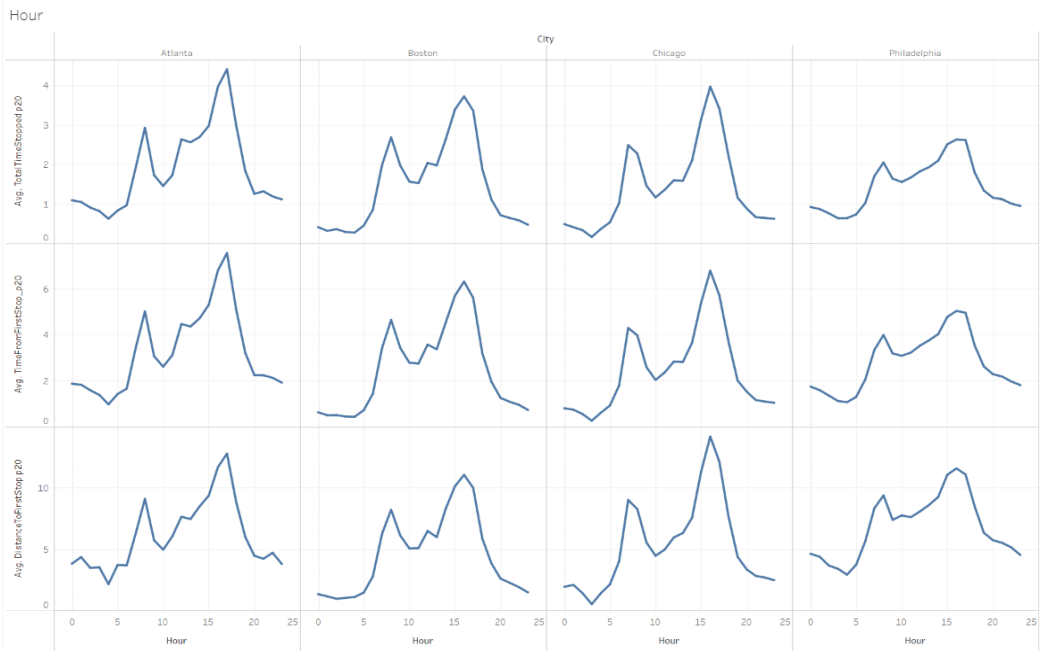


Fig 3: Time of the day vs Congestion

The above visualization shows how congestion time varies with hour of the day. Traffic congestion is at the peak during 17th hour of the day and lowest during 5th hour of the day.

5.2.3 How weekend affects congestion

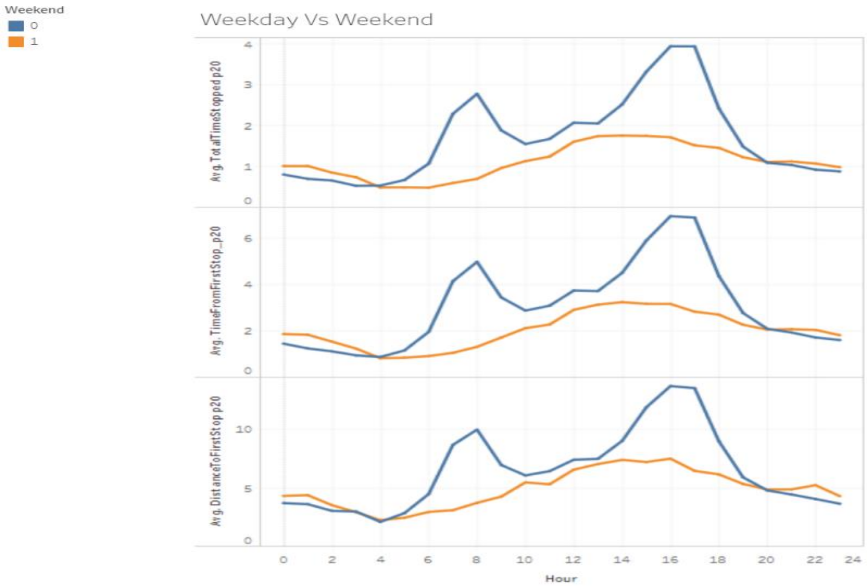


Fig 4: Weekday/Weekend vs Congestion

It was observed that day being weekend or weekday affects the traffic congestion. The weekend curve from the visualization is close to flat but the weekday curve has peaks and there is a lot of variation. Weekday/Weekend feature is included in the feature set as it is an important feature in determining the congestion.

5.2.4 How congestion is different in each intersection

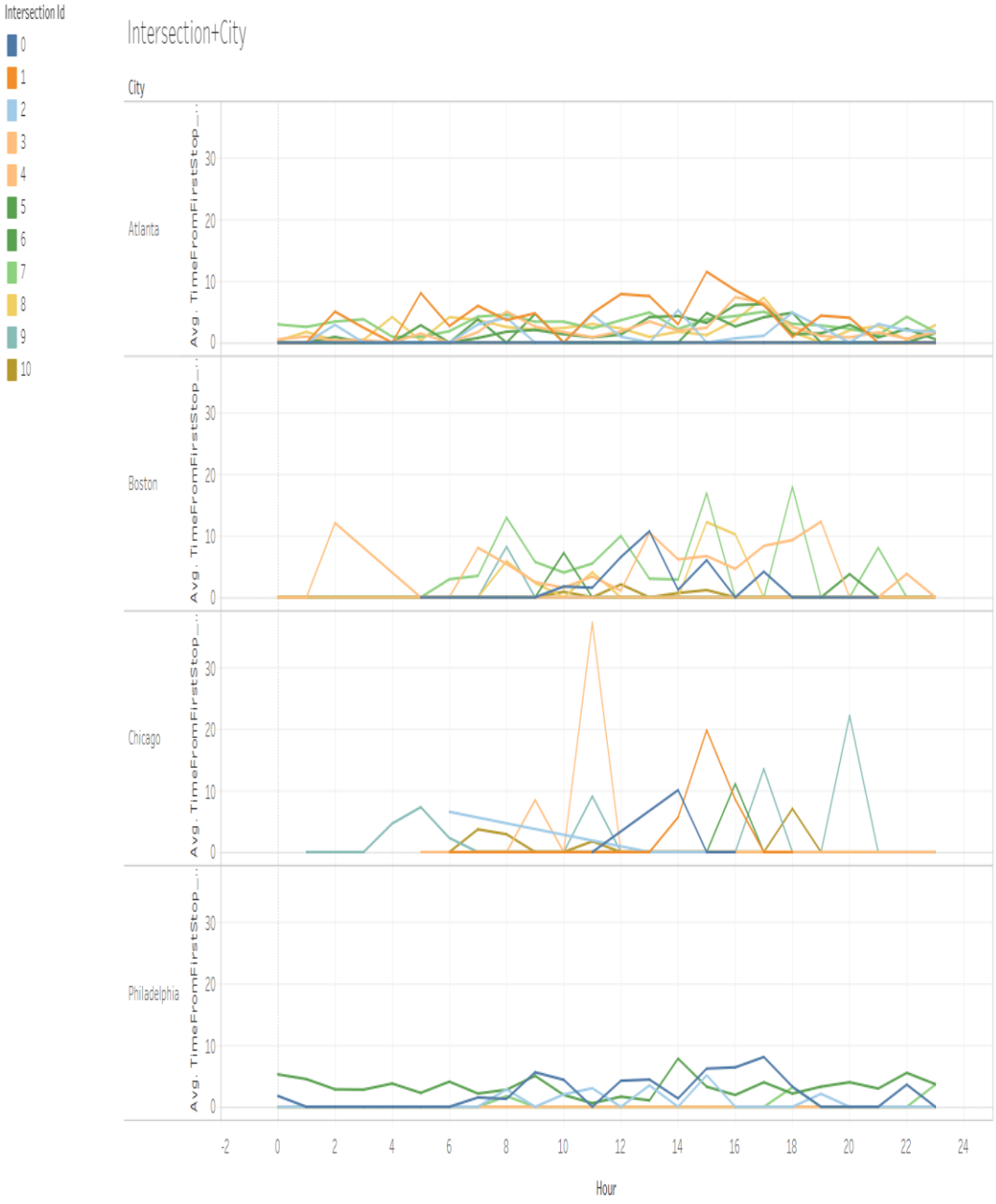


Fig 5: Intersection + City vs Congestion

The above visualization shows how traffic congestion is different in different intersections, for a given city. For instance, Chicago's intersection 3 has an extremely high congestion, and Atlanta's intersection 1 has high congestion. Since congestion varies with respect to intersection and the city both the features are included in the feature set.

5.3 Data Pre-Processing

It is an important stage in the Machine Learning pipeline which focuses on improving the quality of the data. Developing a model on unstructured raw data does not yield good results, and the data should be prepared and provided in the form the model can understand.

5.3.1 Data Cleaning

In this step, incomplete and inaccurate parts of the data are removed. In the dataset used in this project there were records with missing values under column PATH, which is of categorical type. Since this column did not possess useful information, the missing values were handled by eliminating the column form the feature set.

5.3.2 Categorical Encoding

In this step the non-numeric data is transformed into its numeric form. The techniques to perform categorical encoding are Label Encoding, One-Hot-Encoding and Learned Embedding. In this project, label encoding was used to encode categorical values. The features that were encoded include Month, City, ExitHeading and EntryHeading.

5.3.3 Feature Selection

Irrelevant or partially relevant features can negatively impact model performance, hence in this step the feature which does not contribute much to the relevancy of the model performance is eliminated. This step can be performed using various algorithms, but in this project this step was handled during exploratory data analysis.

5.3.4 Feature Extraction

Feature Extraction aims to reduce the number of features in a dataset by creating new features from the existing ones and then discarding the original features. This was applied to EntryStreetName and ExitStreetName to extract the type of the road, for example Road, Street, Bridge, Boulevard etc. Hence a new feature called Street Type was created. Also, a new feature was created using feature crossing of City and Intersection Id.

5.4 Data Modelling

5.4.1 Light Gradient Boosting Regressor

Light GBM is a good performance gradient boosting framework that uses decision tree for prediction. This method is known for splitting the tree with respect to the leaf nodes. This results in lower loss when compared to other methods that tend to split the data based on the levels. Light GBM works well when the dataset is large, and it gives quicker results. Hence, we selected this method for modelling the data.

5.4.2 Extreme Gradient Boosting

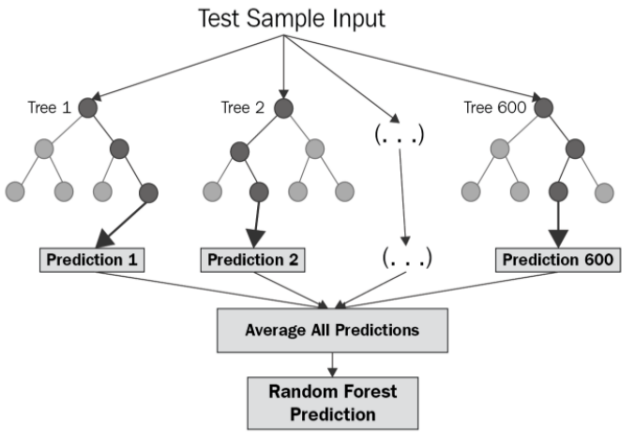
This algorithm is an extension to the normal gradient boosting, this makes the original model faster and more robust. The performance of this model is better and requires less computation time to compute the output. The model works very well-organized data and with tabular structured data. The model makes efficient use of all available resources and produces strong results.

We are making use of Xgboost library to get the functionality of this model. We can also fine tune this model to reduce the overfitting by adjusting the learning rate while describing the model. This model is widely used in Kaggle competition for achieving high ranks. With the current set of conditions, the model performs well and has a better RMSE score compared to the normal Gradient Boosting model.

5.4.3 Random Forest Regressor Model

Random Forest Algorithm can be used for classification and regression problem, in this algorithm decision trees are made using the training data. It's also an ensemble learning method. The efficiency of the model is dependent upon the various trees that are generated during the training phase.

As we had a huge amount of training data and this model generally suffered from huge training time. When we kept the value of tree value to 200 while training and 50 also while training the model performed similarly on both occasions with 200 configuration has a better RMSE score. Given the current scenario of predicting 6 labels and computational cost the model performances the best.



RF Structure [1]

Fig 6: Random Forest Model

6 Experiments and Final Analysis

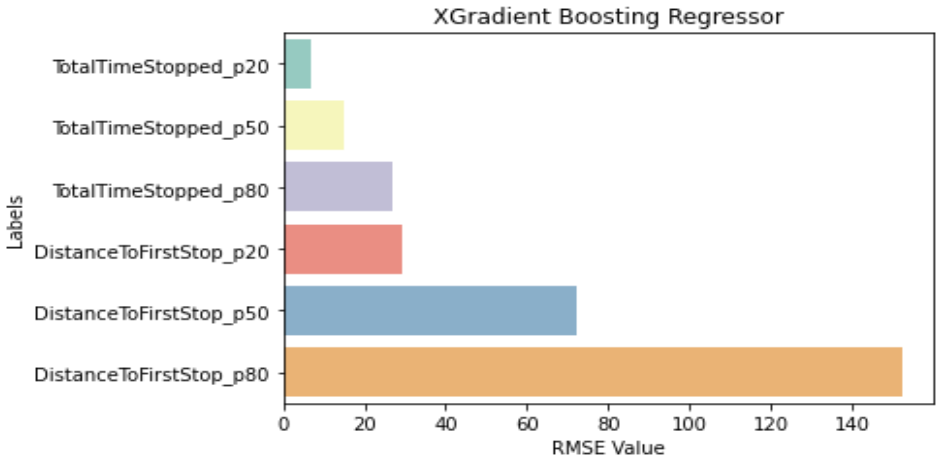


Fig 7: XGB Results

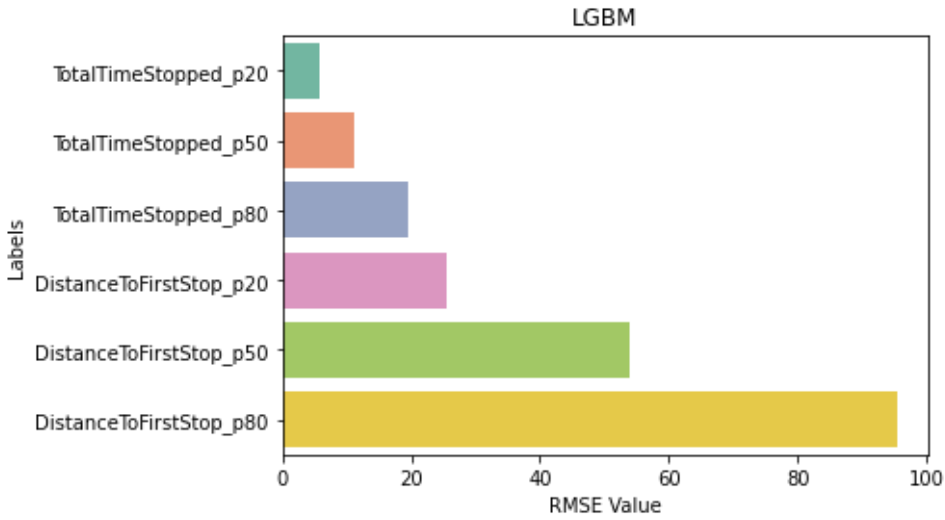


Fig 8: LGBM Results

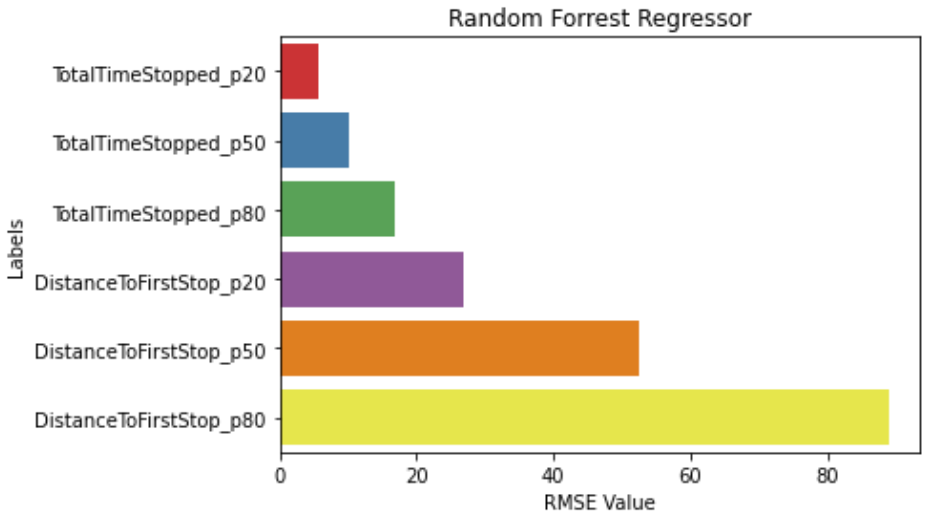


Fig 9: RFR Results

These are the results we got after running the models, as we were doing a kaggle competition we didn't have access to the testing labels so we had to split the training data into testing data to obtain these results the metrics that we have used for the calculation in RMSE Score. We can clearly see from the below result that the Random Forest Tree performance that best compares to other two models. Though it has more training time compared to other best its results are significantly better.

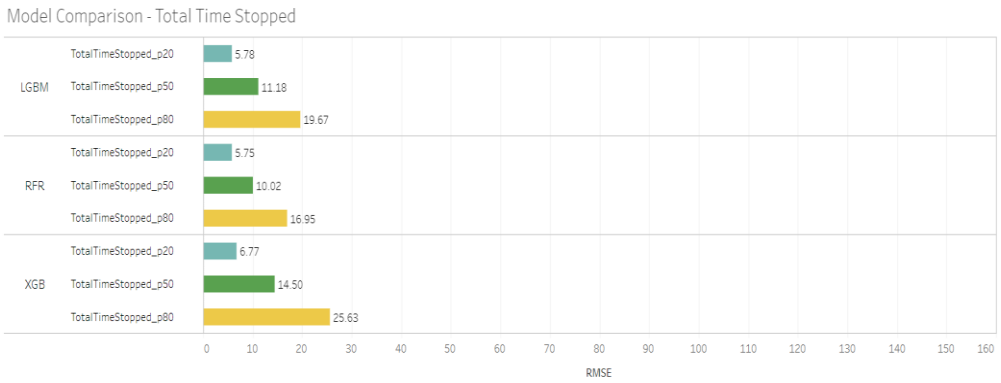


Fig 9: Model Performance Comparison based on Time

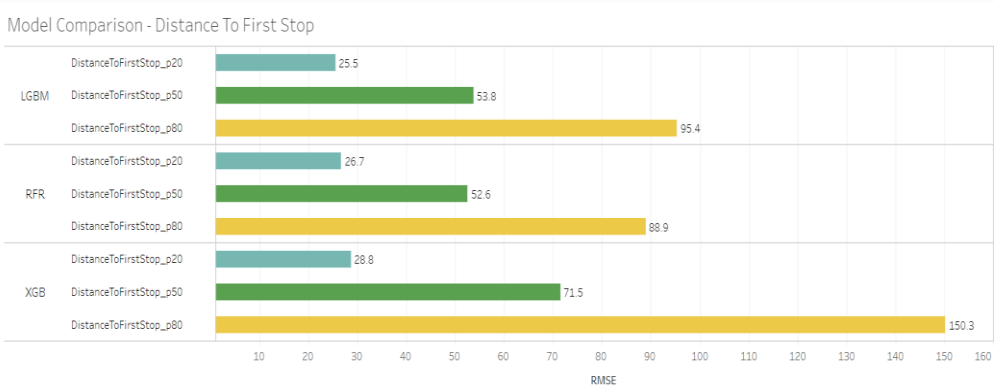


Fig 10: Model Performance Comparison based on Distance

Results

Labels	RFR	XGB	LGBM
DistanceToFirstStop_p20	26.73	28.76	25.54
DistanceToFirstStop_p50	52.56	71.52	53.81
DistanceToFirstStop_p80	88.93	150.25	95.38
TotalTimeStopped_p20	5.75	6.77	5.78
TotalTimeStopped_p50	10.02	14.50	11.18
TotalTimeStopped_p80	16.95	25.63	19.67

Future Work:

We can try to add more features from different datasets like weather data and try to decrease the RMSE score of the model. We have considered 50 n_estimators for Random Forest model due to computational challenges. We can try to increase the number of estimators to yield a better score.

7 Conclusion

The dataset comprised of mainly categorical features. We focussed on important features as we had to predict six different labels dependent on time and distance. We used regression models where the Random Forest Tree gave the best result when compared to other models. LGBM performed better than XGB. Including more auxiliary features like Latitude, Longitude and weather conditions along with fine tuning of parameters in random forest tree in future we can get better RMSE score.

Code Repository

https://github.com/rpm360/MLProject

References

1) Narrowing the Gap: Random Forests in Theory and In Practice. Authors: Misha Denil, David Matheson, Nando de Freitas.
2) Natekin, Alexey & Knoll, Alois. (2013). Gradient Boosting Machines, A Tutorial. Frontiers in neurorobotics.
3) Ridgeway, Greg. (2010). GBM: Generalized Boosted Regression Models.
4) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Authors Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu