New York City Taxi Trip Duration Prediction

Khushboo Agrawal
University of California San Diego
khagrawa@ucsd.edu

Anmol Popli
University of California San Diego
apopli@ucsd.edu

ABSTRACT

In this report, we present an evaluation study of various models and features to predict the total trip duration of a taxi ride in the New York City. This task is implemented on the data released by the NYC Taxi and Limousine Commission. To model the predictive task, an exploratory analysis was done on the data to extract relevant features and utilize them in the prediction model. For this task we have used four models namely: Baseline, Linear Regressor, Gradient Boosting Regressor and Ensemble Linear-Gradient Boosting Regressor. We conclude by showing that the decision tree based Gradient Boosting Regressor performs best with a Root Mean Square Logarithmic Error (RMLSE) of 0.37173 on the test data, ranked among top 100 on the private leader board of the Kaggle competition.

KEYWORDS

NYC Taxi, K-means clustering, Linear Regression, Gradient Boosting Regression

1 INTRODUCTION

Electronic dispatching systems are very useful to keep the public transit running smoothly especially in populated cities like the New York City. Therefore, to improve the efficiency of an electronic taxi dispatching system, it is important to predict the occupancy of the taxi accurately. A significant implication to this prediction is to identify taxi trip assignment for each pickup request. Hence, this prediction task would be beneficial for time management of both passenger and driver as well as result in an efficient city transit system.

To achieve an optimal model for this task, we embark by doing an exploratory analysis on the dataset and engineer features which are relevant to the prediction of the trip duration. We then include the extracted features and train four models namely: Baseline, Linear Regressor, Gradient Boosting regressor, Ensemble Linear-Gradient Boosting Regressor and conclude by sharing an optimal model. This report describes in detail the data used, features used, methods to engineer the features, predictive task, models used, model evaluation by reporting their RMSLE values and improvement ideas as future work. This report also incorporate literature as well as informative plots of various results midway and concluded to complement our discussion.

2 DATA

The data used for this task is released by the NYC taxi and Limousine Commission. The train data and test data comprises of a total of 1458644 and 625134 trips respectively. [2]. Each trip data in the train data records the following information namely: Taxi Id, vendor Id, pickup date time,drop off date time, passenger count, pickup longitude, pickup latitude, drop off longitude, drop off latitude, trip duration.

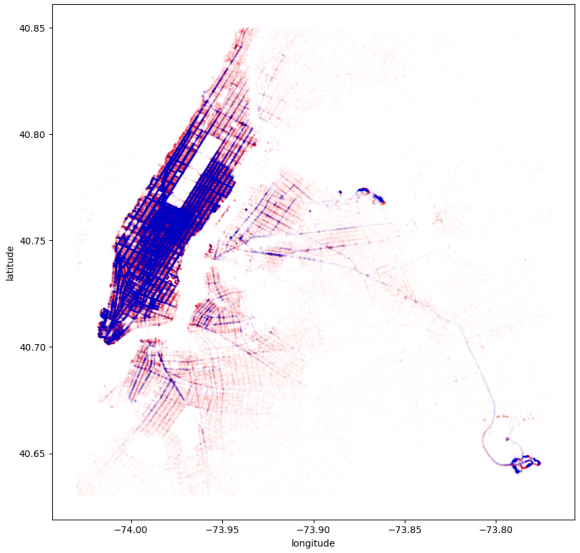

Figure 1: Mapping of pick up and drop off points, where blue points denote pickup locations and red points denote drop off locations

From a preliminary analysis of the data before training the data, we removed the outliers having minor discrepancies like negative trip duration, having trip duration greater than 100 hours, trips originating outside the New York City by the following methods:

- Considering the New York City boundaries: Trips originating or ending inside the city boundary were considered.
- Trip Duration Cleanup: Trip data having trip duration within two standard deviations of the mean trip duration were considered.

After removing the outliers, plot of pickup and drop off coordinates is shown in Figure1, where blue points denote pickup locations and red points denote drop off locations. Additionally, we have also considered open source routing machine (OSRM) data[3] which contains fastest route trip duration, steps and fastest route distance information of each trip data.

2.1 Exploratory Data Analysis

To better understand the task at hand and the features, we perform an exploratory analysis on the data. The purpose of this analysis

is to get insights on various features and how to leverage these features in our models to achieve best possible results. The following features are directly extracted from the data and used for training, while some additional features are engineered using these feature described later in the report.

- Vendor ID: ID of the vendor
- Passenger count: Number of passengers in the taxi
- Pick up longitude: Longitude of the pick up point
- Pick up latitude: Latitude of the pick up point
- Drop off longitude: Longitude of the drop off point
- Drop off latitude: Latitude of the drop off point
- Pick up time: time at pick up
- Fastest route distance: Distance of the fastest route
- Fastest travel time: Trip duration if taken the fastest route
- Number of steps: Each step indicate a straight path before taking a turn
- Pick up weekday: Day of the week

3 FEATURES AND PREDICTIVE TASK

To address one of the important question which would affect the trip duration would be "how does the traffic of taxi rides change along the day?" We would use Mini Batch K-means clustering to cluster the New York City into different groups based on location, and analyze the traffic based on the features engineered on this. For example, one can expect that residential areas would have more incoming traffic in the evening, whereas commercial areas would mostly attract people during the day, areas with rich nightlife would show more traffic in the night and tourist spots will have consistent traffic throughout the day.

We have clustered the map of the New York City into 100 clusters by randomly choosing 500,000 points from a stack of all the trip coordinates and using a batch size of 10,000 for training. For ease of visualization, we have shown 20 clusters and their centers in Figure2 and Figure 3 respectively.

3.1 Feature Engineering

Based on clusters developed, following features were engineered to incorporate traffic which would affect the trip duration.

- Haversine distance: Haversine distance between pick up and drop off coordinates[1].
- Manhattan distance: Manhattan distance between pick up and drop off coordinates.
- Bearing Direction: Bearing direction from pick up coordinate towards drop off coordinate[1].
- Center Longitude and Latitude: Mid point of pick up coordinate and drop off coordinate.
- Pick up and Drop off clusters: Two different feature each representing cluster to which pick up coordinate and drop off coordinate belong.
- Week of year: represents the week of the year.
- Cluster Haversine: Haversine distance between pickup cluster centroid and drop off cluster centroid.
- Cluster Manhattan: Manhattan distance between pickup cluster centroid and drop off cluster centroid.
- Cluster Bearing: Bearing direction from pickup cluster centroid towards drop off cluster centroid.

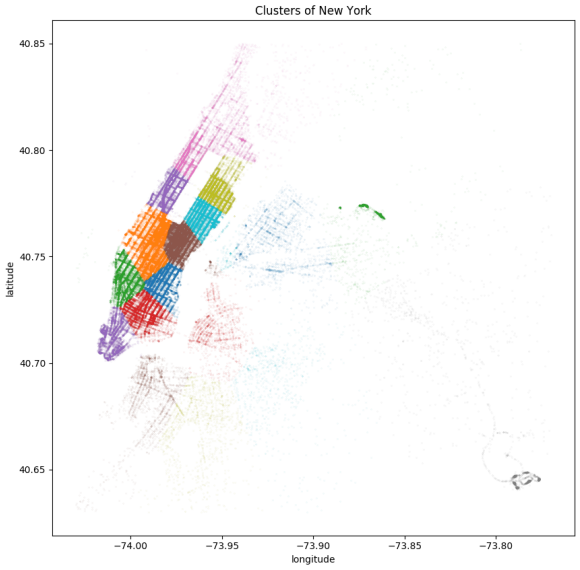

Figure 2: New York City Clusters

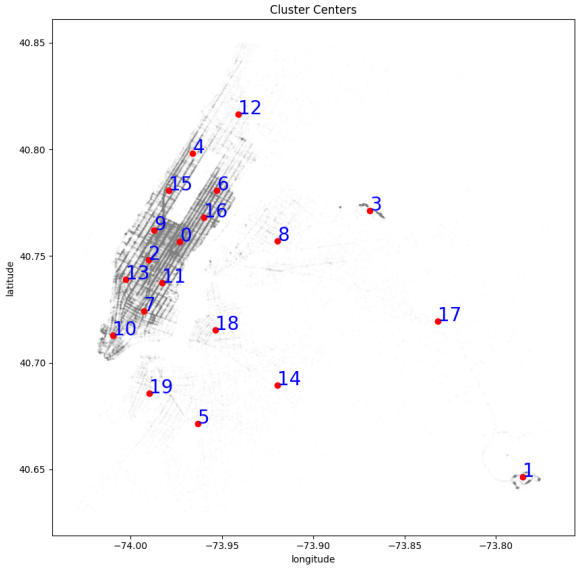

Figure 3: New York City Cluster Centers

- Average speed Haversine pick up date, week hour: Two different feature each representing average haversine speed grouped by pick up date and hour of week.

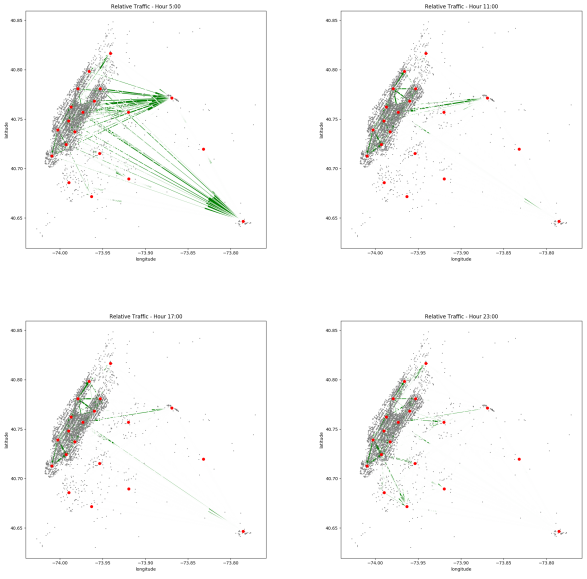

Figure 4: Relative traffic of the New York City at different times of the day

- Average speed Manhattan pick up date,week hour: Two different feature each representing average manhattan speed grouped by pick up date and hour of week.
- Average speed Manhattan pick up and drop off cluster: Two different features representing average manhattan speed for each pick up and drop off cluster.
- Average speed Haversine pick up and drop off cluster: Two different features representing average haversine speed for each pick up and drop off cluster.
- Average speed Haversine pick up hour center: Average haversine speed of center of pick up and drop off coordinate.
- Average speed Haversine pick up hour, pick up and drop off clusters: Two different features representing haversine speed grouped by pick up hour for each pick up and drop off cluster.
- Count pick up hour, pick up and drop off clusters: Two different features representing count of number of trips grouped by pick up hour for each pick up and drop off cluster.
- Count pick up hour center: count of number of trips passing through the mid point of pick up and drop off coordinate grouped by pick up hour.
- Average speed Haversine, pick up to drop off clusters: Average haversine speed from specific pick up to drop off cluster.
- Count pick up to drop off clusters: count of number of trips from specific pick up to drop off cluster.
- Count 60 min: count of trips within 60 minutes in the entire city.
- Count pick up cluster: count of number of trips originating from a pick up cluster over time.
- Count drop off cluster: count of number of trips ending in a drop off cluster over time.

Relative traffic of the New York City at different time is plotted in Figure 4.

3.2 Limitations

Following are the limitations in our predictive modeling task:

- Since the route traversed for each trip is unknown, to incorporate traffic en-route, we consider traffic at the center of pick up and drop off coordinates.
- No driver characteristics (example: gender,age) are given in the data, hence no feature demonstrating driver's characteristics is taken for modeling.

3.3 Predictive Task

Our goal of this task is to predict the trip duration of a taxi trip in the New York City. The predictive task is divided into the following steps mentioned below:

(1) Review data: The data is reviewed, cleaned by removing the outliers and further used for training.
(2) Data Split: The train data is shuffled and split into two disjoint sets of which 80 percent of the data is used for training and 20 percent for validation.
(3) Features: The features described above in the report are extracted from the train data and used on various models for training.
(4) One hot encoding: The following categorical and temporal features are one hot encoded while training the liner model: Trip weekday, Pick up hour of the day, Pick up cluster, Drop off cluster, Trip week of the year.
(5) Models: Four models namely: Baseline, Linear Regressor, Gradient Boosting Regressor, Ensemble Linear-Gradient Boosting Regressor are trained which are described in detail in the following report.
(6) Model Evaluation: The model is evaluated by calculating the Root Mean Square Logarithmic Error(RMSLE) given by the equation below where p_i is the prediction value and a_i is the actual value and N is the sample count:

$$RMSLE = \sqrt{\frac{\sum_{i=1}^{N}(\log(p_i + 1) - \log(a_i + 1))^2}{N}} \quad (1)$$

4 MODELS

The models used for this problem are described in detail below:

4.1 Baseline Model

The range of the trip duration in the data set is very high since the data is recorded in seconds, therefore we scale the trip duration by taking logarithm and use the *logarithmic trip duration* defined below to evaluate the model using RMSLE (1).

logarithmic trip duration= $\log(t + 1)$: t is in seconds

Figure 5 shows the plot of logarithmic trip duration and the number of trips. For the baseline model, we predict the logarithmic trip duration of incoming trip data as the mean value of all the logarithmic trip duration of all the trips in the training data.

This results in an RMSLE of 0.76573658 on the validation data set. We then proceed by training regression models.

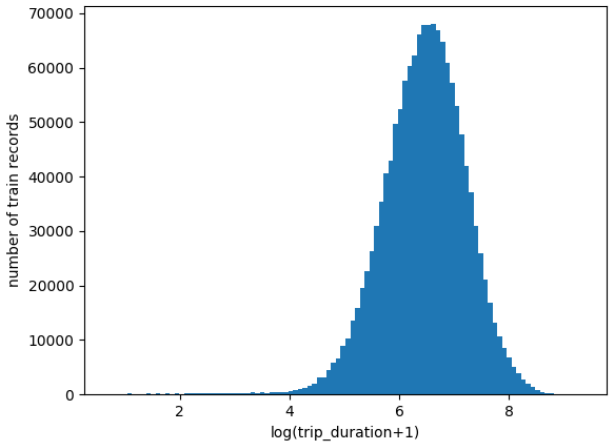

Figure 5: Distribution of Logarithmic Trip Duration

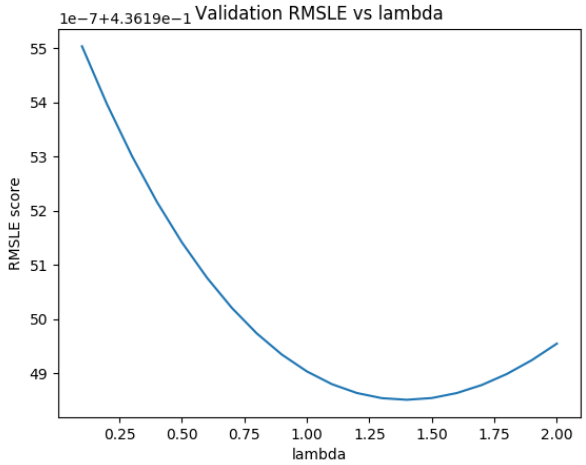

Figure 6: Validation RMSLE vs λ

4.2 Linear Regression

Initially we train the model using regularized linear regression, where we have modelled the logarithmic trip duration as a linear function of all the extracted features. We have implemented linear regression using the Scikit-learn's Ridge Regressor as it regularize the model using l2-norm which helps to avoid over fitting on the training set.

$$\textit{Loss function: } \|Ax - b\|^2 + \lambda\|x\|^2$$

Before training the model with Ridge Regressor, the following categorical and temporal features are one hot encoded: Trip weekday, Pick up hour of the day, Pick up cluster, Drop off cluster, Trip week of the year. We further optimize the regularization hyper parameter λ so as to maximize the performance on the validation set. Figure 6 shows a plot of validation RMSLE and λ.

Significant from the graph, the optimal value of λ is 1.4 which results in an RMSLE of 0.43619485 on the validation set.

4.3 Gradient Boosting Regression

Since traffic is not varying solely based on the magnitude of the coordinates, the linear models fail to account the nonlinear effect of the location of pick up and drop off coordinates. Therefore, to take into account the non-linearity, we train our model using Gradient Boosting Regression.

Gradient Boosting regression builds an additive model in a forward stage wise fashion by allowing for the optimization of differentiable loss functions. At each stage, a regression tree is fit on the negative gradient of the least square regression loss function.

We train a gradient boosting regressor, employing decision tree model as the base learner and modeling the logarithmic trip duration as a function of all extracted features. An advantage of using this algorithm is that it takes care of all the categorical and temporal features therefore, one hot encoding is not required. Gradient Boosting Regressor requires a number of hyper parameters to be optimized, namely *min_child_weight*, *eta*, *colsample_bytree*, *max_depth*, *subsample*, *lambda*.

We optimize the aforementioned hyper parameters through three levels of random search by fine tuning the parameters at each advancing level. At the end, we arrive at optimal values of the hyper parameters such that the reduction in RMSLE on validation set with further change in hyper parameters is not significant. To avoid overfitting on the training data we also implement early stopping, wherein validation error needs to decrease at least every 50 boosting iterations to continue training.

The optimal values of hyperparameters obtained through our random search are as follows: $min\_child\_weight = 10$, $eta = 0.11$, $colsample\_bytree = 0.5$, $max\_depth = 15$, $subsample = 0.9$, $lambda = 1$. The aforementioned values resulted in an RMSLE of 0.325 on the validation set. Figure 7 shows a bar graph of the relative importance scores of various features, obtained from the trained Gradient Boosting Regressor model.

4.4 Ensemble of Linear and Gradient Boosting Regressor

In this model, we consider the mean of the predicted values that were obtained using the linear regression model and the gradient boosting regression model and, further use them to compute the RMSLE on the validation set. The RMLSE on the validation set is 0.35340976 when trained using this ensemble regressor.

5 LITERATURE

- The data sets used for training are taken from the Kaggle website [2],[3]. The primary data set [2] has been released by by the NYC Taxi and Limousine Commission, which includes pickup time, geo-coordinates, number of passengers, and several other variables. We have also used OSRM fastest route data set [3] generated from Open Source Routing Machine.
- [1] presents a variety of calculations for latitude/longitude points, with formulae and code fragments for implementing them. From here, we use the *haversine* formula to calculate the great-circle distance between two points - that is, the

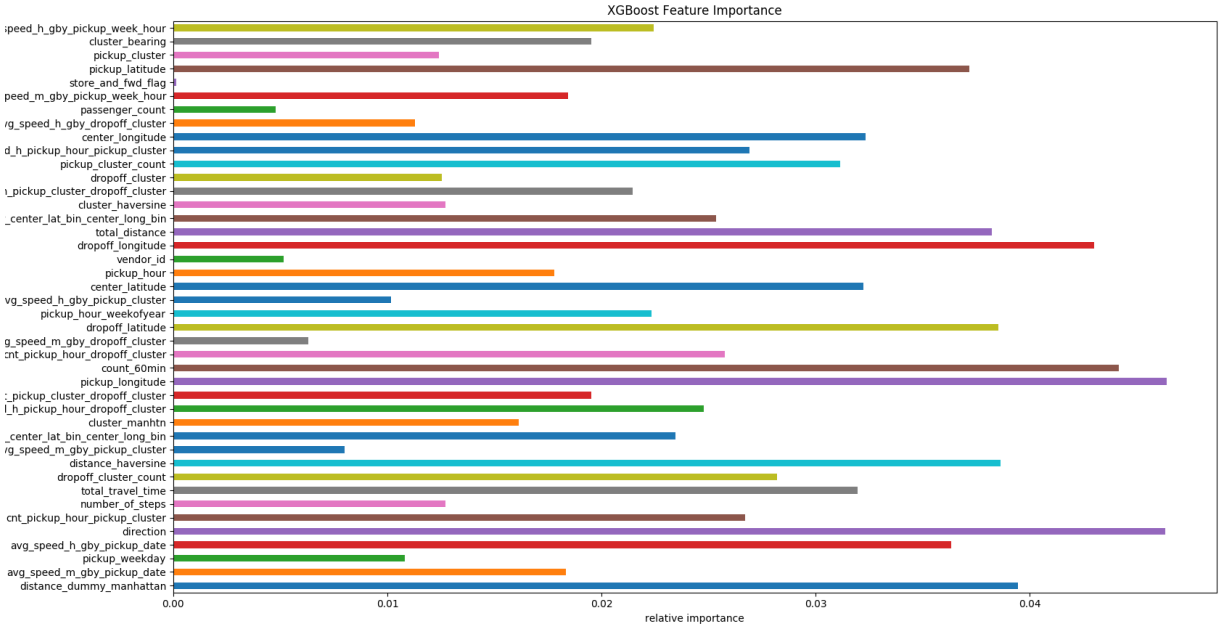

Figure 7: Relative importance scores of various features used to train GB Regressor

shortest distance over the earth's surface - giving an *as-the-crow-flies* distance between the points.

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi 1 \times \cos\varphi 2 \times \sin^2(\Delta\lambda/2)$$
$$c = 2 \times \arctan 2(\sqrt{a}, \sqrt{1-a})$$
$$d = R \times c$$

where φ is latitude, λ is longitude, R is earthâĂŹs radius

We use the following formula for the initial bearing (sometimes referred to as forward azimuth) which if followed in a straight line along a great-circle arc will take you from the start point to the end point.

$$\theta = \arctan 2(\sin\Delta\lambda \times \cos\varphi 2, \cos\varphi 1 \times \sin\varphi 2 - \sin\varphi 1 \times \cos\varphi 2 \times \cos\Delta\lambda)$$

where $\varphi 1, \lambda 1$ is the start point, $\varphi 2, \lambda 2$ the end point ($\Delta\lambda$ is the difference in longitude)

Survey and comparative analysis of other works that have used the NYC Taxi Trip data set for predictive task are as follows:

- [7] uses Linear Regression and Random Forest model to predict the duration of taxi trips in NYC. Most of the streets and avenues in Manhattan are aligned in a grid structure. With the hypothesis that the avenue or street could explain some of the effect of the location, they transform the coordinates using Principal Component Analysis. In our model, we have used additional features based on pickup and drop off clusters which help in modeling the effect of the location (example: community, landmark), not employed in their study.
- [9] In this report, the authors have discussed the importance of various features based on K-means clustering of the New York City map. They have used decision tree based models like Gradient boosting regression and Random Forest to predict fare and trip duration. We have adopted the idea of using decision tree based models and appended many several cluster based features to train our model. Moreover, we have also taken into consideration the ensemble linear and gradient boosting regression.
- [10] proposes a model that allows users to visually query taxi trips. Besides standard analysis queries, the model supports origin-destination queries that enable the study of mobility across the city. This model is able to express a wide range of spatio-temporal queries, and it is also flexible in that not only can queries be composed but also different aggregations and visual representations can be applied, allowing users to explore and compare results.
- [11] the authors tackle the problem by using data from from buses (GPS) and an algorithm based on Kalman filters for prediction of the trip duration. Using a similar approach, [6] uses real time data from smart phone placed inside vehicles.
- In [8] the authors use a combination of traffic modelling, real time data analysis and traffic history to predict travel time in congested freeways. They try to overcome the assumption that real time analysis communication is instantaneous. A lot of other papers also work on freeways. In [4] the prediction is done using Support Vector Regression (SVR) while in [5] Neural Networks (SSNN) are used.

6 OBSERVATIONS AND RESULTS

6.1 Observations from Clustering

We obtain some pretty interesting insights when we plot our clusters over the map of New York (Figure 2). We observe that our resulting clusters are representative of the way NYC is divided into different neighborhoods. In Figure 2, we can see Upper East and West side of Central park in blue and pink respectively, Chelsea and West Village in dark blue, East Village and SoHo in yellowish green. The airports JFK and La LaGuardia have there own clusters, and so do Queens and Harlem. From Figure 4, we can see that in the morning most of the traffic is in Manhattan island. The share of taxis travelling to Brooklyn area, mostly Williamsburg, becomes much larger in the late evening. Since there's no similar movement in the morning hours (in the opposite direction), this is unlikely to be the result of commuting to work. Instead, and since the traffic is mostly seen later in the night, these are probably people travelling out of the city. In the very early morning, most of the traffic is to and from the two airports.

Clustering gives us many useful pieces of information. For example, if we know pickup is in cluster 3 and drop off is in cluster 10, and maximum traffic flow is from cluster 3 to cluster 10, this indicates that the trip duration is going to be more than otherwise. We model this behaviour through features like *count of trips between origin and destination clusters*, *average speed between origin and destination clusters* among others. And that is also the way most cab services decides surge pricing. If the traffic flow from cluster X to cluster Y is more than that from cluster X to cluster Z, the surge will be more for a trip from cluster X to cluster Y.

6.2 Results of the Trained Prediction Models

After training the various models on the extracted features, we obtained following results on the validation set:

Model	RMLSE
Baseline	0.7657365
Linear Regression	0.4361948
Gradient Boosting Regression	0.3250000
Ensemble Linear-GB Regression	0.3534097

Table 1: Models and their RMLSE scores on the validation set

- Gradient Boosting Regression performs best among all the other models on the validation set with an RMLSE value as low as 0.325.
- We observed that features engineered based on clustering play a profound role in the prediction model. Figure 7 shows the relative importance of each feature while training the GB reressor model.
- Since the best performance on validation set is achieved by Gradient Boosting model, we implement it on Kaggle's test dataset. On submitting our results to the Kaggle competition, we attain an RMSLE score of *0.37173*, which ranks among the top *100* of its Private Leaderboard (out of a total of 1257 participants).

7 CONCLUSION AND FUTURE WORK

From our study for the predictive task we conclude and propose the following:

- The score shows that our developed method has achieved competitive performance on Kaggle's test dataset for New York City Taxi Trip Duration prediction.
- Gradient Boosting model is a fairly good model to predict the trip duration of the taxi ride considering no real time data.
- We can further enhance the same model, if we can gather data of driver characteristics (example: gender, age) and also the route taken.
- We can further improve the same model if we have all year round data, as it can include the features corresponding to holiday and different seasons.
- This model can be clubbed with real time data (GPS) and algorithm such as Kalman Filtering can be applied for better results.
- This model can also be clubbed with real time weather data for traffic prediction resulting in improved prediction of the trip duration.

8 ACKNOWLEDGEMENT

The authors would like to thank Prof. Julian McAuley for giving us the opportunity to learn and apply data mining and machine learning techniques through this project.

REFERENCES

[1] 2002-2017. "Haversine distance and Bearing between latitude/longitude points". https://www.movable-type.co.uk/scripts/latlong.html
[2] 2017. "New York City Taxi Trip Duration". https://www.kaggle.com/c/nyc-taxi-trip-duration/data
[3] 2017. "New York City Taxi with OSRM". https://www.kaggle.com/oscarleo/new-york-city-taxi-with-osrm
[4] Wu, Chun-Hsin, Jan-Ming Ho, and Der-Tsai Lee. 2004. "Travel-time prediction with support vector regression". (December 2004).
[5] Van Lint, J. W. C., S. P. Hoogendoorn, and Henk J. van Zuylen. 2005. "Accurate freeway travel time prediction with state-space neural networks under missing data". (December 2005).
[6] Biagioni, James. 2011. "Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones". (December 2011).
[7] Christophoros Antoniades, Delara Fadavi, Antoine Foba Amon Jr. 2016. "Fare and Duration Prediction: A Study of New York City Taxi Rides". (December 2016).
[8] Yildirimoglu, Mehmet, and Nikolas Geroliminis. 2013. "Experienced travel time prediction for congested freeways. Transportation Research Part B: Methodological 53". (December 2013).
[9] Jaiswal, Bansal, Jakate, Saxena. 2017. "NYC Taxi Rides: Fare and Duration Prediction". (2017).
[10] Jorge Poco, Huy T. Vo, Juliana Freire, Cláudio T. Silva. 2013. "Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips. Nivan Ferreira". (January 2013).
[11] Vanajakshi, L., S. C. Subramanian, and R. Sivanandan. 2009. "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses". (December 2009).