TAXI TRIP DURATION PREDICTION.

## ABSTRACT.

In building modern intelligent transportation systems like taxi or ride-sharing apps, the accurate prediction of travel time is irreplaceable. It can not only improve the customers’ experience on traveling, but also help taxi drivers manage their routes and orders in a more efficient way. This problem is challenging mainly due to its large dataset and the complex relationship between the

model and features. By using the New York City’s taxi record in 2017, this paper investigates several machine learning methods to predict the travel time. Since the number of the original feature is small, we introduce several external features that improve the performance of the models. Finally, we present evaluation results of the proposed method and ablation study on the design of the proposed model.

## INTRODUCTION.

#### Problem Formulation.

Given the pick-up and drop-off location, our goal is to predict the travel duration based on the dataset X that contains the taxi travel records of the New York City. Other information may also be introduced to dataset X eg. Pickup weekday.

#### Brief Summary

Nowadays, ride-sharing apps are playing an increasingly important role in people’s lives. It is especially important for customers and drivers to predict travel time in advance. Furthermore, the predicted travel time is also useful for city planning. This problem is challenging because we need to find the appropriate features that have great impact on the travel time, which is not covered in the original dataset. Also, Since the dataset is very large, we need to explore some practical way to deal with it. In the following section, we describe our learning model and feature engineering method specifically. In this paper, our main focus is on finding better learning model and useful external features. To extract valuable features from the raw dataset, the original data have been engineered to be new features in the learning model. We proposed several different models and introduced some new features. By comparing among all the models, we figure out our best model is the Gradient Boosted Tree. Also, the ablation test on features shows that the holiday has the most impact on our model. The rest of this paper is organized as follows. In Section 2, we discuss background and related work. We introduce our data and make a visualization in Section 3. In Section 4, we discuss the method that has been used for solving this problem, which covers our

model and feature engineering process. We evaluate our model and show the results. Finally, we make a conclusion in Section 7 .

# Background and Related works.

To predict the travel time, two different methods are frequently used. The first one is path-based method and the second one is origin-destination method. Since path-based method requires the whole route information of the travel, it is not applicable in our problem. So we used the origin- destination method, which is more relative to our raw dataset.

The main advantage of the origin-destination method is that it saves the time for calculating and estimating the travel path. Recently some researchers has used this method to predict travel time. Wang et al. (2016) introduced a nearest neighbor approach to the origin-destination method. In their approach, the travel time is predicted by calculating the average travel time of similar origin & destination location. Jindal et al. (2017) proposed a neural network to jointly estimate travel time and distance, but this method fails to capture spatial relationship between regions. Recently, Li et al. (2018) proposed a graph embedding learning method to capture the road links information.

However, lacking the road links and trajectory information in the dataset, this method is not applicable to this task.

Compared to the path-based methods, the drawback of origin-destination methods is , instead of using the route information thoroughly, the origin-destination method ignores the underlying information about the road network. So the prediction result may not be very accurate.

# Dataset.

The dataset we used in this project is approximately 60 million taxi rides recorded by New York City taxicabs in 2017. The starting and ending locations are pre-processed by discretizing the region and reporting the index of the starting and ending sub-regions. Thus we do not have the exact pickup and drop-off latitudes and longitudes. The original dataset contains vendor ID (integer), passenger counts, pickup and drop-off date times (string), pickup and drop-off region ID (integer), and payment methods (integer). The dataset contains around 0.5% outliers, the duration of which is negative or larger than 8 hours. We dropped out these outliers in the dataset. We did not process beyond the given data or combine the given data with other datasets, but we did add more features.

# Data Visualization.

In data visualization with help of matplotlib and seaborn , we have plotted a graph based on univariate and bivariate analysis .

In Univariate analysis we have done the analysis like ,

* + No. of Passenger.
  + Trip duration
  + Speed
  + Distance
  + Vendor ID.

In Bivariate analysis we have done analysis .

* + Trip Duration per Hour
  + Trip duration per Day
  + Trip duration per Weekday
  + Trip duration per Month.
  + Trip duration per Vendor.

Their were many Outliers we cleaned the outliers .

# Methods.

### Feature Engineering.

We first appended the latitude and longitude of the centroid that represent each region that each record belongs to as features of the dataset. Then we parsed the date time string as a date time object and extracted year, month, day, weekday, hour and minute from the date and time of each ride as individual features. We calculated the duration of each ride by subtracting the drop-off time from pickup time, and dividing by 60 to obtain the trip duration in minute.

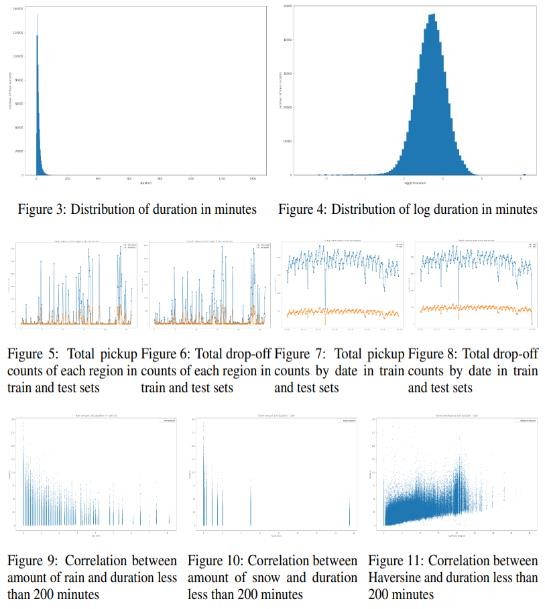
Based on location coordinates, we calculated Haversine distance, Manhattan distances as well as driving directions.

### Feature Analysis.

We split the dataset into training and testing sets with a ratio of 0.8 and 0.2 simply for the purpose of feature analysis.

We plot the distribution of trip and logged duration as shown in Figure 3 and 4. As we can see, Figure 3 indicates a right skewed distribution while the logged duration is a normal distribution. In Figure 5 and Figure 6, we showed that the training and testing sets have a roughly similar shape of distribution on total pickup as well as drop-off counts, i.e., a popular pickup location in training set is also a popular pickup location in test set, proving the randomness of data selection. By comparing both graphs, we can also infer that a popular pickup location is very likely a popular drop-off location. Figure 7 and Figure 8, on the other hands, show the total pickup and drop-off counts by date. Both graphs also show a similar pattern in train and data sets, as well as a high popularity correspondence between pickup dates and drop-off dates. To better analyse the correlations between different features and duration, we focused on the duration less than 200 minutes. Figure 9, Figure 10, and Figure 11 show the correlation between duration and rain, snow, Haversine distance respectively. The Haversine distance is calculated according to equation (1), where r represents the average earth radius (6371 km ).

This is possibly because during bad weather, people tend to not travel very far by taxi. Figure 11, on the other hand, shows a roughly positive correlation between distance and duration, which is reasonable.

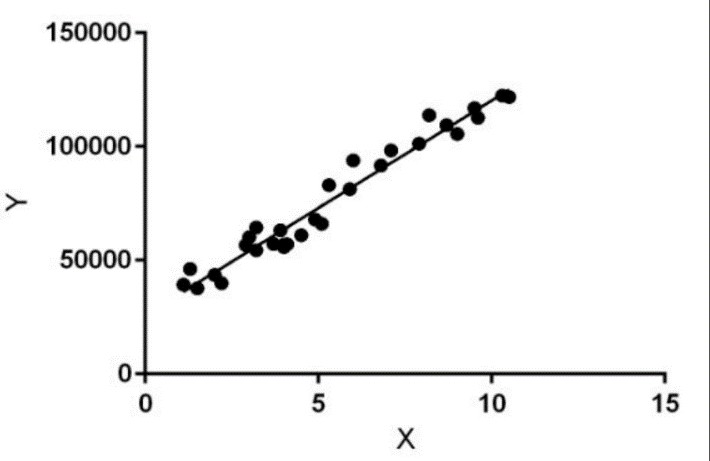


# Model.

#### LINEAR REGRESSION MODEL.

**Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs

a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.



#### XGB BOOSTING.

XGBoost (Extreme Gradient Boosting) is an optimized distributed gradient boosting library. It uses gradient boosting (GBM) framework at core. It belongs to a family of boosting algorithms that convert weak learners into strong learners. A weak learner is one which is slightly better than random guessing.

'Boosting' here is a sequential process; i.e., trees are grown using the information from a previously grown tree one after the other. This process slowly learns from data and tries to improve its prediction in the subsequent iterations.

#### Model training.

We will train the model on the filtered features. Our data has already been split so we will not split the data further.

We used **GridSearch** to tune the **hyperparameters** of XGBoost regressor to get the best possible test score. We will compare results from the default regressor and the tuned regressor.

#### Observations

* There is a significant **improvement** in the RMSE score for the **tuned** XGBoost regressor over the Random forest regressor when trained on the feature selection group.
* But the performance of the **default** XGBoost regressor is quite **worse** than the default RF regressor on the same data.
* Also, the RMSE score on the raw data and feature selected data are same, which disproves the theory that it is always better to select the relevant features which are statistically important. As the data behaves differently in different models.
* Not to mention the fact that RMSE score for the XGBoost regressor of the feature extraction group is still bad along with the variance score.

# CONCLUSION.

* We compared MSE, RMSE and R2 for all four regression models, to find which is the best model to predict the NYC taxi trip duration.
* The Linear Regression and Lasso Regression didn't show any good prediction as compared to the other two.
* From above comparison table we can clearly see that **XGBoost** and **LightGBM** are the best models to predict trip duration of the NYC taxi. While **LightGBM** is fastest and more accurate than **XGBoost**. So, in between these two **LightGBM** is the best model.
* **R Square:** R2 score represents the coefficient of how well the values fit compared to the original values. The value from 0 to 1 interpreted as percentages. The higher the value is, the better the model is.
* **RMSE (Root Mean Squared Error):** RMSE is the standard deviation of the residuals. Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it displays how concentrated the data is around the line of best fit.