**New York City Taxi Trip Duration Prediction**

**Introduction**

In New York City, due to traffic jams, construction or road blockage etc. user will need to know how much time it will take to commute from one place to other. Increasing popularity of app-based taxi such as Ola or Uber and there competitive pricing levels made user decisive to choose based on trip pricing and duration. Taxi Drivers also have to choose best route having lesser trip time. Here we build a model which predicts the trip duration of taxies running in New York. This prediction will help customers to select the taxi based on trip duration and driver to select optimum route to their destination.

**Problem Statement**

We have the dataset based on the first six months of 2016 NYC Yellow Cab trip record data and it is available on Kaggle website. The data was originally published by the NYC Taxi and Limousine Commission (TLC). The data was sampled and cleaned for the purposes of this project. Based on individual trip attributes, we predict the duration of each trip in the test set.

**Additional Dataset**

The taxi trip duration dataset has winter time values. The precipitation in winter could affect the trip time duration. Therefore, we use daily weather data at central park and calculate the amount of precipitation which is the sum of snow and rain for each day and define a new categorical feature which is if there was precipitation in each day. The central park weather data is available at Kaggle website. In some columns instead of values there is a T which means there is a trace of precipitation or snowfall. We convert them to 0.05 (a small number), so we can calculate the desired features.

**Goal:**

Goal of this study is develop a model that predicts the trip duration of taxies in NY. There are four steps to rich this goal:

1. Data Exploration
2. Exploratory Data Analysis (EDA)
3. Feature Engineering and Feature Selection
4. Building and Evaluating Model

**Data Exploration and Data Cleaning**

Data is available from January to July 2016 with several features like pick up and drop off latitude and longitude which is used to calculate distance, the number of passengers in the vehicle, and trip duration assumed as a dependable variable that we try to predict. List of columns are as follows:

* id: a unique identifier for each trip
* vendor\_id: a code indicating the provider associated with the trip record
* pickup\_datetime: date and time when the meter was engaged
* dropoff\_datetime: date and time when the meter was disengaged
* passenger\_count: the number of passengers in the vehicle (driver entered value)
* pickup\_longitude: the longitude where the meter was engaged
* pickup\_latitude: the latitude where the meter was engaged
* dropoff\_longitude: the longitude where the meter was disengaged
* dropoff\_latitude: the latitude where the meter was disengaged
* store\_and\_fwd\_flag: This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server - Y=store and forward; N=not a store and forward trip
* trip\_duration: duration of the trip in seconds

**Missing values and Duplicates**

This dataset does not have any missing values or duplicates.

**Outliers**

Some taxi trip durations are too short or too long like 2 seconds or 24 hours. We consider taxi trips less than 10 seconds and more than 20 hours as outliers and remove them. Where the number of passengers is nine also is removed as outliers.

**Data Format**

We convert dates into a common format and extract necessary information like day of the week, month, and hour. Also convert vendor\_id into string.

**Exploratory Data Analysis (EDA)**

First we check the distribution of log transformation of trip duration (Figure 1). It has a normal distribution and the mean is a between 6 and 7.

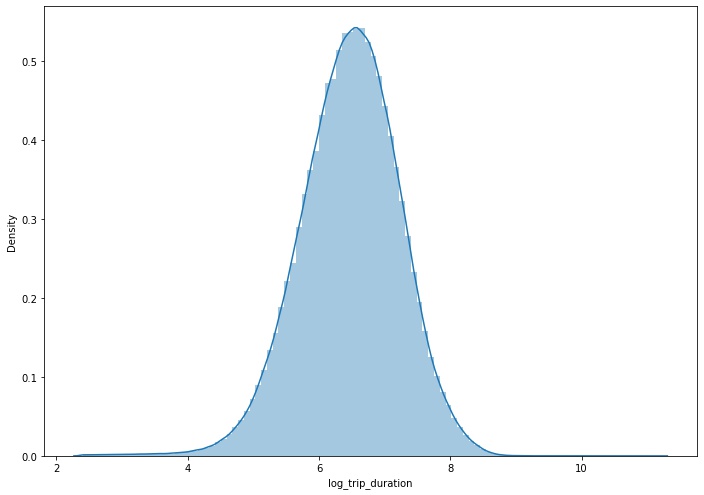


Figure 1. Distribution of log transformation of trip duration

Vendor\_id has two values: 1 and 2 first we check number each vendor\_id and try to separate them for other features in the plots to understand if there is a difference between vendors. Figure 2 shows the number of vendors. Vendor 2 has more rides than vendor 1.

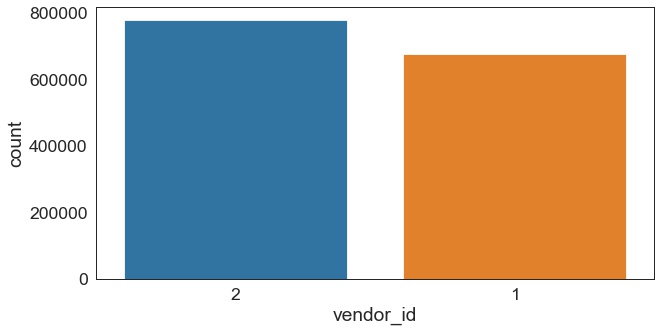


Figure 2. Number of trips for each vendor

We also investigate the number of passengers for total trips and separate them for each vendor\_id (Figure 3). Fig .3 shows that the vast majority of rides had only a single passenger, with two passengers being the (distant) second most popular option. Towards larger passenger numbers there is a smooth decline through 3 to 4, until the larger crowds (and larger cars) give us another peak at 5 to 6 passengers. Except one passenger, vendor 2 has more trips than vender 1.

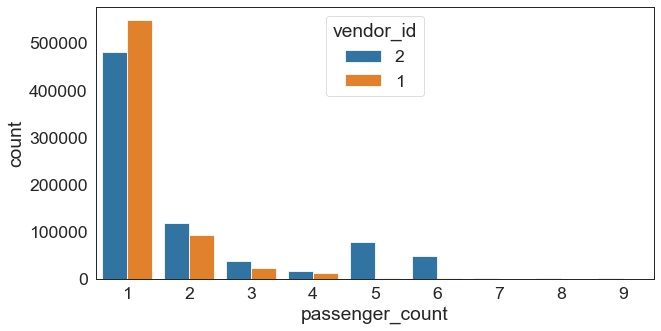


Figure 3. Number of passengers in total trips

To consider the vender\_id as a potential feature for the rest of the plots we show the values for vendors separately. Figure 4 shows the number of trips for each day of the week. Vendor 2 has significantly more trips in this data set than vendor 1. This is true for every day of the week. Monday is the quietest day and Friday is the busiest. This is the same for the two different vendors.

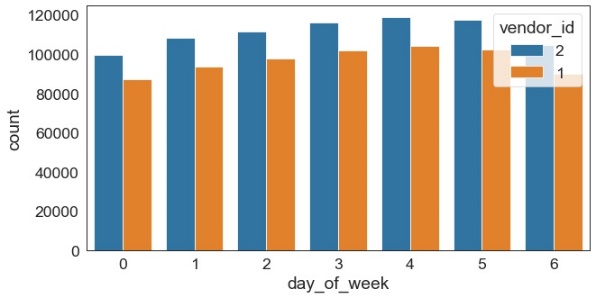


Figure 4. Number of trips for each day of the week

Figure 5 shows the number of trips for each hour. The number of trips are higher in the evening after 5PM until 20 PM and it has the lowest value at 5 AM.

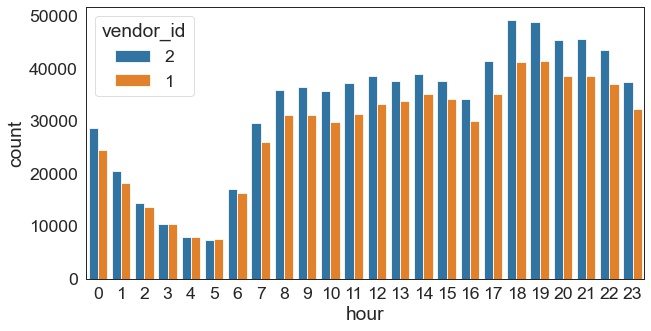


Figure 5. Number of trips for 24 hours

To investigate more we plot number of trips for 24 hours for each day of the week (Figure 6). The weekend (Sat and Sun, plus Fri to an extend) have higher trip numbers during the early morning hours but lower ones in the morning between 5 and 10, which can most likely be attributed to the contrast between NYC business days and weekend night life. In addition, trip numbers drop on a Sunday evening/night.

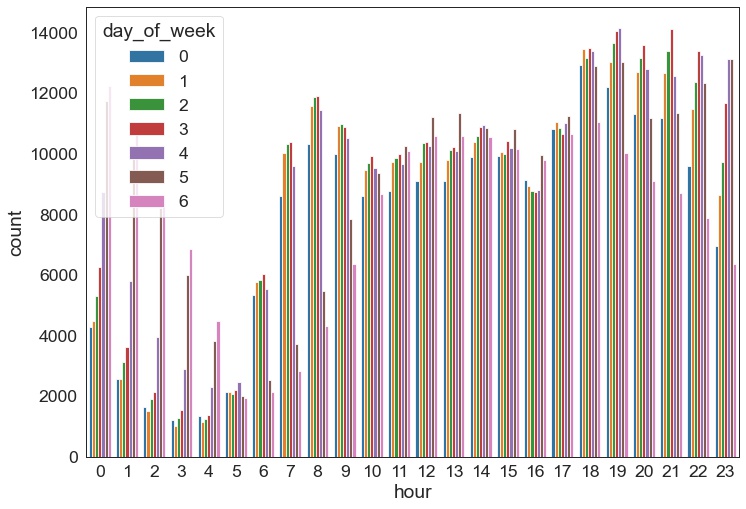


Figure 6. Number of trips for 24 hours for each day of the week

In addition, we plot Number of trips for 24 hours for each month, January and June have fewer trips, whereas March and April are busier months (Figure 7).

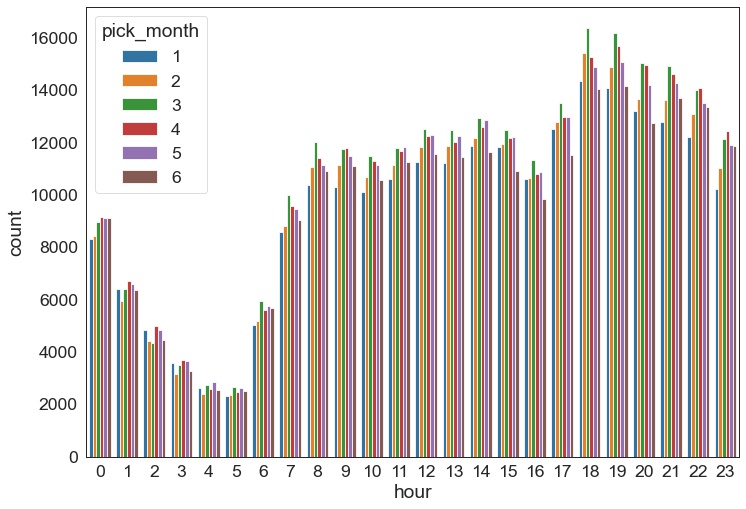


Figure 7. Number of trips for 24 hours for each month

The weekday and hour of a trip appear to be important features for predicting its duration and should be included in the model.

We also look at the number of trips for 24 hours separated by having precipitation or not (Figure 8). The days that there is no precipitation the number of trips are significantly higher.

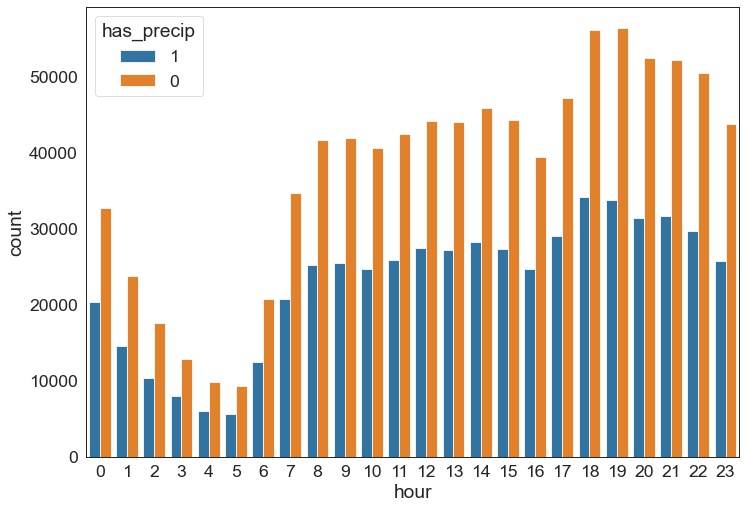


Figure 8. Number of trips for 24 hours separated by precipitation

To understand that number of passengers affects the trip duration we look at the boxplots trip durations for number of passengers and separated them by vendor\_id (Figure 9). There is no significant difference for time duration for different number of passengers and vendors, except for 7 passengers in vendor 2.

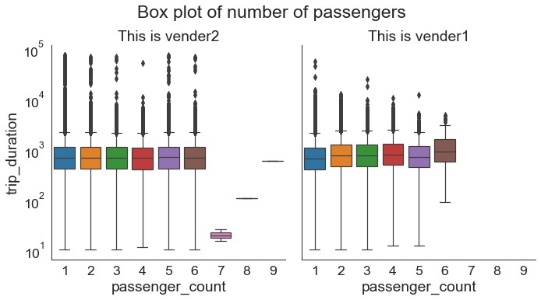


Figure 9. Trip duration for each number of passengers

Figure 10 is showing the pick-up latitude and longitude for both train and test sets. They overlap completely, so it is possible to predict trip duration for test set. pickup\_longtitude, pickup\_latitude, dropoff\_longtitude, dropoff\_latitude have some values which dominate because in this problem we focus on NYC.

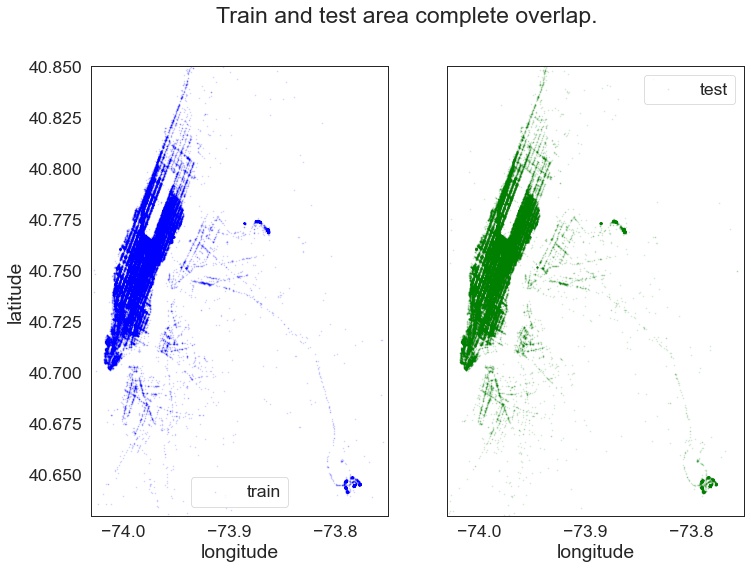


Figure 10. Pick-up latitude and longitude for train and test set.

We calculate average speeds based on distance and trip duration and plot them versus 24 hours and week days (Figure 11). Because at 5am less people using Taxi so the street won't be crowded speed will be high. The opposite is true for hours around 16-20pm. On Monday and Sunday, the speed is higher than other days.

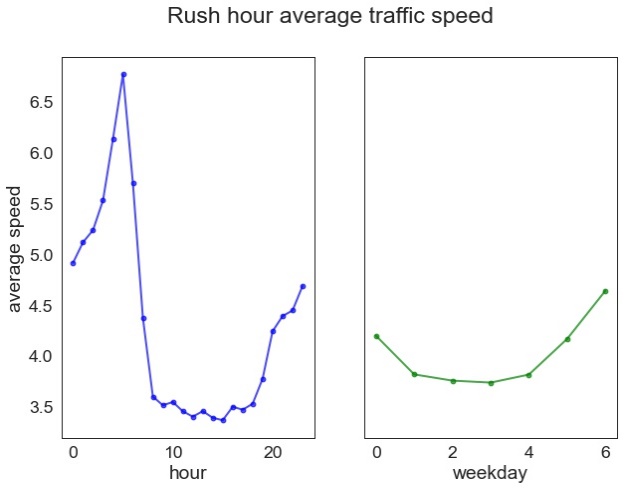
****

Figure 11. Average traffic speed for 24 hours and wee days.

Finally after cleaning and removing outliers we look at the correlation between features and time duration. The strongest correlations (~0.3) are between distances (Manhattan and Haversine) and trip duration (Figure 12).

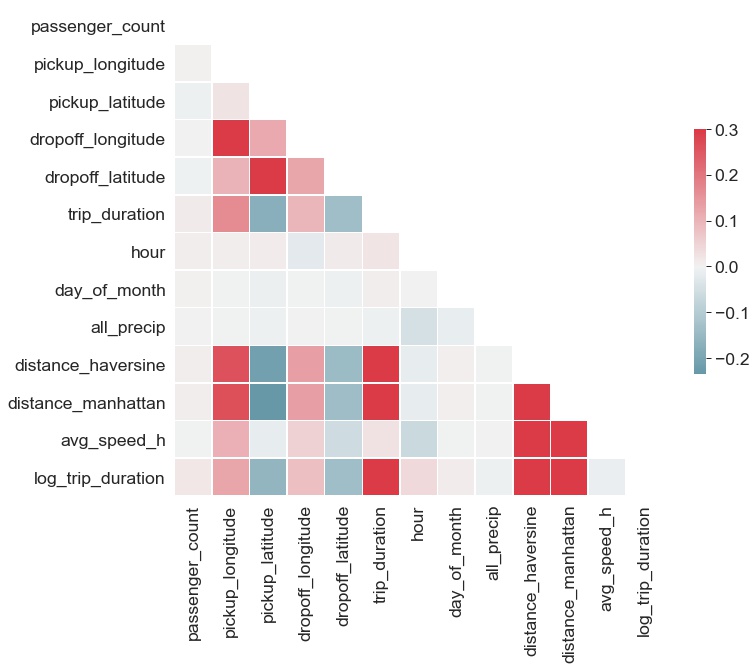


Figure 12. Correlation heat map between features and trip time duration

**Feature Engineering and Feature Selection**

The effect of each column on predicting trip\_duration:

id: id of trip => does not related to trip duration => we do not use to predict trip duration.

passenger\_count: the number of passengers does not indicate the speed of that taxi => we test to use or not to use to predict trip duration.

Features from pickup\_datetime: pick\_month, hour, day\_of\_month, day\_of\_week: this is the basic information of time => Based on those, we can know on which time, the number of vehicles will increase. We do not add minute column because this column is too detail or that can be known by other previous added columns => Feed to model.

avg\_speed\_h: do not use these columns to predict the trip duration because they are calculated based on the target column.

**Features rejected**: id, trip\_duration, avg\_speed\_h.

**Features used**:

vendor\_id,

pick-up month, hour, day of month, and day of week,

distance\_haversine, distance\_manhattan

pickup\_longitude, pickup\_latitude, dropoff\_longitude, dropoff\_latitude

passenger\_count

**Added Features**

EDA shows that precipitation could be feature for trip duration prediction also trip durations are different during week days and weekends. In addition, average speed is higher at early morning and late evening. Correlation plot shows the strongest correlation is between distance and trip duration. Therefore we added four features to the existing list of features:

1. Amount of precipitation (numerical feature)
2. If there is precipitation in each day (categorical feature)
3. If the hour is the highest speed hours (categorical feature)
4. If the day is week day or weekend (categorical feature)
5. Haversine and Manhattan Distance (numerical feature)

**Building and Evaluating Model**

We split the train set into train (80%) and validation set (20%), so we can evaluate model efficiency. The evaluation metrics that use are as follows:

* Bias: Mean of the difference between predicted and actual trip duration for all samples
* Normalized Bias: Mean of the difference between predicted and actual trip duration divided by actual trip duration for all samples. Presented as a percentage.
* Error: Mean of the absolute value of the difference between predicted and actual trip duration for all samples.
* Normalized Error: Mean of the absolute value of the difference between predicted and actual trip duration divided by actual trip duration for all samples. Presented as a percentage.
* R Squared: the proportion of the variation in the dependent variable that is predictable from the independent variables.
* Root Mean Squared Logarithmic Error (RMSLE): measure of the differences between predicted and actual trip duration.

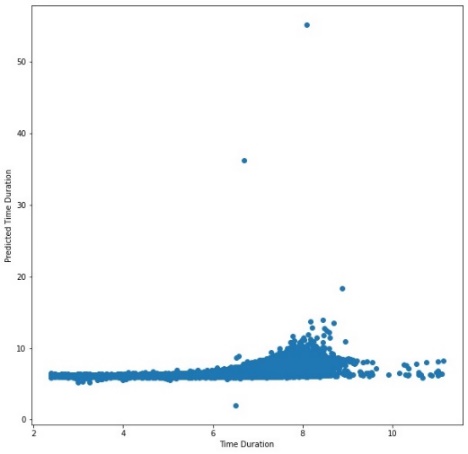
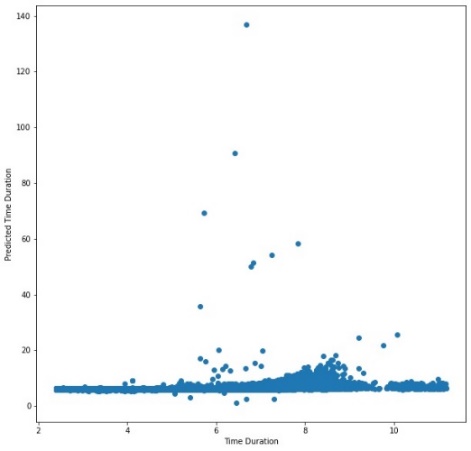
**Linear Regression Model**

Linear Regression is a regression of dependent variable on independent variable. It is a linear model that assumes a linear relationship between dependent (y) and independent variables (x). The dependent variable (y) is calculated by linear combination of independent variable (x).

Table 1 shows the evaluation metrics are as follows for train and validation sets.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Dataset | R2 | RMSLE | Bias | Norm\_Bias | Error | Norm\_Error |
| Train | 0.365 | 0.602 | -1.49e-15 | 1.0405 | 0.438 | 7.210 |
| Validation | 0.406 | 0.583 | 0.0004 | 1.054 | 0.438 | 7.217 |

Figure 13 shows scatter plots for train and validation sets. This model poorly predicts time duration, some predicted trip durations are too long, and R2 is relatively low for both train and test sets.



Train Set Test Set

**XGBoost Model**

XGBoost comes under boosting and is known as extra gradient boosting. GBM first calculates the model using X and Y then after the prediction is obtain. It will again calculates the model based on residual of previous model. Loss function will give more weight to error of previous model and this process continuous until MSE gets minimized.

XGBoost is just an extension of GBM with following advantages.

* Regularization
* Parallel Processing
* High Flexibility
* Handles Missing values
* Tree pruning
* Buitin cross validation
* Continuous on existing model

**Hyper parameter Tuning**

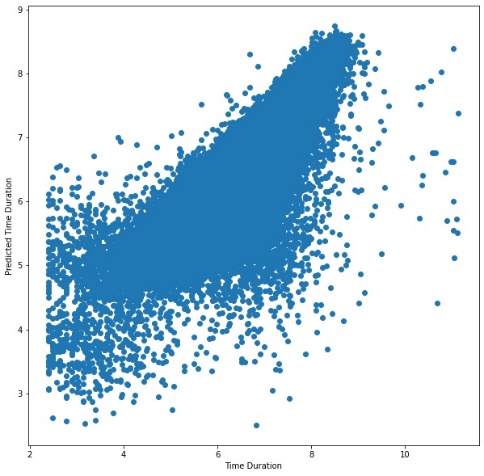
Hyperparameters are sets of information that are used to control the way of learning an algorithm. For XGBoost we estimated eta, min\_child\_weight, colsample\_bytree, max\_depth, subsample, and Lambda based on the model best score. The chosen hyperparameters are shown in bold

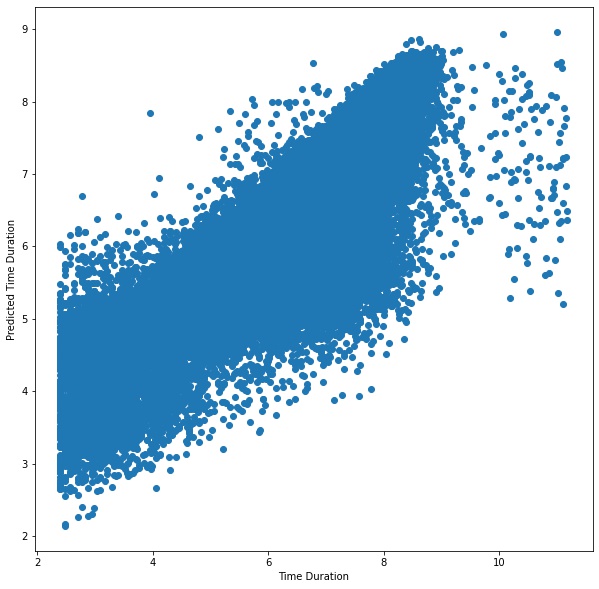
|  |  |  |
| --- | --- | --- |
| Hyperparameters | XGB |  |
| Eta | [0.05, 0.1, 0.15, **0.3**] |  |
| Min\_child\_weight | [**50**, 100] |  |
| Colsample\_bytree | [0.3, 0.4, **0.5**] |  |
| Max\_depth | [8, **10**] |  |
| subsample | [0.5, **0.8**] |  |
| Lambda | [0.5, **1.5**] |  |
| eval\_Score | RMSE |  |

Table 2 shows the evaluation metrics are as follows for train and validation sets. All the evaluation metrics are better than linear regression model. Normalized bias and error are low and R squared is around 0.8 for both datasets.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Dataset | R2 | RMSLE | Bias | Norm\_Bias | Error | Norm\_Error |
| Train | 0.840 | 0.302 | -0.00002 | 0.291 | 0.219 | 3.55 |
| Validation | 0.797 | 0.341 | 0.0008 | 0.363 | 0.236 | 3.866 |

Figure 14 shows the predicted trip duration for both train and validation datasets, the predicted values are in good agreement to the actual values. The XGBoost model performance is better than linear regression model.





Train dataset Validation dataset

**Feature Importance**

Feature Importance refers to techniques that calculate a score for all the input features for a given model, the scores simply represent the “importance” of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict a certain variable.

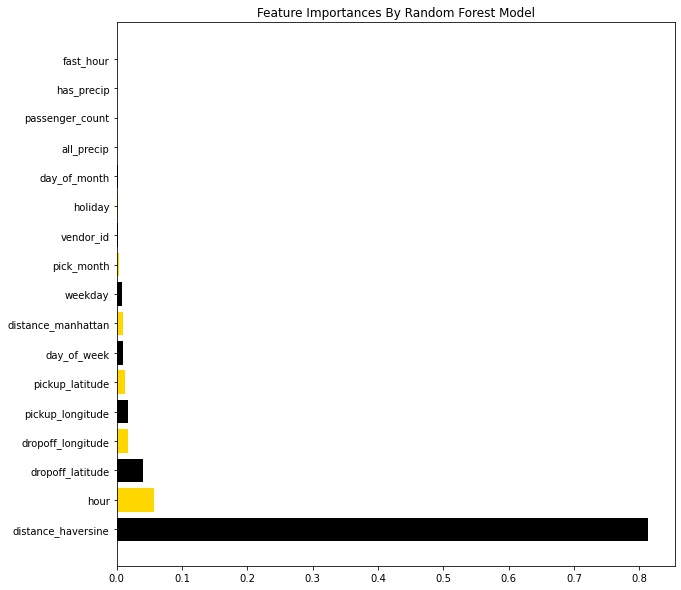
We check the feature importance for this model, table 3 shows the features and corresponding importance. The most important feature for this model is the amount of precipitation.

|  |  |
| --- | --- |
| feature\_name | Importance |
| day\_of\_month | 0.214118 |
| day\_of\_week | 0.119769 |
| dropoff\_longitude | 0.113163 |
| pickup\_longitude | 0.111971 |
| dropoff\_latitude | 0.111628 |
| pickup\_latitude | 0.111480 |
| passenger\_count | 0.111322 |
| pick\_month | 0.034971 |
| hour | 0.016167 |
| weekday | 0.011789 |
| vendor\_id | 0.009752 |
| distance\_haversine | 0.009735 |
| all\_precip | 0.007195 |
| fast\_hour | 0.006275 |
| distance\_manhattan | 0.005924 |
| holiday | 0.004738 |
| has\_precip | 0.000000 |

**Random Forest Regressor**

Table 3 shows the evaluation metrics are as follows for train and validation sets. All the evaluation metrics are better than linear regression model.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Dataset | R2 | RMSLE | Bias | Norm\_Bias | Error | Norm\_Error |
| Train | 0.794 | 0.343 | 0.000008 | 0.383 | 0.243 | 3.970 |
| Validation | 0.772 | 0.361 | 0.0004 | 0.411 | 0.256 | 4.191 |



**Decision Tree**

