NYC Taxi Trip Time Prediction

Ganeshkumar Patel, Akanksha Agarwal,
Data Science Interns, AlmaBetter, Bangalore

Abstract:

New York City, with a relatively large number of populations and busy streets require taxi with some good algorithms to move from one place to another without any delay. A lot of streets and roads in New York city are quite busy due to traffic jams, construction or road blockage etc. Therefore, it is very important to predict the trip duration of taxi so that the user will know how much time it will take to commute from one place to other.

In our project, for predicting the taxi trip duration we have applied Linear Regression, Random Forest, XG Boost and Light GBM models to find out which one is giving better accuracy with less amount of prediction time. At last, a comparison was performed and found that Light GBM is working best for given dataset.

Keywords: machine learning, trip duration, categorical features, feature engineering

1. Problem Statement

The dataset is based on the 2016 NYC Yellow Cab trip record data made available in Big Query on Google Cloud Platform.

The data was originally published by the NYC Taxi and Limousine commission (TLC). The main objective is to build a predictive model, which could help them in predicting **t**rip duration of taxi. This would in turn help them in matching the right cabs with the right customers quickly and efficiently.

- 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 ofpassengers in the vehicle (driverentered 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 - Y=store and forward; N=not a store and forward trip
- trip_duration - duration of the trip inseconds

2. Introduction

New York City taxi rides taken every day by New Yorkers in the busy city can give us a great idea of traffic times, dense places of city a nd so onThus, predicting the duration of a taxi t rip is very important to know the time taken by ride both for the users as well as cab drivers.

Therefore, in this project, we used data which c ustomers would provide at the start of a ride, or while booking a ride to predict the duration.
The flow of our project is as follows:

- Data Discovery
- Exploratory Data Analysis
- Feature Engineering
- Model Building
- Hyperparameter Tuning
- Feature Importance
- Check on Kaggle with score
- Conclusion & Future Scope

3. Factors affecting Trip duration & its Variation

Trip duration is normally calculated based on the distance between pickup and dropoff point and average speed of the vehicle covering this distance. However, in reality there are many factors which affects the trip duration. Following are some of the factors:

- Peak hours: there are certain hours where route are might get busy due to moment of peoples commutating from office to home or vice versa.
- bad weather conditions (rain, snow, etc)
- big events or festivals
- traffic conditions

4. Working flow of Project

4.1 Data Loading and exploration

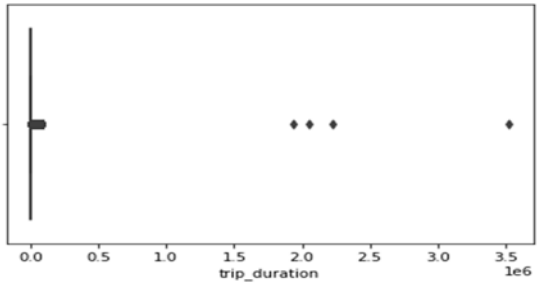
We have loaded the data from the given csv files using a function from pandas library. Then we checked the general information about data. We observed that the data contains 1458644 records and 11 features. We see that our data contains three different data types i.e. floats, stringsand datetime objects.

4.2 Null values Treatment

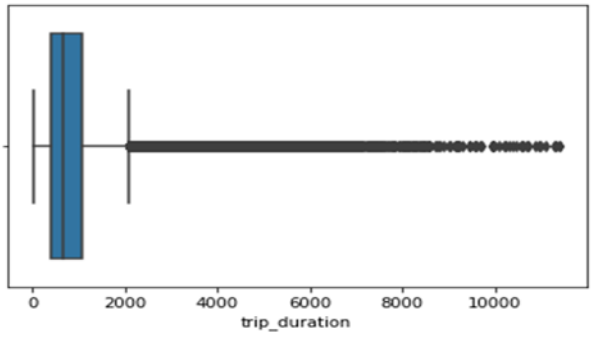
We inspected the dataset and found out that our dataset has no null value present in it. So, luckily, we skip this step.

4.3 Exploratory Data Analysis

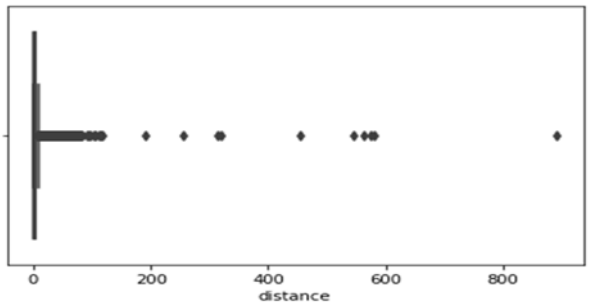
We begin our EDA by first checking the distribution of our dependent variable i.e. trip duration. We observed that the data is highly positively skewed.



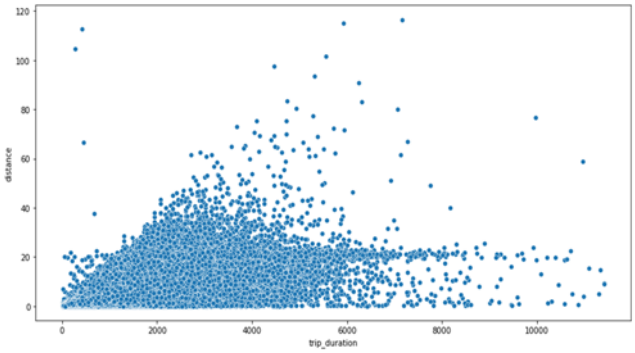
We also plotted the box plot and observed that there are many outliers present in the variable. To eliminate the outliers, we have segregated the data variable into different segment and observed that majority of trip duration is within an hour some observation are within two days but a very few observation are having more than two days. We eliminate such values from out dataset.



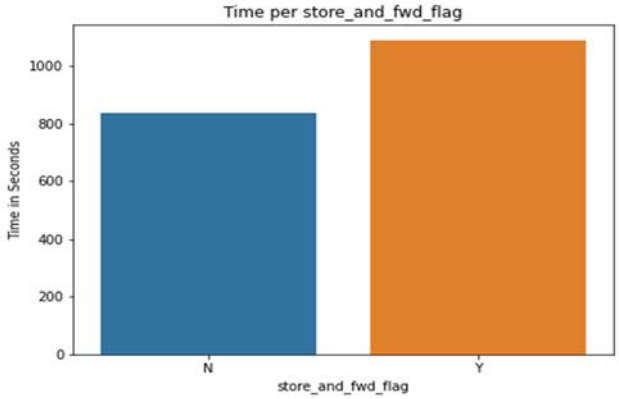
We removed id variable as it doesn't give much interpretation. We then calculated the distance based on haversine formula from pickup and drop-off latitude and longitude.



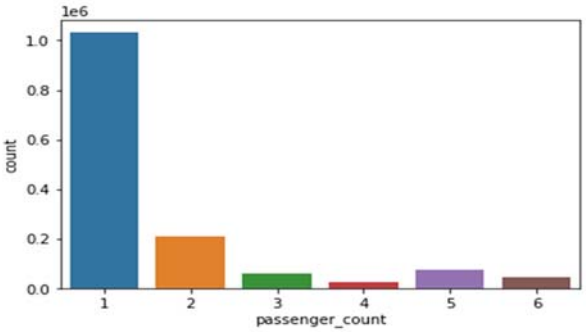
Then we plotted the box plot for the variable and observed there are many outlier so we segregate this variable and see that most of the trip are within 10km, some trip are within50km while a very few trip crosses 50km. so we eliminate trip with less than 50m and above 180km distance.

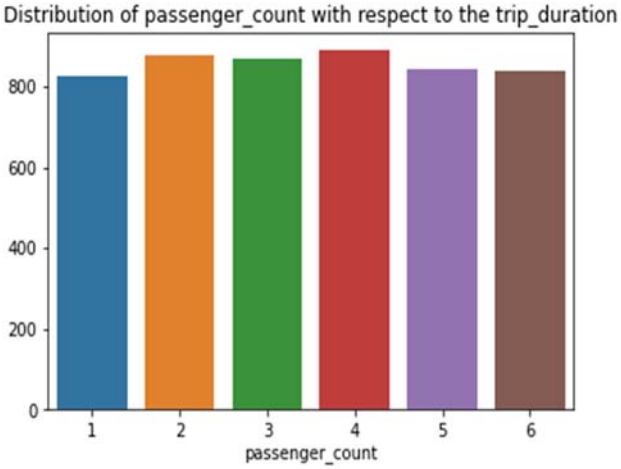


We then checked for categorical variable store_and_fwd_flag and passenger_count. We observed that store and fwd flag contains majority of one category but when considering the mean over trip duration then the scenario seems different.

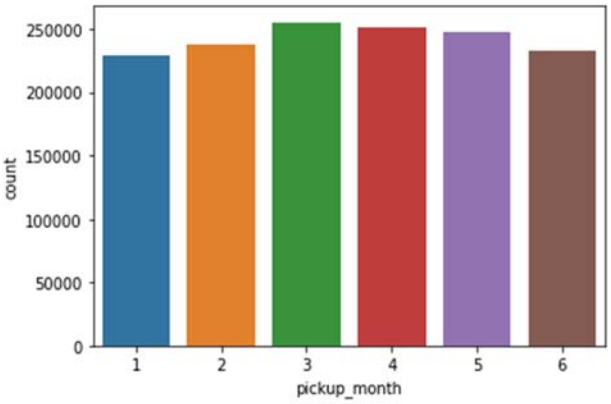


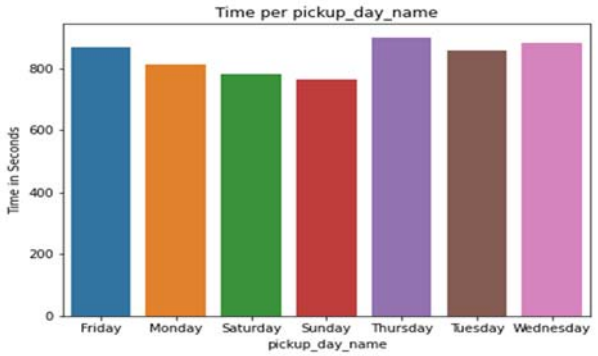
Passenger count variable has entries from 0 to 9. Since there are no trips with 0 passenger either this a miss entry or the driver forgot to enterpassenger count of that trip. Also, in ataxi maximum six person are allowed to sit including minor. So, we eliminated 0 and 7-9 records from our dataset and took the mean of each passenger count over trip duration.

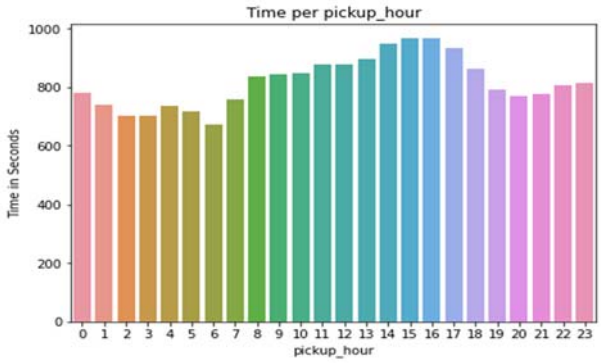




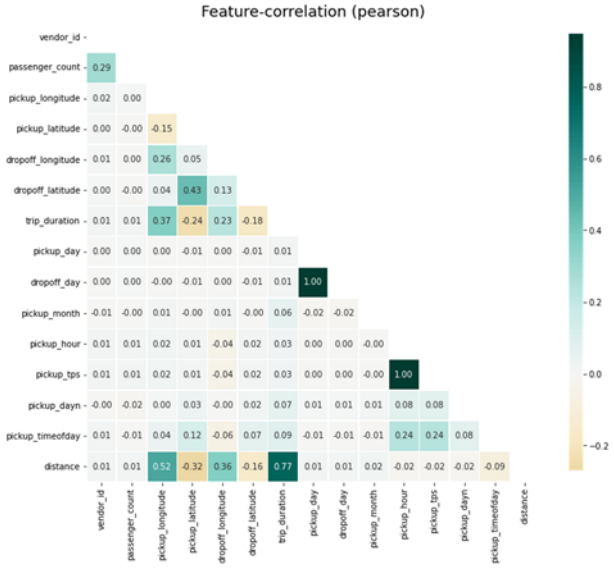
We also created some more feature i.e. pickup month, pickup weekday and pickup hour, to get a good insight of trip duration and drop pickup date and drop-off timecolumn.







Then we checked for correlation between variables and observed that geographic coordinates are very less correlated and VIF is also high between these variables so we drop off this variable from our data set.



4.4 Encoding of categorical columns

Since some of our categorical variables are in string format. So, we cannot pass this variable to our model directly therefore we used onehot encoding to convert it into numerical variable having binary integers 0 and 1.

4.5 Features Normalization

This is one of the important steps for getting good accuracy as you can see there are some columns having different ranges of values then other column. Therefore. It is important to do scaling the data so that our data set will have uniformity and we get good accuracy. So, here we use MinMaxscaler function.

4.6 Fitting different models

For modelling we tried variousclassification algorithms like:

1. **Linear Regression**
2. **RandomForest**
3. **XGBoost classifier**
4. **LightGBM**

4.6.1 Linear Regression:

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).

$$Y=B_0+B_1x_1+B_2x_2$$

The cost function for linear regression isgiven by: **Minimum sum of square error**

$$MSSE= \sum_{1}^{n}(Y_i act - Y_i pred)^2$$

4.6.2 Random Forest:

It is a versatile machine learning method capable of performing both regression and classification tasks. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data exploration, and does a fairly good job. It is a type of ensemble learning method,

where a group of weak models combine to form a powerful model.

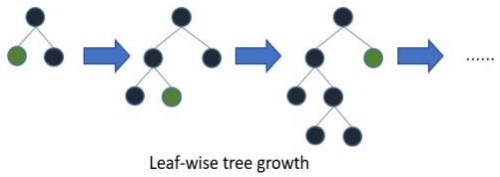
4.6.3 XGBoost:

Sometime in building a model we cannot just rely on the result of a single model. Ensemble offer a systematic solution for this by combining the prediction of multiple models. The resultant model is superior then individual model called base learner and is obtained from aggregation of base learner prediction. Bagging and boosting are two types of ensemble method.

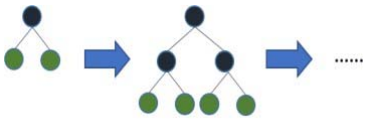
XGBoost comes under boosting and is known as extra gradient boosting. GBM first calculates the model using X and Y then after the prediction is obtain. It will again calculate the model based on residual of previous model, here loss function will give more weightage to error of previous model. and this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

4.6.4 LightGBM:

Sometime in building a model LightGBM is a fast, distributed high performance gradient boosting framework. It is widely used for ranking, classification, regression and many other machines learning task. LightGBM is based on decision tree algorithm. But it splits the tree leaf wise rather than level wise like another boosting algorithm. So, when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms.



Leaf-wise tree growth



Level-wise tree growth

4.7 Model performance:

The model performance can be evaluated by various regressionmetrics such as:

4.6.3 Mean Squared Error (MSE)- Mean squared error is the most widely used evaluation metric for regression task. It is the average of squared difference between actual and predicted value of dependent variable. The constant base line model is chosen by taking the mean of the data and drawing a line at mean.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

4.6.4 R2 score

The coefficient of determination or R-squared represents the proportion of the variance in the dependent variable which is explained by the linear regression model. It is a scale-free score i.e. irrespective of the values being small or large, the value of R square will be less than one.

$$R^2 = 1 - \frac{MSE(model)}{MSE(Baseline)}$$

4.8 Hyperparameter tuning

Hyperparameters are sets of information that are used to control the way of learning an algorithm. There are three types of tuning Grid Search CV, Randomized Search CV and Bayesian Optimization for hyperparameter tuning. This also results in cross validation and in our case, we divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization, but we opted for Randomized Search CV for less computational cost.

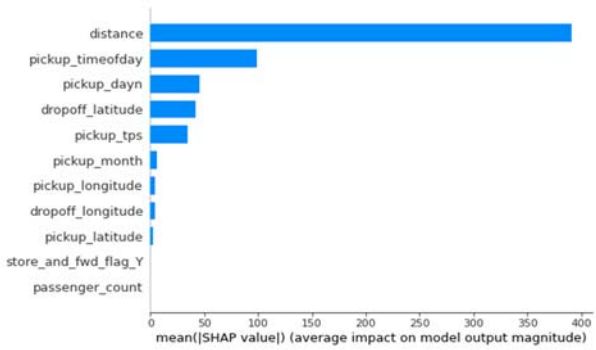
Randomized Search CV- In Random Search, the hyperparameters are chosen at random within a range of values that it can assume. The advantage of this method is that there is a greater chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist's control.

Model/ Metric	LR	Random Forest	XGB	Light GBM
Train MSE	$15.99e^4$	$11.73e^4$	$82.79\ e^4$	64.46
Test MSE	$15.99e^4$	$11.77e^4$	$82.94\ e^4$	81.78
Train R2	0.63	72.89	80.86	84.63
Test R2	0.63	72.86	80.88	81.78
Adj R2	0.63	72.86	80.88	81.15

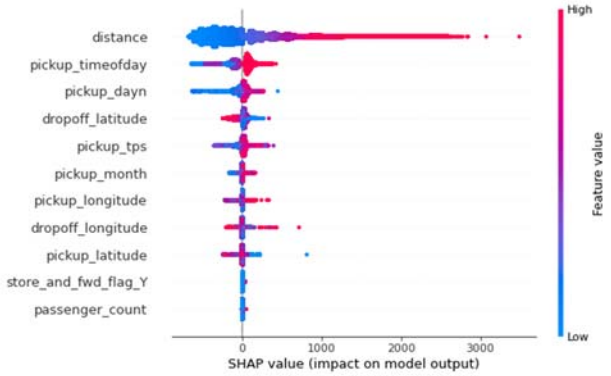
4.7 Model Explainability

Model explainability refers to the concept of being able to understand the machine learning model. It is necessary that once the model is deployed in the real world then, the model developers can explain the model

.**SHAP (SHapley Additive exPlanations)-** SHAP shows the impact of each feature by interpreting the impact of a certain value compared to a baseline value. The baseline used for prediction is the average of all the predictions. SHAP values allow us to determine any prediction as a sum of the effects of each feature value. The only disadvantage with SHAP is that the computing time is high. The Shapley values can be combined together and used to perform global interpretations also.



From the above plot we can conclude that **distance has the highest feature importance**



The above summary plot combines feature importance with feature effects. It can be inferred that:

1. High value of distance has greater impact on forcing predicted value of trip duration and vice versa.
2. Higher Drop off latitude values shown in red color helps to predict trip duration on lower side.

5. Conclusion

Thus, our project has achieved its target of predicting the trip duration with test score of 81% through Light GMB model. This score is reliable as we tested it further on Kaggle and achieved score of approx. 0.46. For achieving our target, we tried our level best to put maximum efforts in feature engineering and by adding few more features. But there is always room to further work and train our model to predict the duration, by adding some more relevant features or by performing hyperparametertuning.