Capstone project 2

NYC TAXI TRIP TIME PREDICTION

Individual project
DSN NAVEEN KUMAR

CONTENT

- Introduction
- Problem statement
- Data summary
- Exploratory Data Analysis (EDA)
- Feature Engineering & Selection
- Building and Evaluating Model
- Conclusion

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.
- So here we will be building a model which will be predicting the trip duration of taxies running in NewYork. 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 which is based on the 2016 NYC Yellow Cab trip record. The data was originally published by the NYC Taxi and Limousine Commission (TLC). Based on individual trip attributes, our task is to build a model that predicts the total ride duration of taxi trips in New York City.

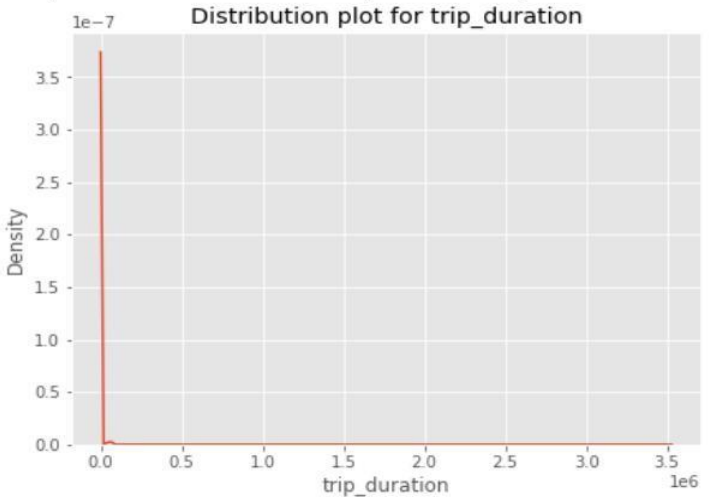
DATA SUMMARY

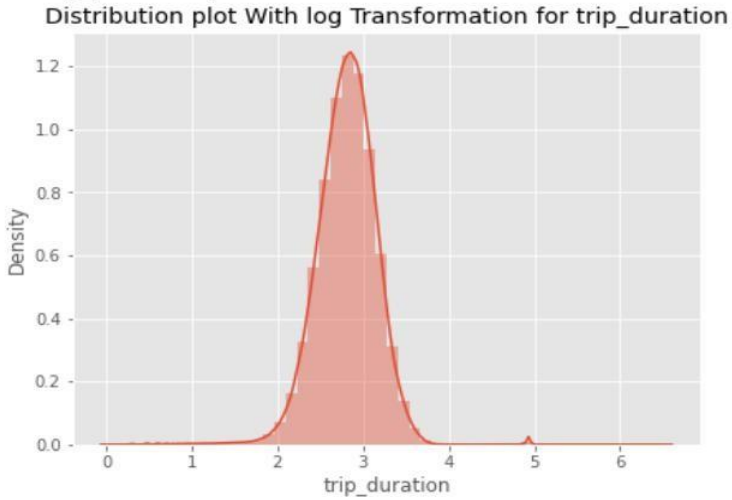
- id - a unique identifier for each trip
- vendor_id - a code indicating the provider associated with the trip record
- pickup_datetime - date and time when the meter was engaged
- dropoff_datetime - date and time when the meter was disengaged
- passenger_count - the number of passengers in the vehicle (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

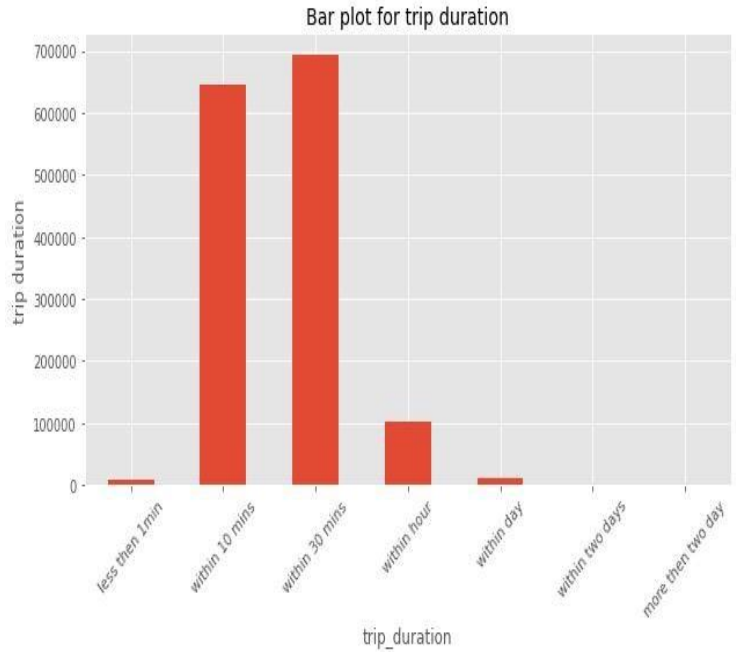
BASIC EXPLORATION

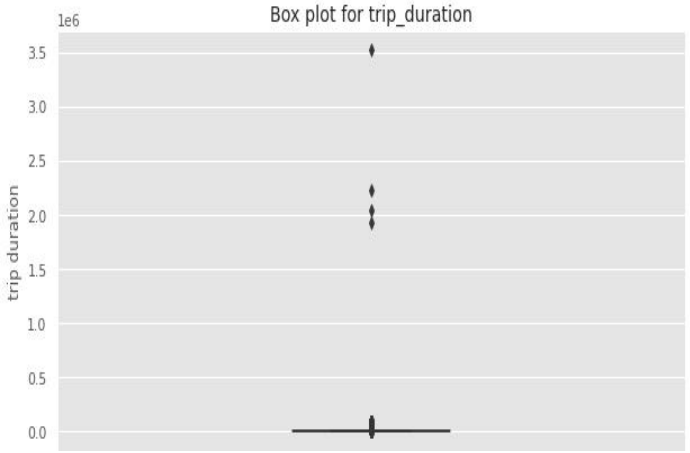
- The dataset contains 1458644 rows and 11 columns.
- Two categorical features ‘store_and_fwd_flag’ and ‘vendor_id’
- Outliers present in all numerical features
- Data formating steps required for datetime features
- No null values present

TRIP DURATION DATA ANALYSIS



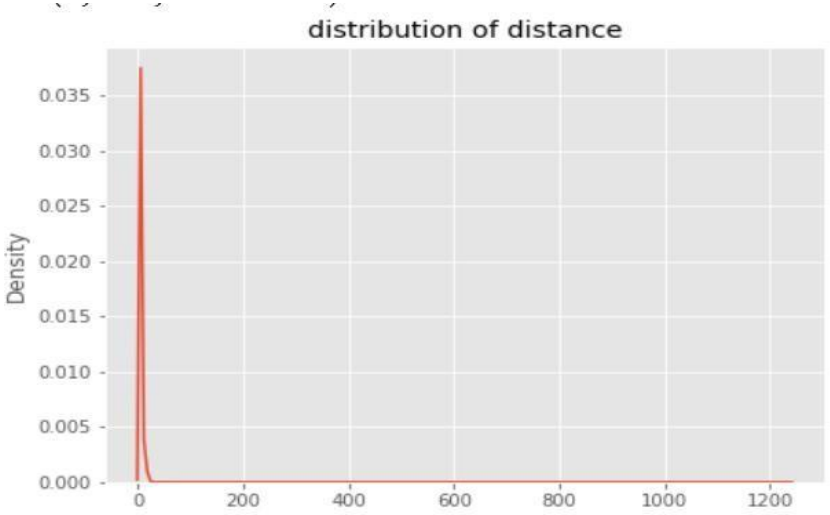


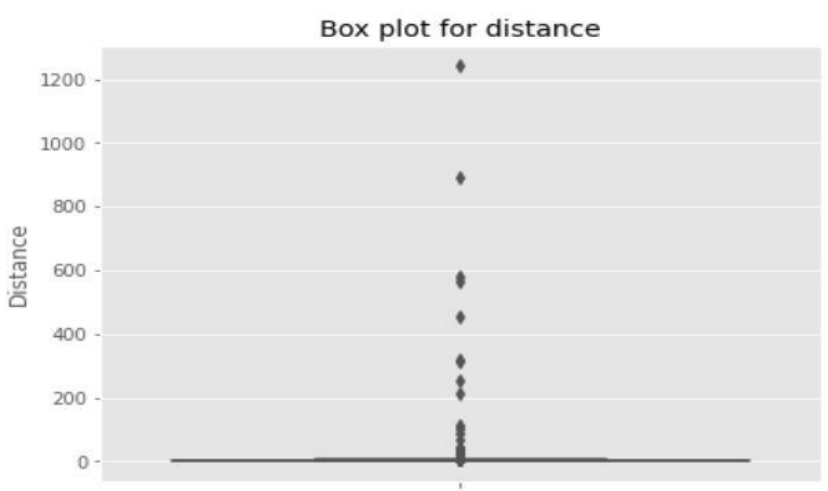


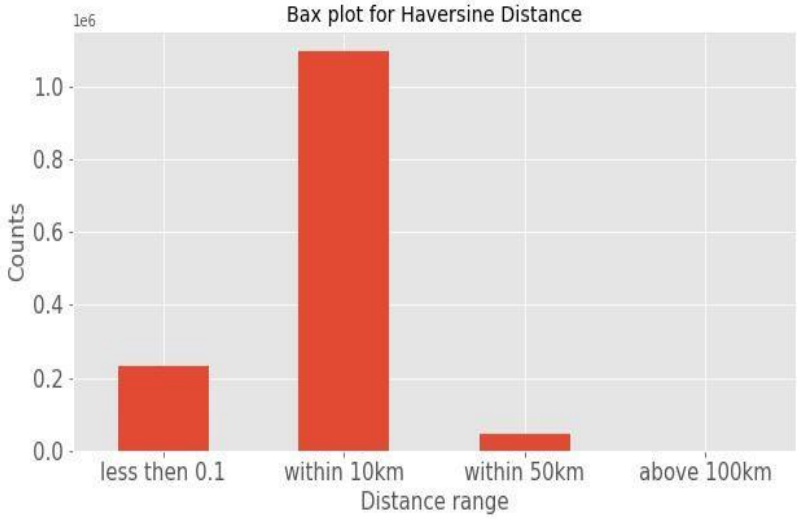


Haversine Distance

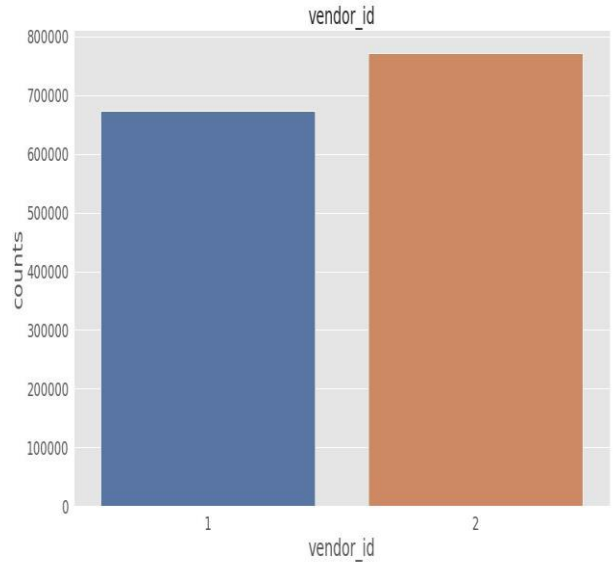
$$D = 2rsin^{-1}\left(\sqrt{sin^2\left(\frac{\varphi_2-\varphi_2}{2}\right) + cos(\varphi_1)cos(\varphi_2)sin^2\left(\frac{\lambda_2-\lambda_2}{2}\right)}\right)$$

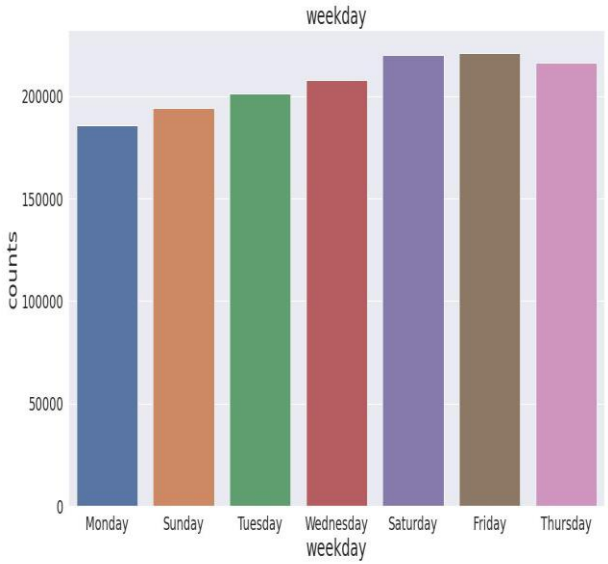


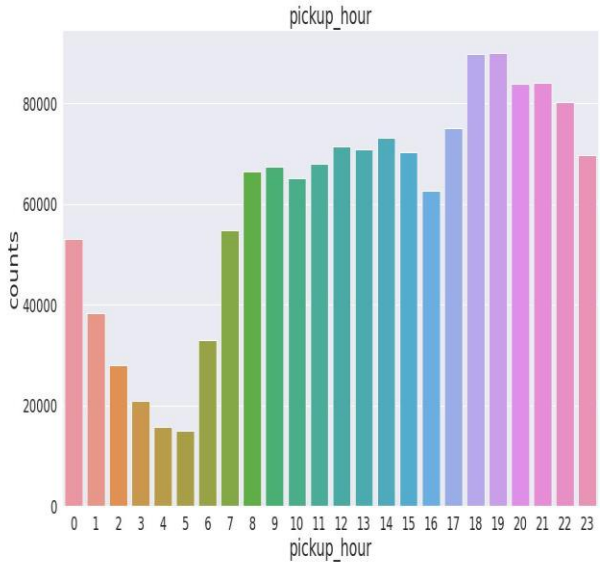


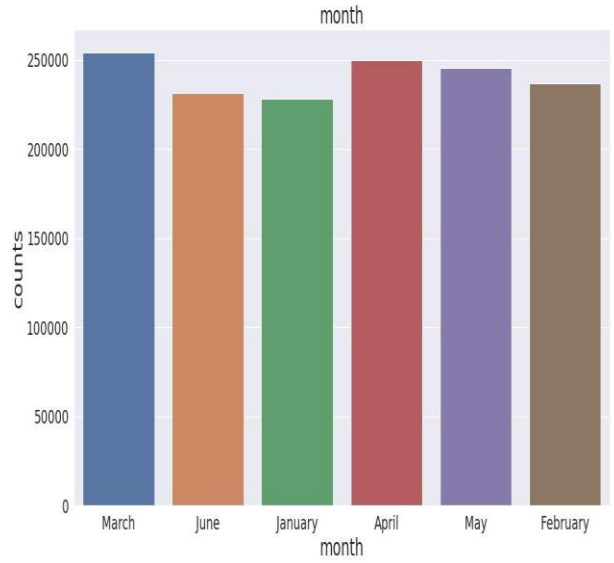


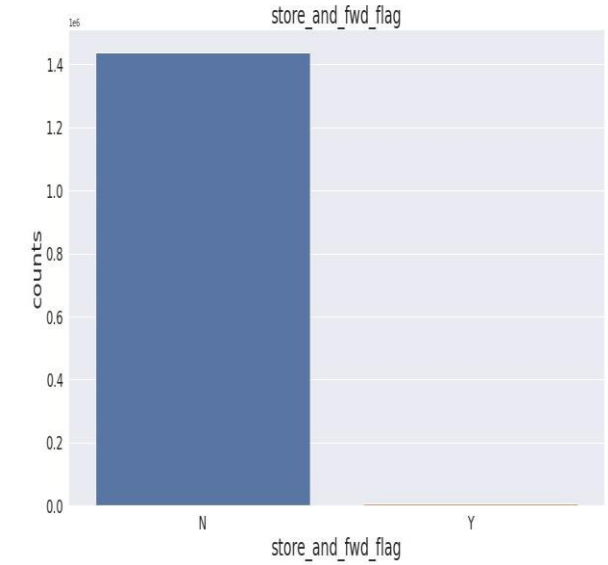
Categorical Variable Analysis

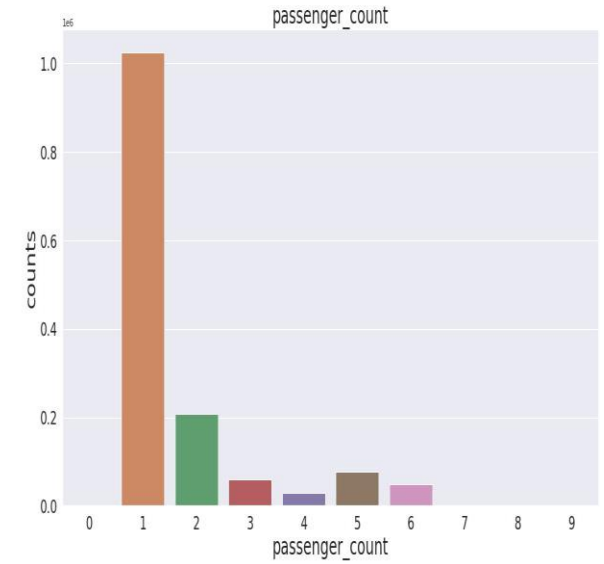




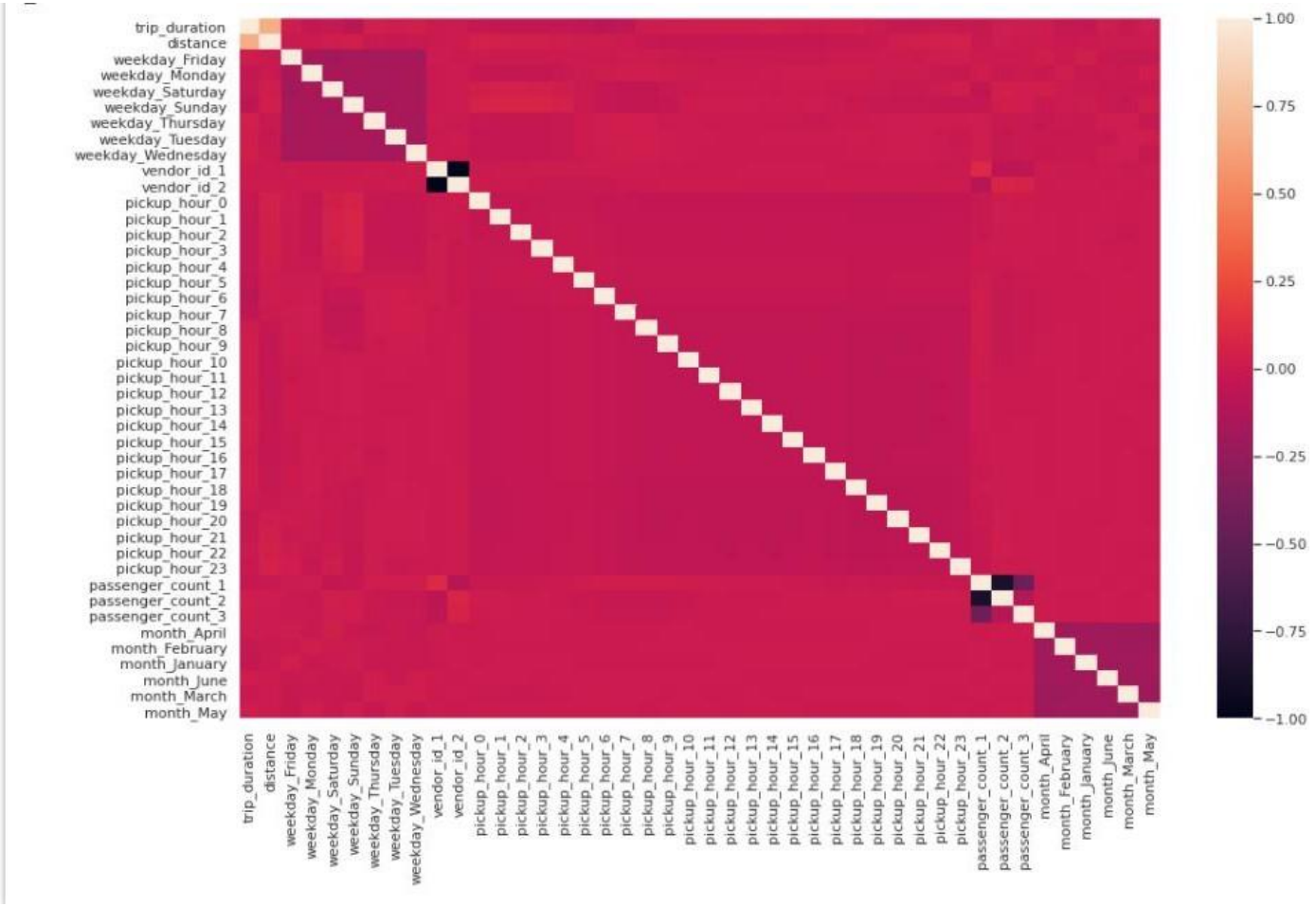








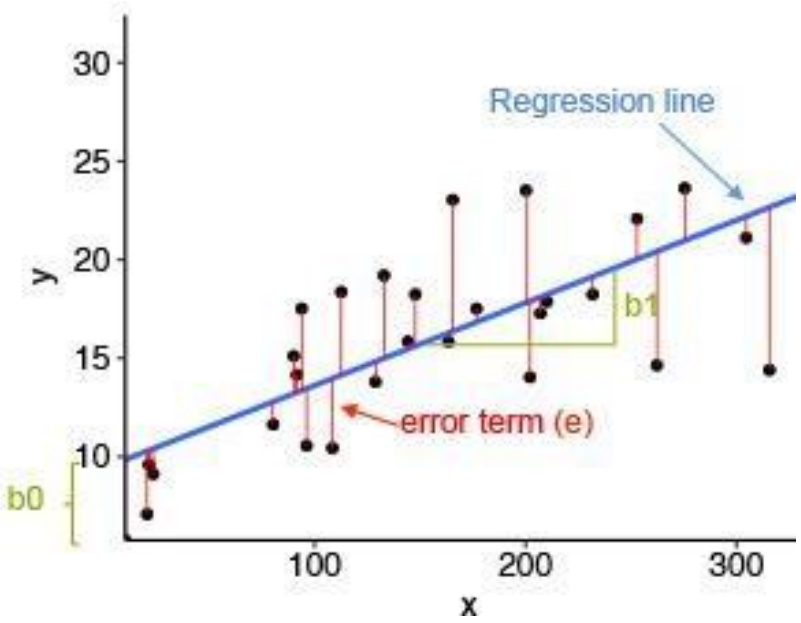
Correlation Analysis



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).

$$y=b_0+b_1 x_1 +b_2 x_2$$



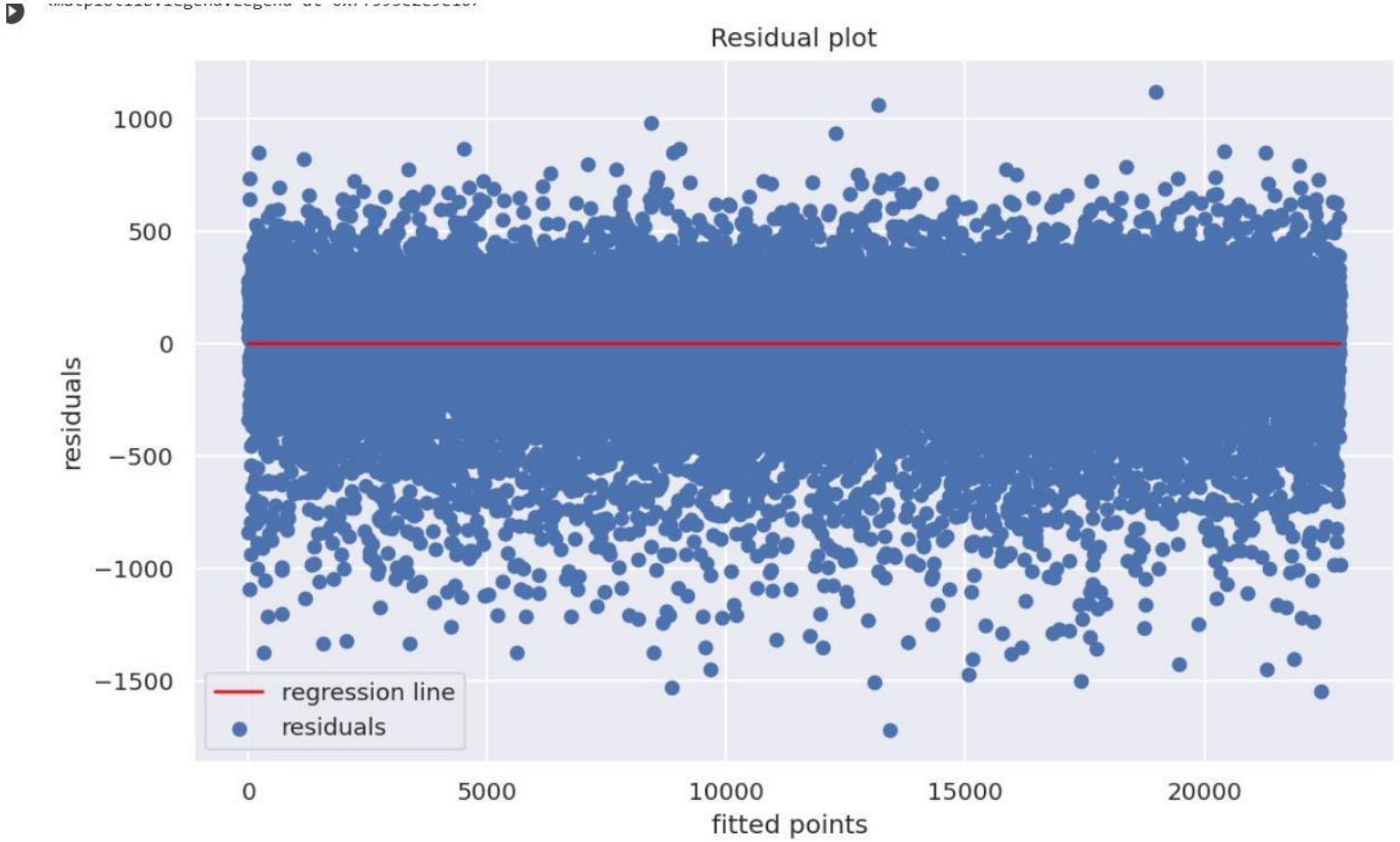
The cost function for linear regression is given by:

Mean of sum of square error

$$S = \frac{\sum (cul - rdicd)^2}{N}$$

MSE	130903.06
RMSE	361.81
R2	0.6089
Adjusted_R2	0.6083

Homoscedasticity check



XGBoost

- 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 weightage to error of previous model. and this process continuous until MSE gets minimizes.

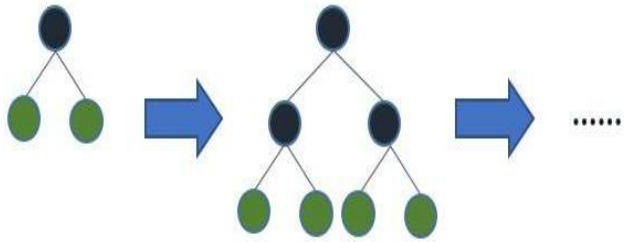
XGBoost is just an extension of GBM with following advantages.

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

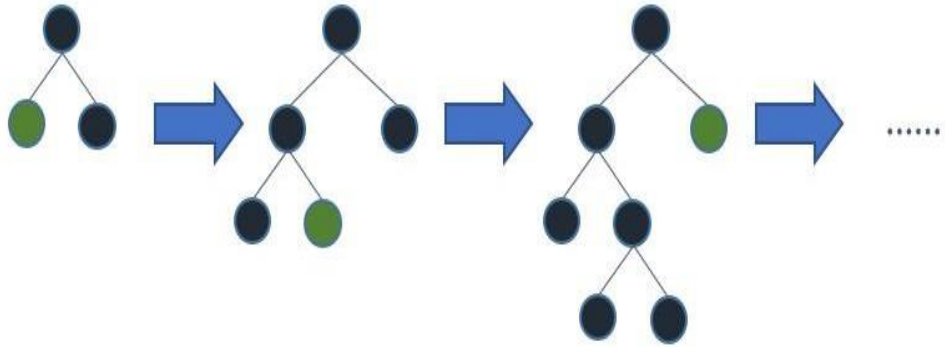
MSE	105226.37
RMSE	324.39
R2	0.6856
Adjusted_R2	0.6083

LightGBM

- LightGBM is a fast, distributed high performance gradient boosting framework.
- LightGBM is based on decision tree algorithm. But it splits the tree leaf wise rather then level wise like other 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.



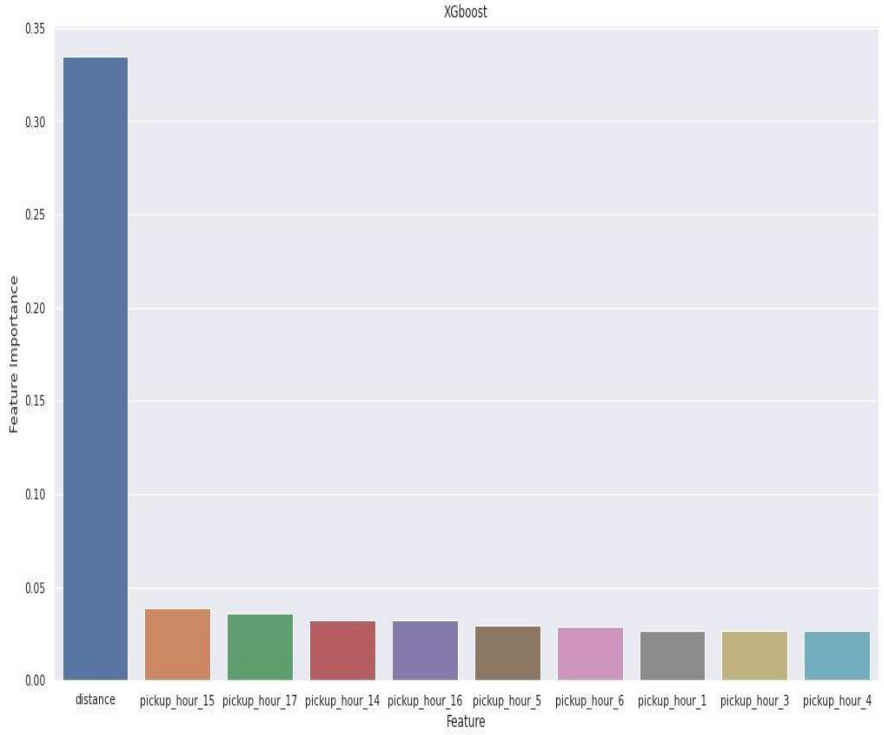
Level-wise tree growth

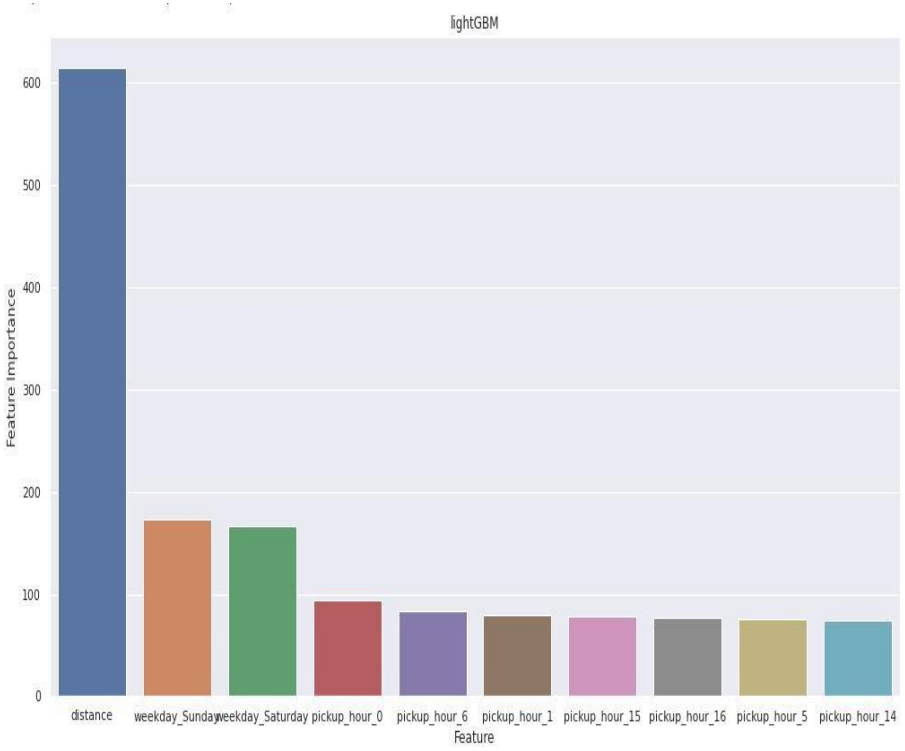


Leaf-wise tree growth

MSE	95652.24
RMSE	309.28
R2	0.7142
Adjusted_R2	0.7138

Feature Importance

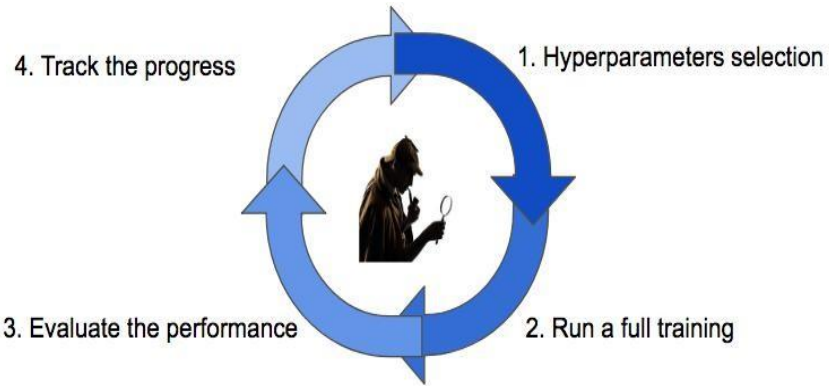




Hyperparameter Tuning

Hyperparameters are sets of information that are used to control the way of learning an algorithm.

We used Grid Search CV, for hyperparameter tuning. This also results in cross-validation and in our case we divided the dataset into different folds



Hyperparameters	XGBoost	LightGBM
n_estimator	[5,10,20]	[5,10,20]
max_depth	[5,7,9]	[5,7,9]
min_samples_split	[40,50]	[40,50]
cv	3	3
eval_Score	R2	R2

Metric	XGBoost	LightGBM
MSE	103672.2	103515.51
RMSE	321.98	321.74
R2	0.6933	0.6938
Adjusted_R2	0.6857	0.6862

Final Metrics Conclusion

Metrics	Linear regressor	XGboost	lightGBM
MSE	130903.0600	105226.3700	95652.2400
RMSE	361.8100	324.3900	309.2800
R2	0.6089	0.6856	0.7142
Adjusted_R2	0.6083	0.6851	0.7138

Conclusion

- In this project, I tried to predict the trip duration of a taxi in NYC.
- I am mostly concerned with the information of pick up latitude and longitude and drop off latitude and longitude, to get the distance of the trip.
- Hyperparameter tuning doesn't improve much accuracy.
- Linear regression gives 60.89 % accuracy,XGBoost gives 68.56% accuracy, LightGBM gives 71.42% on the test set.
- LightGBM is more fitter and efficient than XGBoost for taxi trip duration-based predictions
- LightGBM will be the best model to predict the trip duration for a particular taxi.

Challenges

- Handling Large Dataset
- Feature Engineering
- Computation Time
- Optimising The Model


THANK YOU