Capstone Project – 2

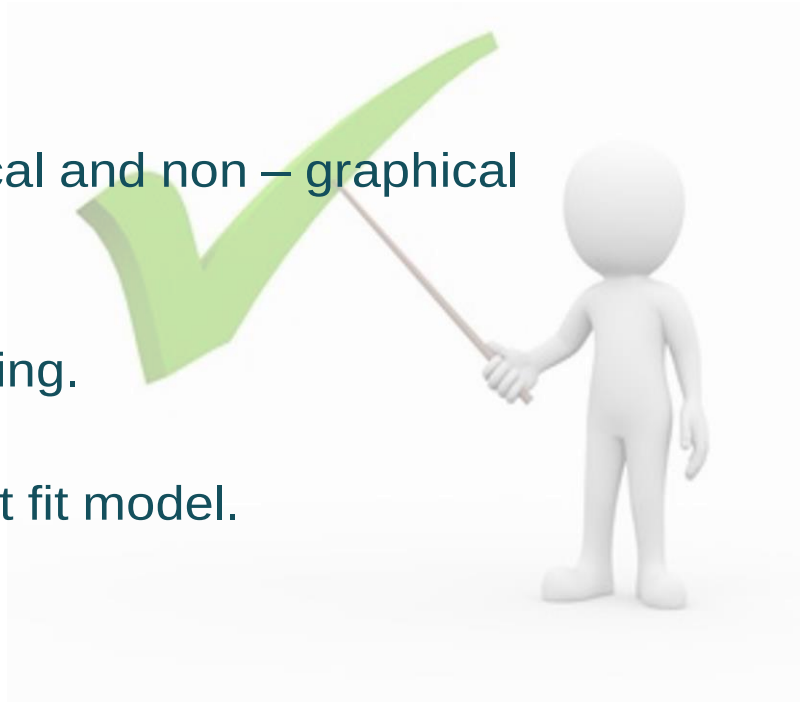
NYC Taxi Trip Time Prediction Supervised ML-Regression

Team Members

JUNAID A R

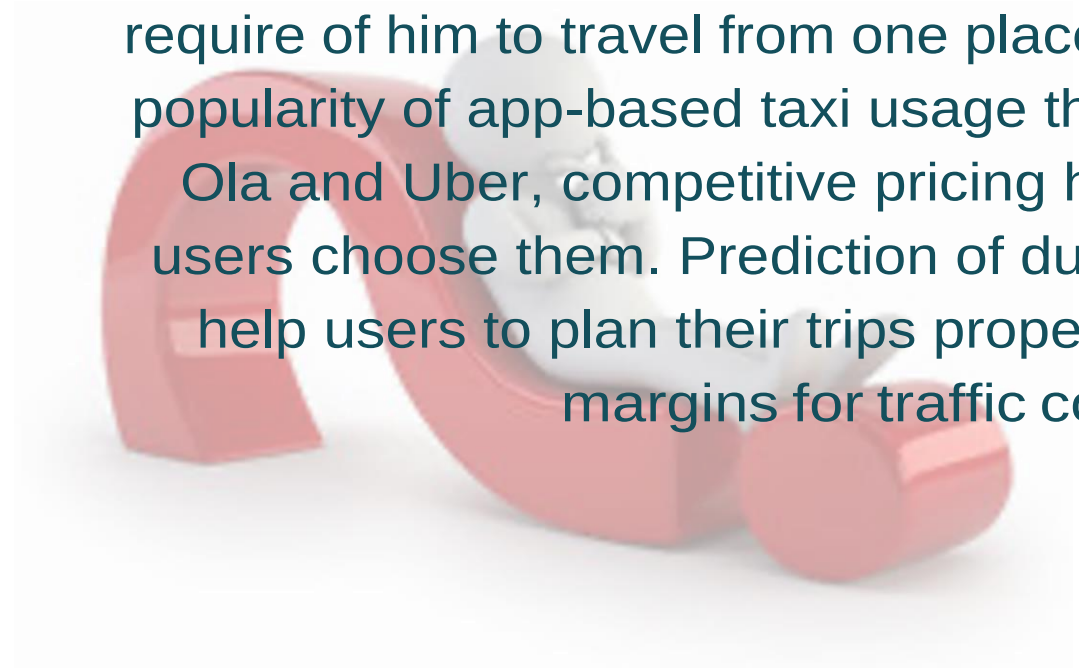
INTRODUCTION

- Explain what I am trying to solve.
- Exploratory data analysis using graphical and non – graphical approach.
- Feature selection and feature engineering.
- Applying models and validating the best fit model.
- Conclusion.



What a solving ?

Predicting the duration of a taxi trip is very important since a user would always like to know precisely how much time it would require of him to travel from one place to another. Given the rising popularity of app-based taxi usage through common vendors like Ola and Uber, competitive pricing has to be offered to ensure users choose them. Prediction of duration and price of trips can help users to plan their trips properly, thus keeping potential margins for traffic congestions.



Data Features

- **id** - a unique identifier for each trip
- **Vendor_id** - 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(dependent variable)** - duration of the trip in seconds

Data Pipeline

- Data Preprocessing - In this part we removed unnecessary features and peculiar observations. Since there were some observations which were out of the box and unnatural.

- Feature engineering - Here we manually went through each of the features and constructed new features from existing data to extract valuable information from some features to train model.

- EDA - Then we did some exploratory data analysis on the features from the previous actions to visualize the pattern.

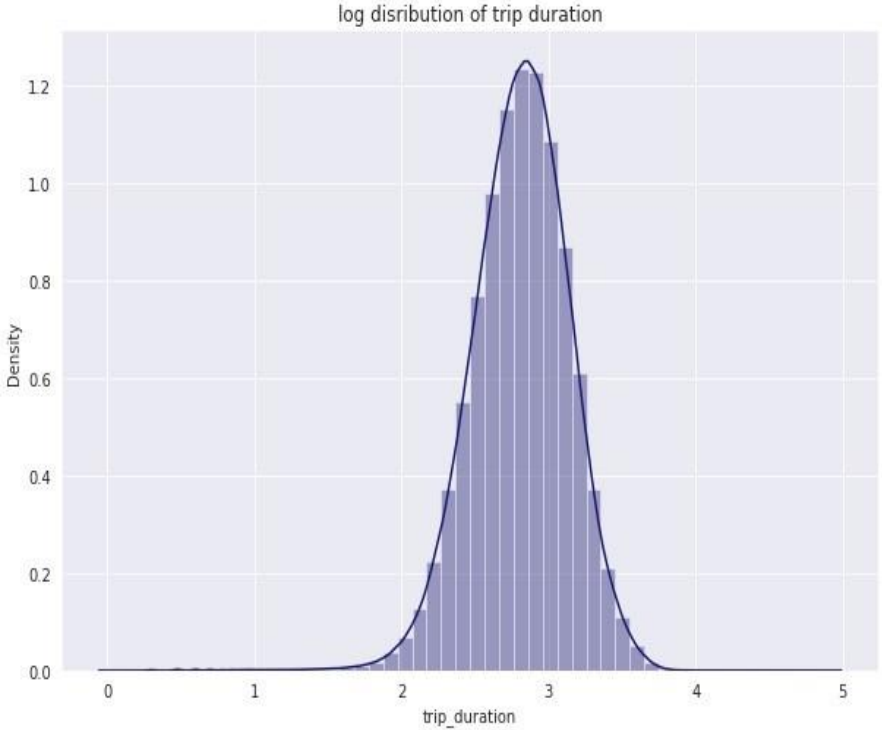
- Model evaluation - Finally we prepared our features to train various models and evaluate their performance in an iterative manner starting from the simplest model and gradually increasing the complexity.

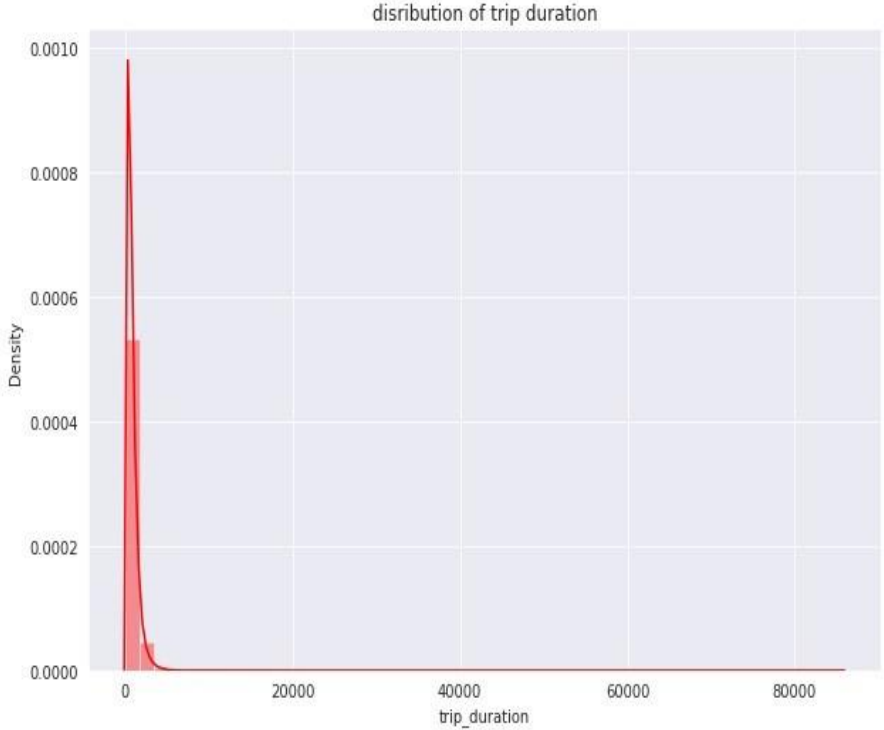
Exploratory data analysis

- **Univariate non-graphical and graphical.** Non-graphical methods don't provide a full picture of the data. Graphical methods are therefore required. Representations like histogram, Bar Plot, etc.

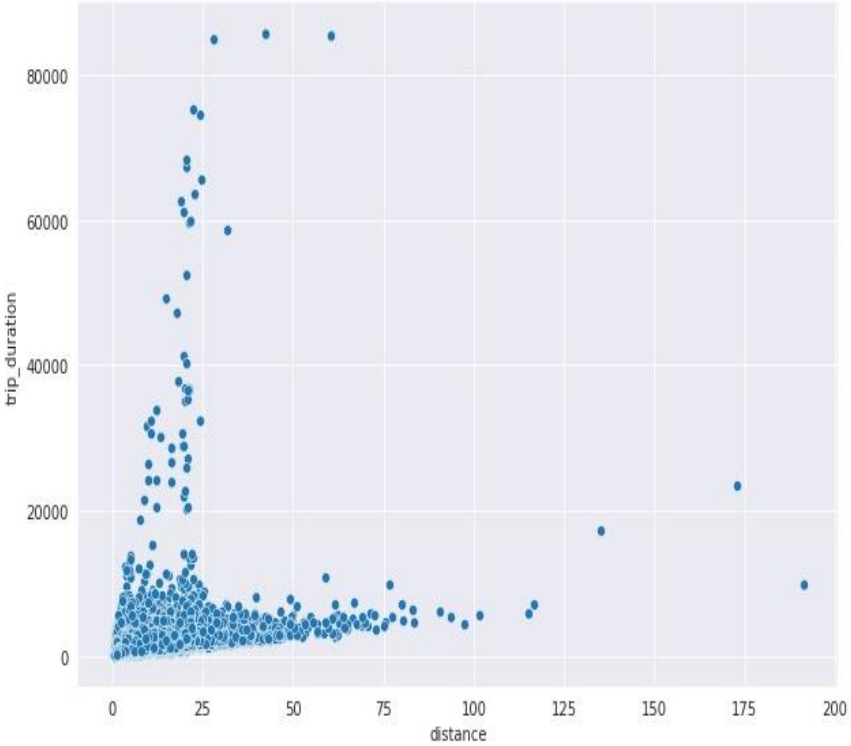
- **Multivariate non-graphical and graphical:** Multivariate data uses graphics to display relationships between two or more sets of data. The most used graphic is a grouped bar plot or bar chart with each group representing one level of one of the variables and each bar within a group representing the levels of the other variable. For example Heat Map.

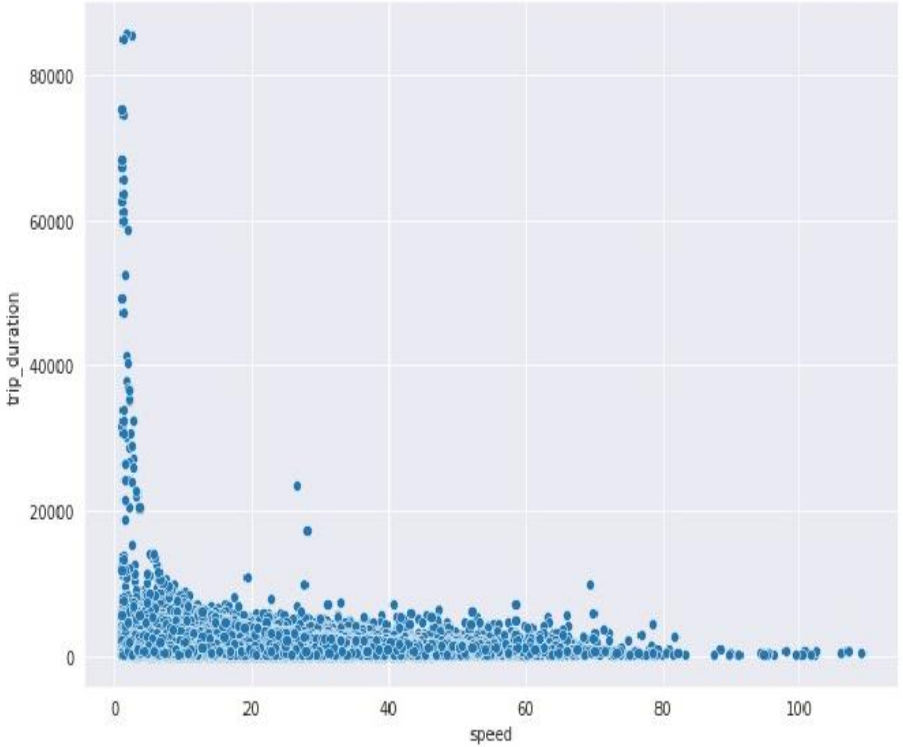
Dependent variable – Trip duration



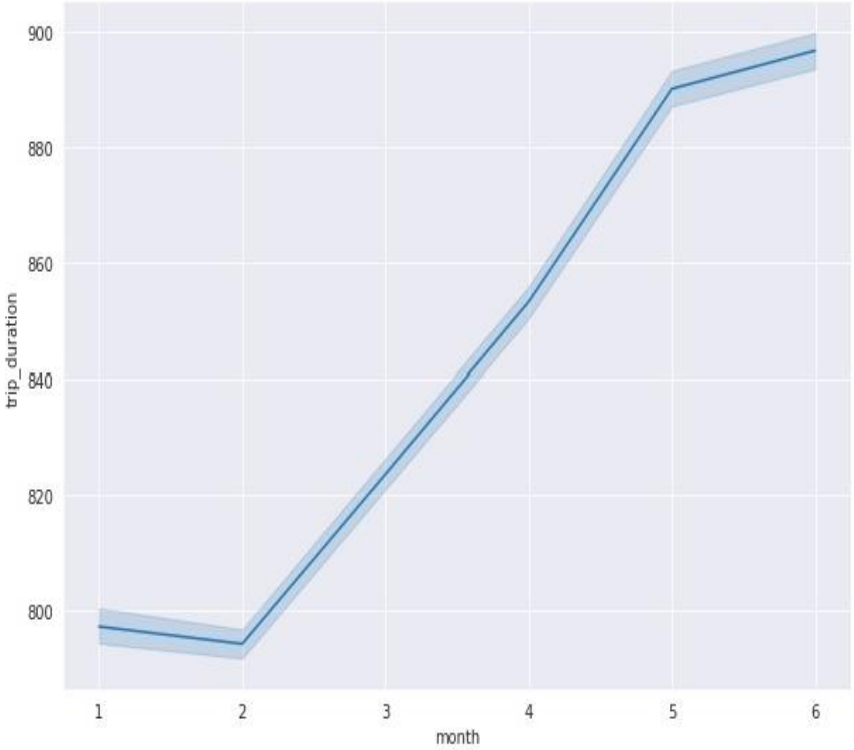


Trip duration vs distance and speed

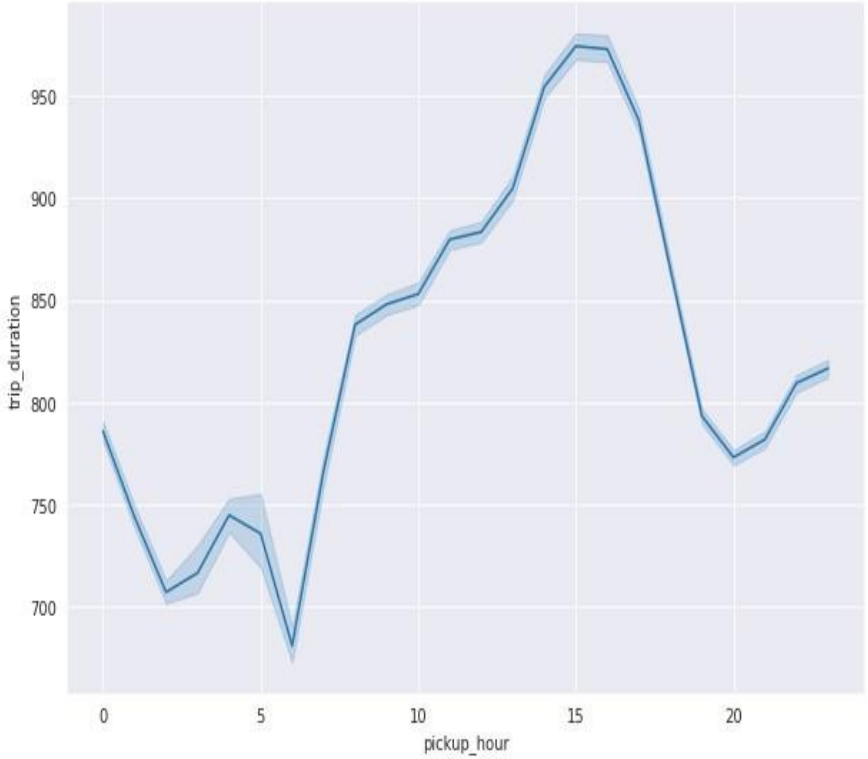


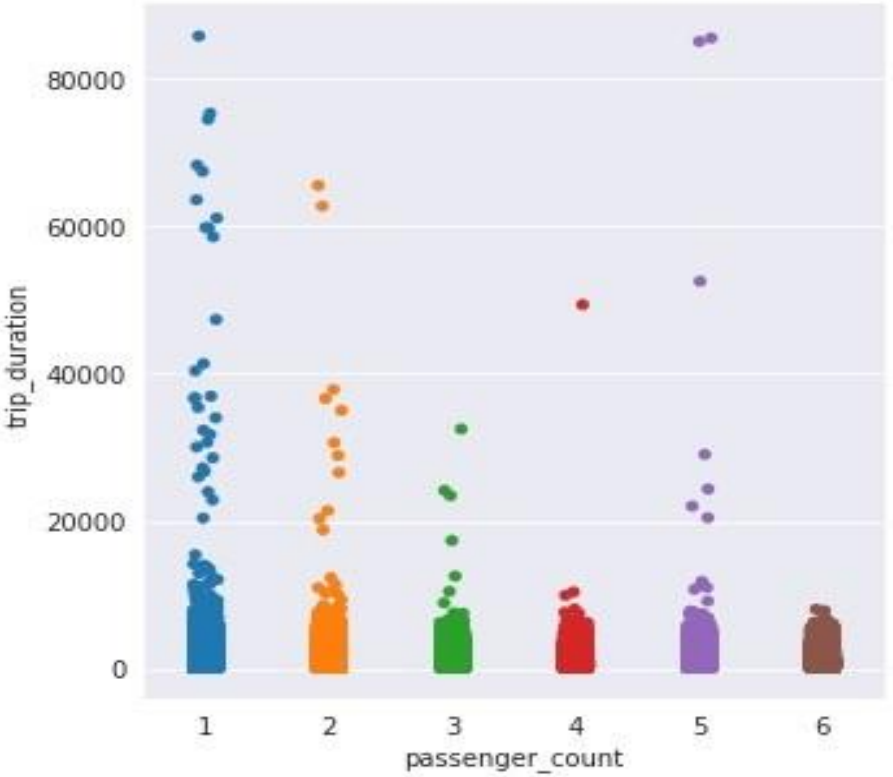


Trip duration vs month and pickup day

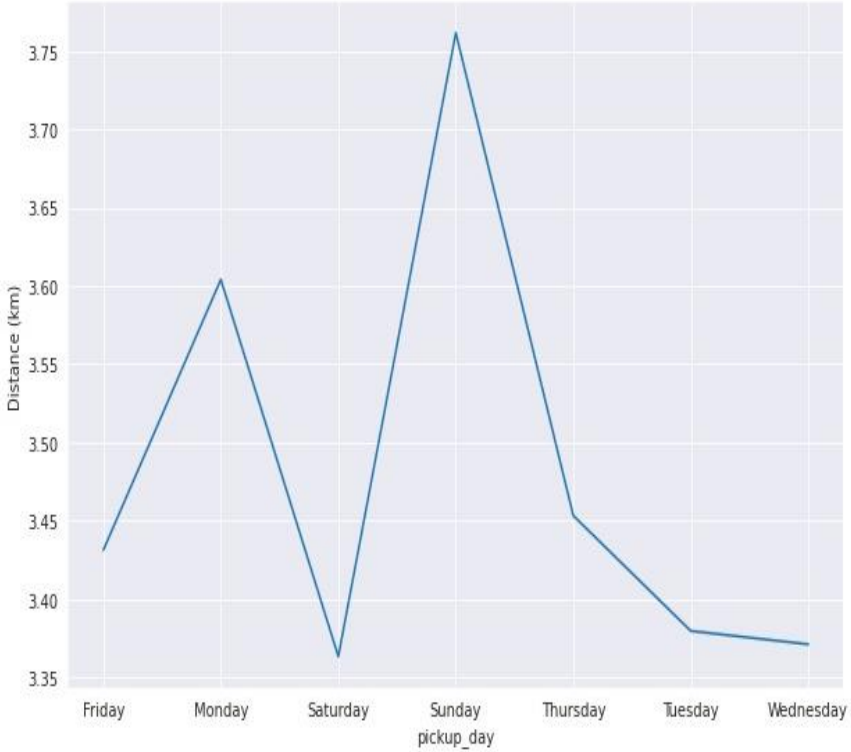


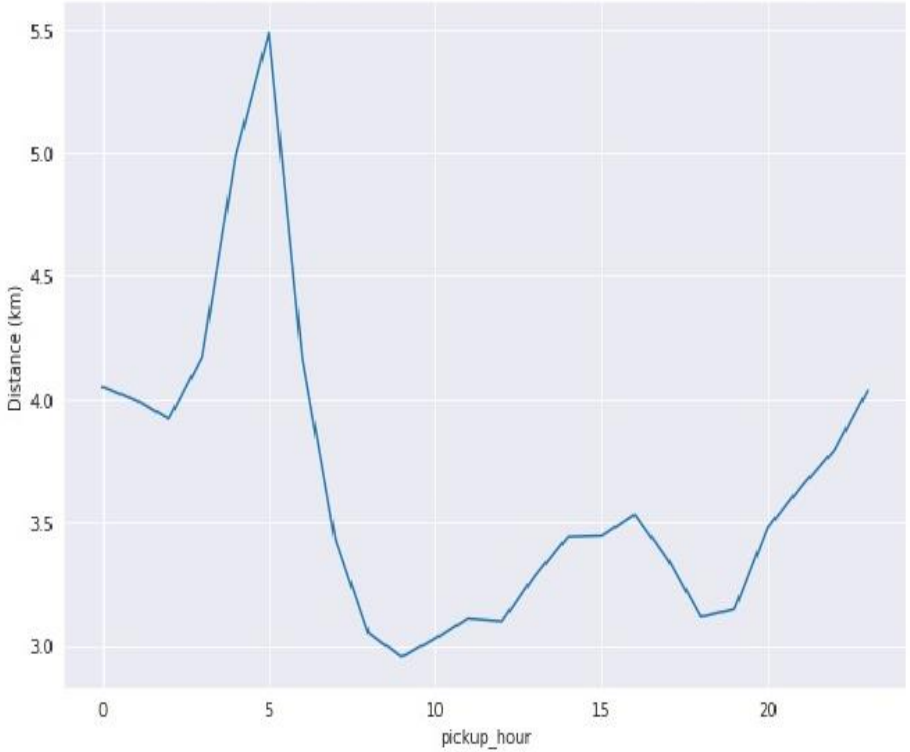
Trip duration vs pickup hour and passenger count

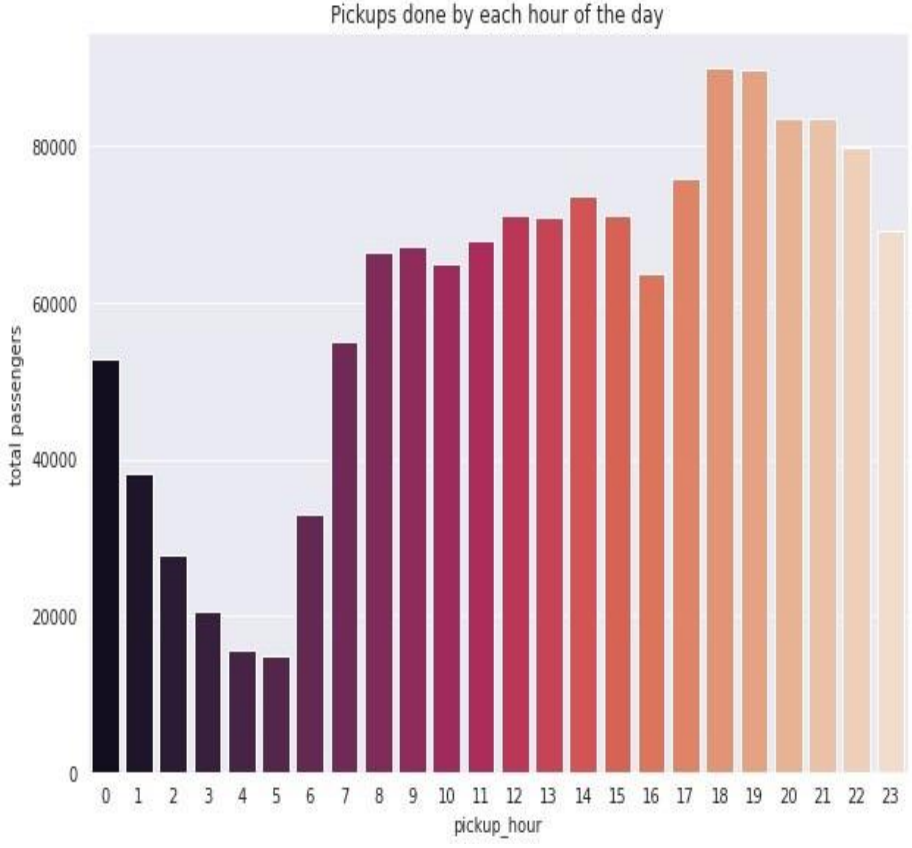


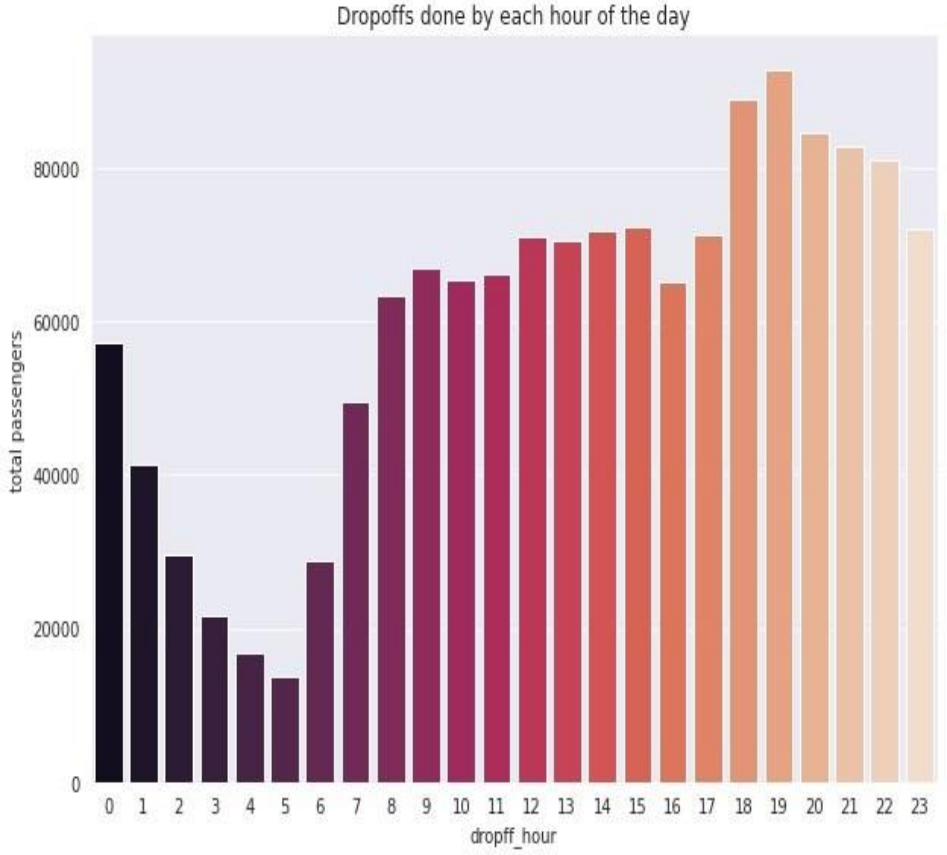


Distance vs pickup day and pickup hour






Pickups done by each hour of the day
total passengers
0
20000
40000
60000
80000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
pickup_hour


Dropoffs done by each hour of the day
total passengers
0
20000
40000
60000
80000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
dropff_hour

REGRESSION MODELS' ANALYSIS

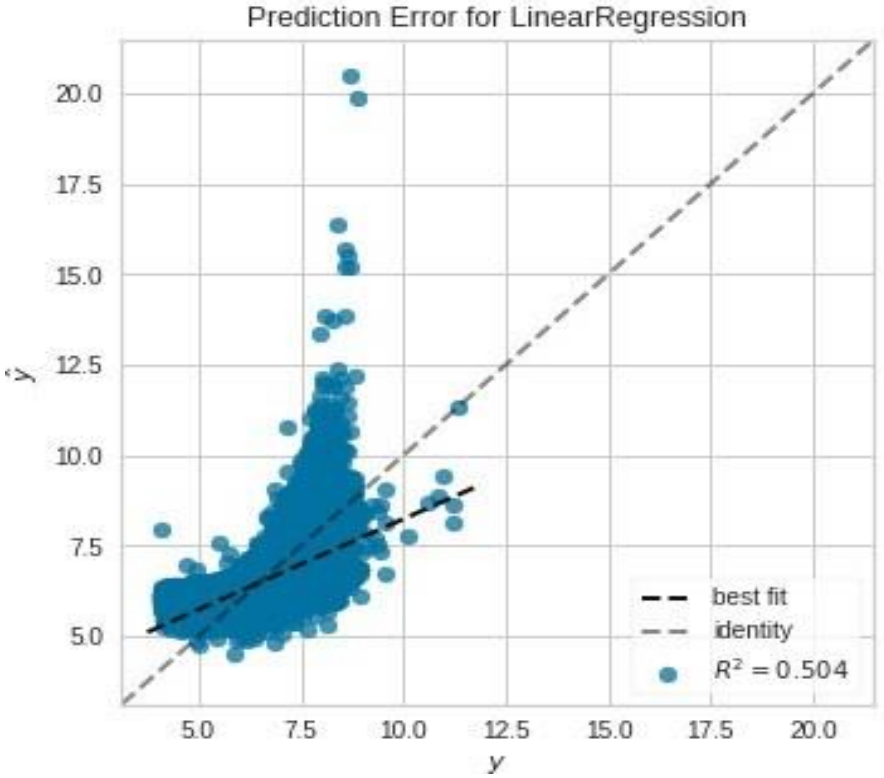
Supervised Machine Learning
Regression Analysis

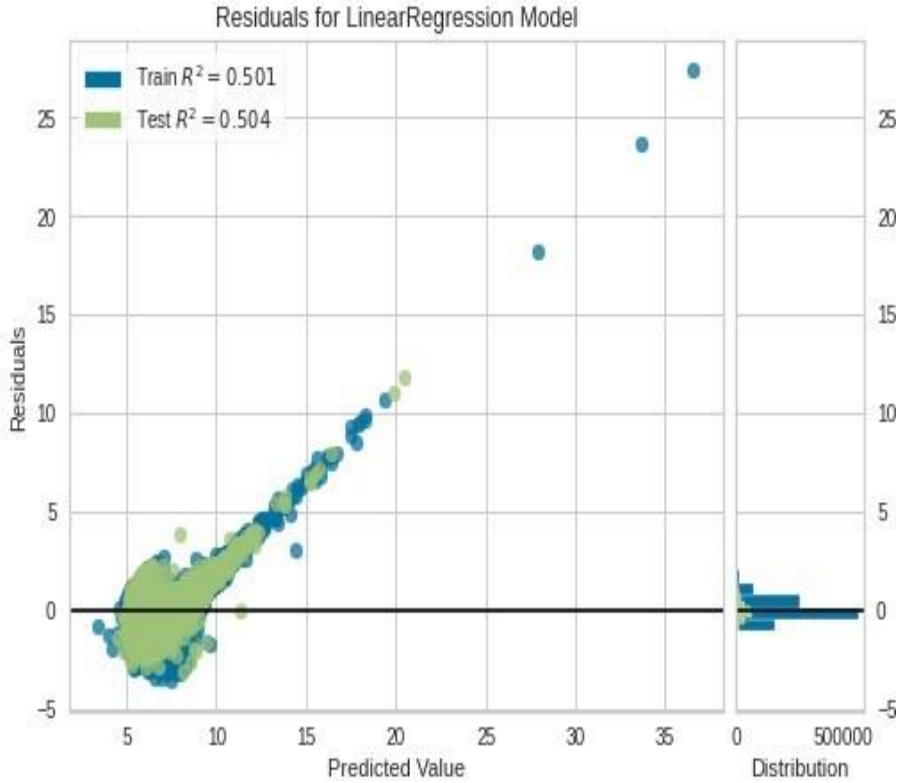
Supervised machine learning algorithms is defined by its use of labelled datasets to train algorithms to classify data or predict outcomes accurately.

Supervised learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized.

Regression is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business and many more.

Linear Regression

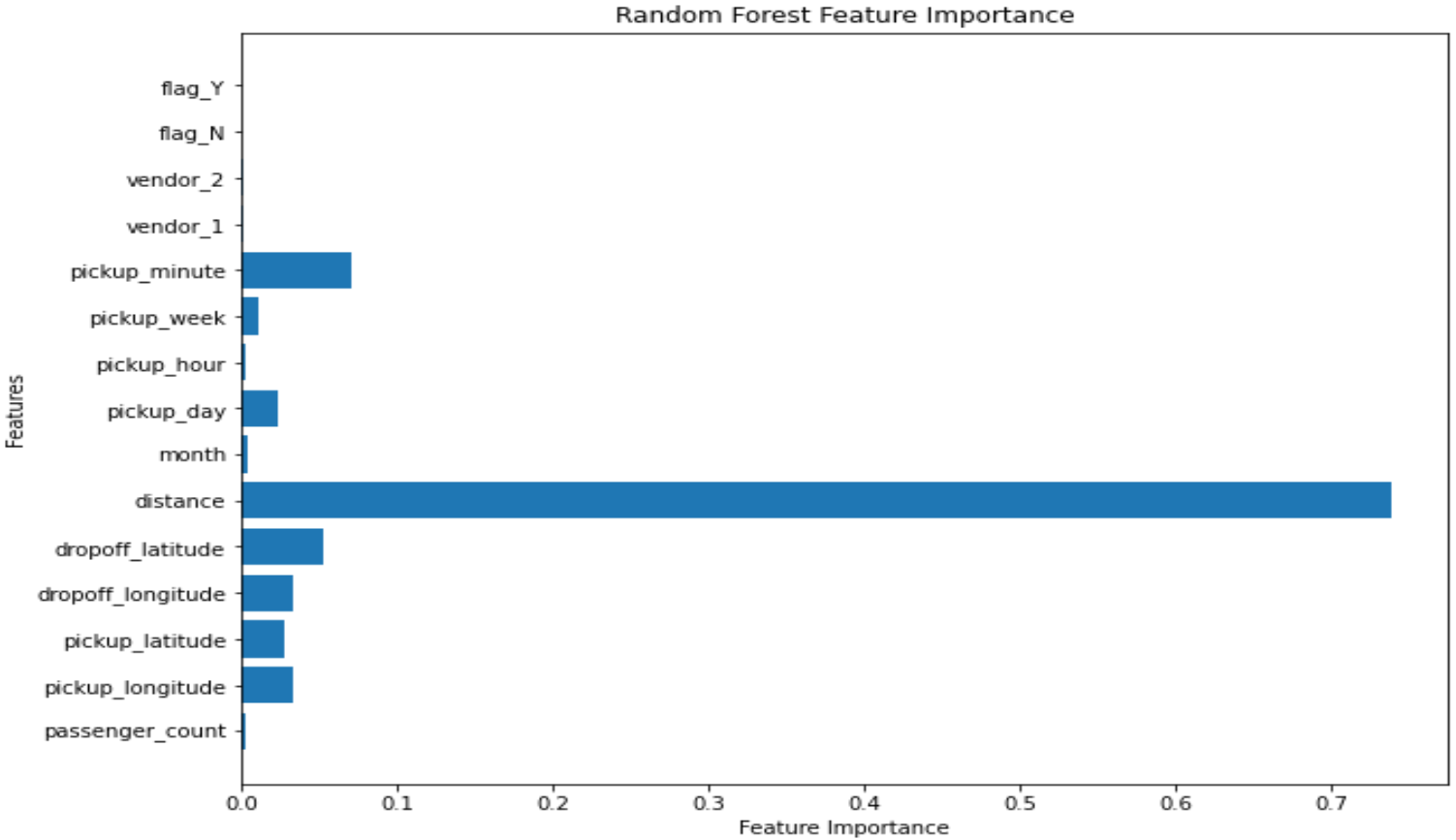




Random Forest Hyperparameters

- bootstrap=True
- criterion='mse’
- max_depth=20
- max_leaf_nodes=None
- max_samples=0.8
- min_samples_leaf=1
- min_samples_split=2
- n_estimators=80
- n_jobs=-1

Random Forest

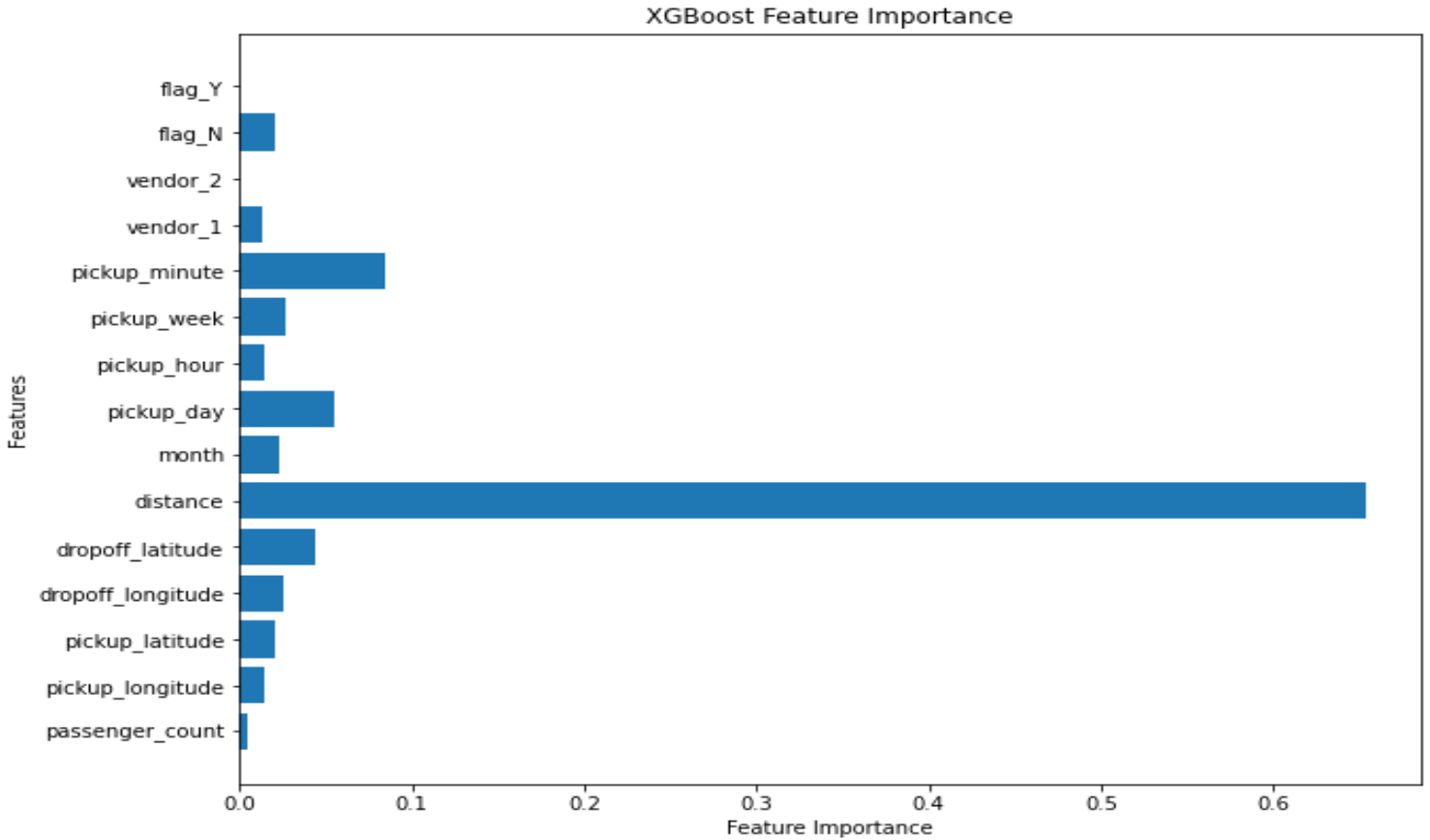


XGBoost Hyperparameters

- booster = 'gbtree’
- eta = 0.09
- max_depth = 25
- n_estimators = 100
- n_jobs = 1
- objective = 'reg:squarederror’
- random_state = 0
- subsample = 0.8

XGBoost Algorithm

XGBoost


XGBoost Feature Importance
flag_Y
flag_N
vendor_2
vendor_1
pickup_minute
pickup_week
pickup_hour
pickup_day
month
distance
dropoff_latitude
dropoff_longitude
pickup_latitude
pickup_longitude
passenger_count
Features
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Feature Importance

Performance evaluation

MODEL	ROOT MEAN SQUARED ERROR	R SQUARE	MEAN ABSOLUTE ERROR
XGBoost	0.29	0.83	0.22
Random Forest	0.32	0.80	0.24
Linear Regression	0.51	0.50	0.40

Conclusion

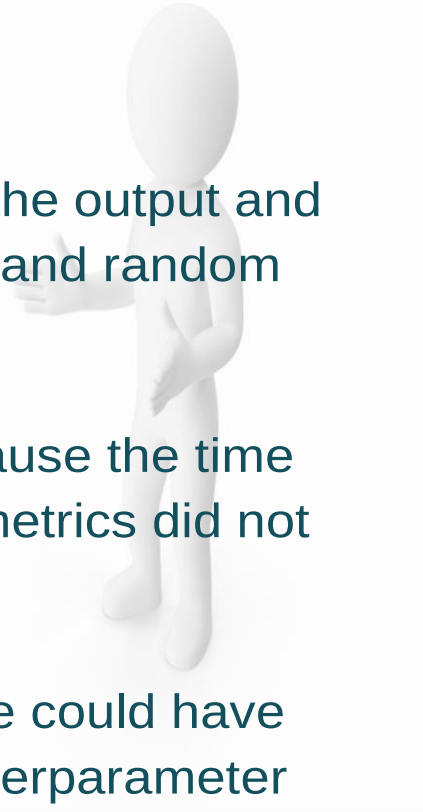
- We tried to fit various models to predict the trip duration given the dataset.
- Linear Regression, Random Forest, XGBoost.
- Linear Regression did not perform that good for the very reason that it did not fit well to the pre-assumptions that are necessary to be validated in order to fit the linear regression algorithm and to give satisfactory results.
- Random Forest came out way better than linear regression model.

Conclusion

- XGBoost proved to be much more efficient in predicting the output and better RMSE and R2 Score value than linear regression and random forest model.

- We didn't run grid search CV for modelling purpose because the time consumption for it was very high and the performance metrics did not improve much either so we sticked to train test split only.

- There's always scope for improvement. In our models we could have also used Principal component analysis, some more hyperparameter tuning and various other methods but we will save that for later as still with small hyper parameter adjustments we are doing pretty good.



THANK YOU