Capstone Project
NYC Taxi Trip Time Prediction

INTRODUCTION

- We have heard about the famous city of New York. A magnificent city, where everybody dreams to work and live.

- In this project we will be focussing on the great New York City itself, but from a slightly different angle. An angle that is very important for everyone living in that city. We will be focussing mostly on the taxis of New York City.

- Public transportation plays a very big role in every metropolitan city, especially taxis. They help commute to work and to other parts throughout the city, while making the journey comfortable for the passengers.

- As a Data Scientist, we can gather different variables and predict the taxi trip duration. We will be doing exactly that in this particular capstone project.

The Data

The data we have received is released by NYC Taxi and Limousine Commission (TLC). The data is vast and contains many important features.

The data consists of 1458644 rows, each row representing a taxi trip and 11 columns.

The columns are :-

- 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)

The Data

- 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

The trip_duration here is the dependent variable.

We can make use of other variables to create new variables and help our analysis.

Data Cleaning and Feature Engineering

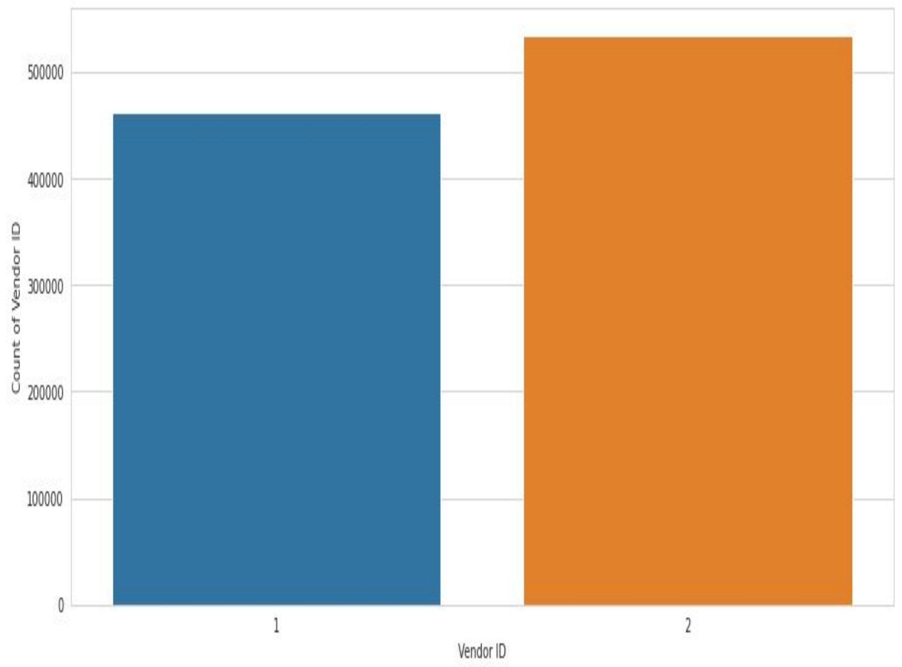
In this section we will mainly focus on cleaning the data and also add some features to the data.

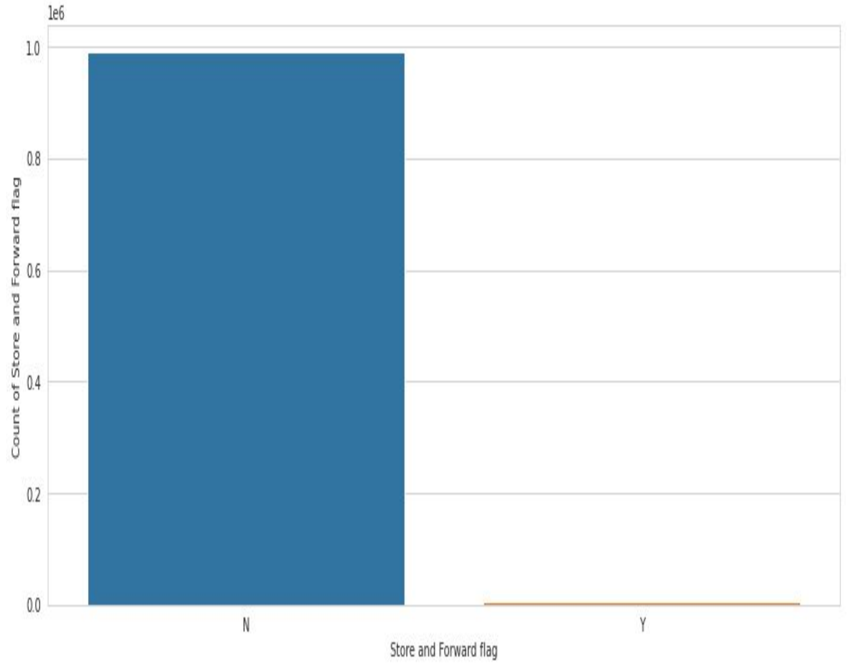
Some of the cleaning and feature engineering done are :-

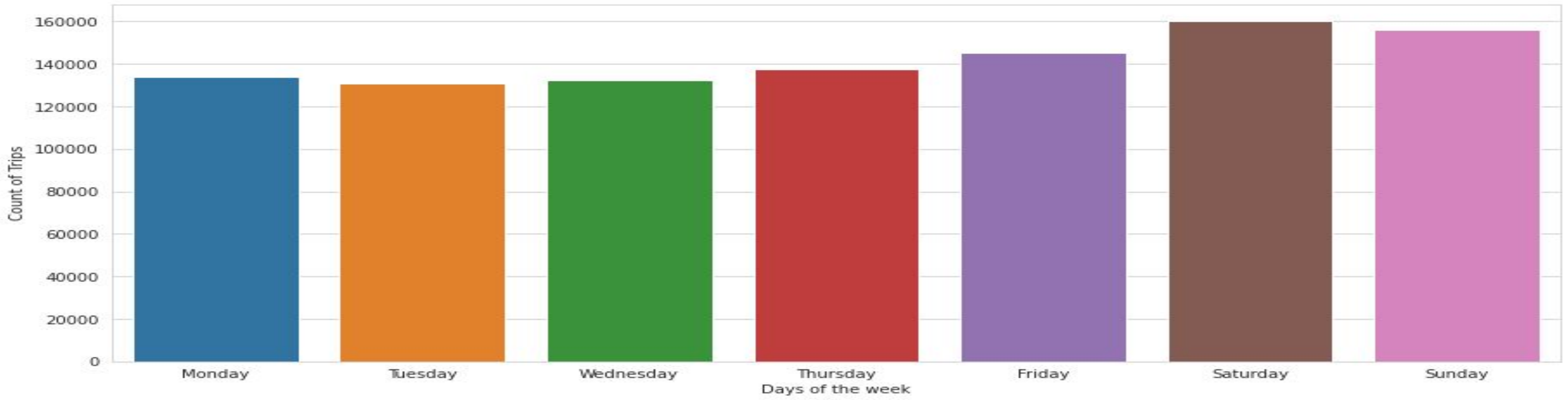
- Remove trips with no passengers.
- Convert pickup_time to datetime variable and extract important features such as year, month, date, day of week and hour.
- Extract the locations that only belong to New York City.
- Haversine distance using longitude and latitude of pickup and dropoff points.
- Remove entries with 0 distance.
- Calculate speed using distance and time, to filter out entries that have reasonable distance and durations.
- Filtering out entries with speed from 10 km/h to 60 km/h.

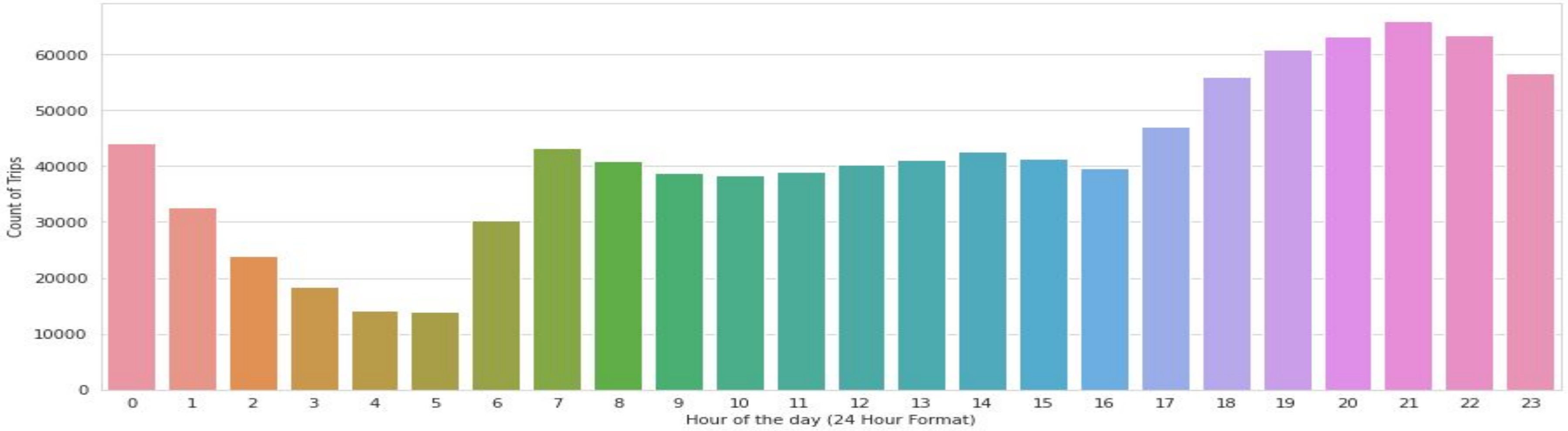
Exploratory Data Analysis (EDA)

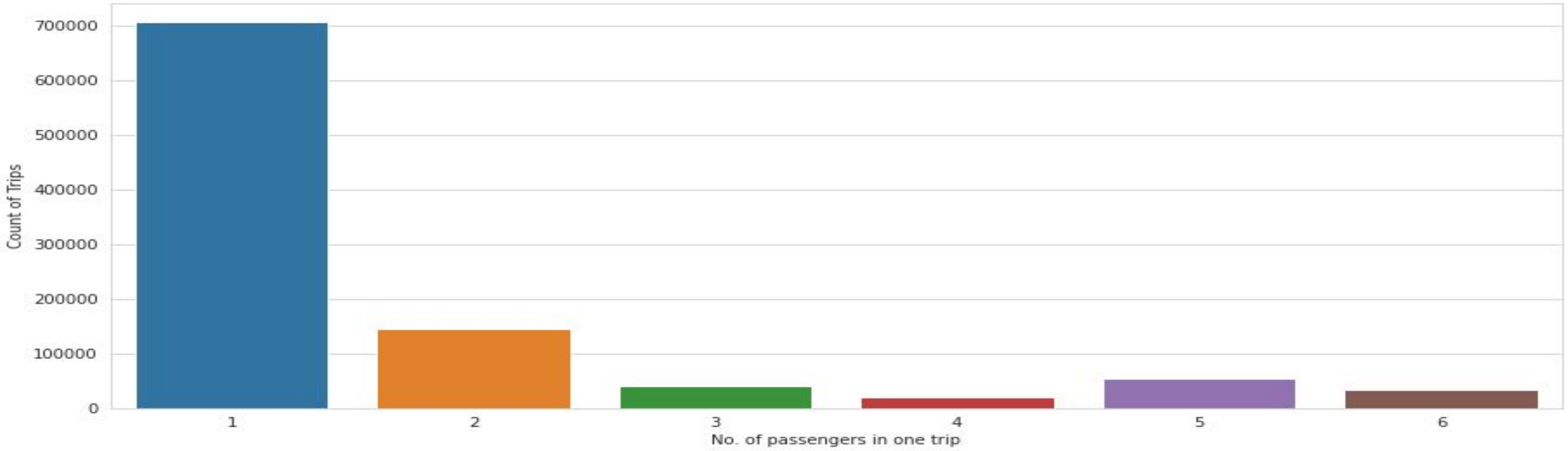
Univariate Analysis

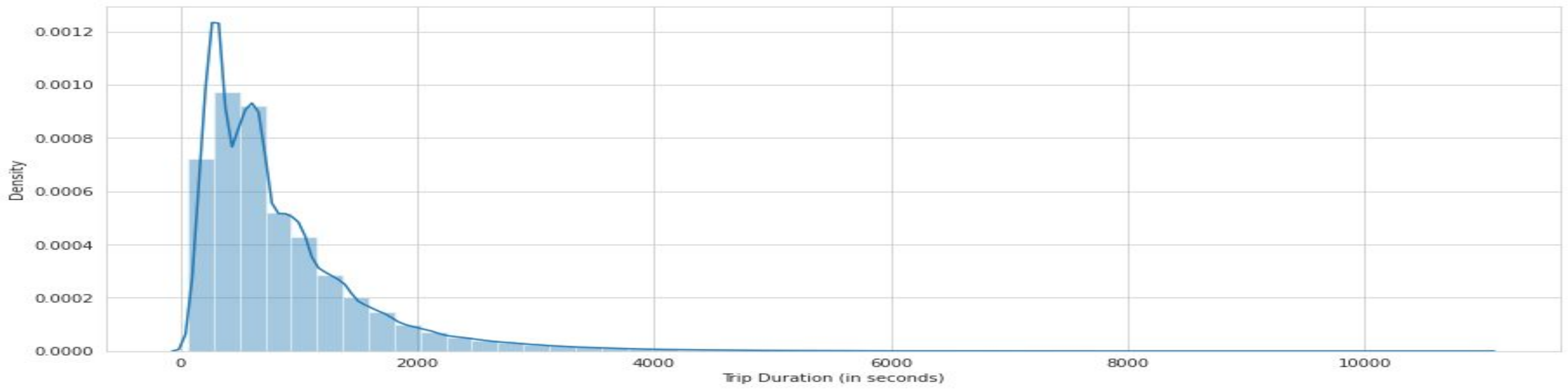


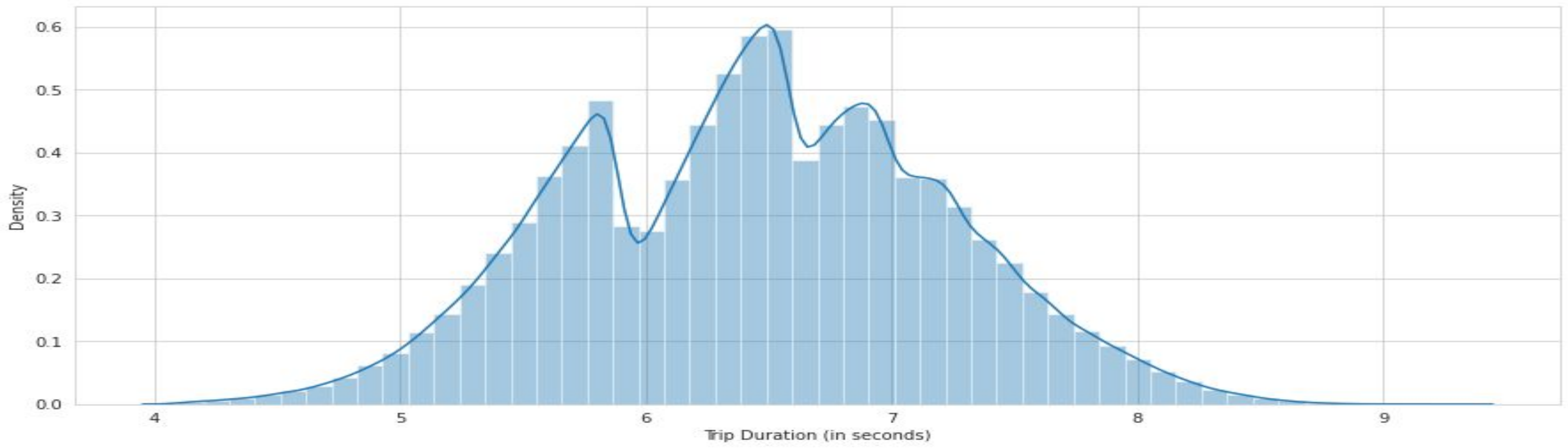



160000
140000
120000
100000
80000
60000
40000
20000
0
Count of Trips
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Days of the week

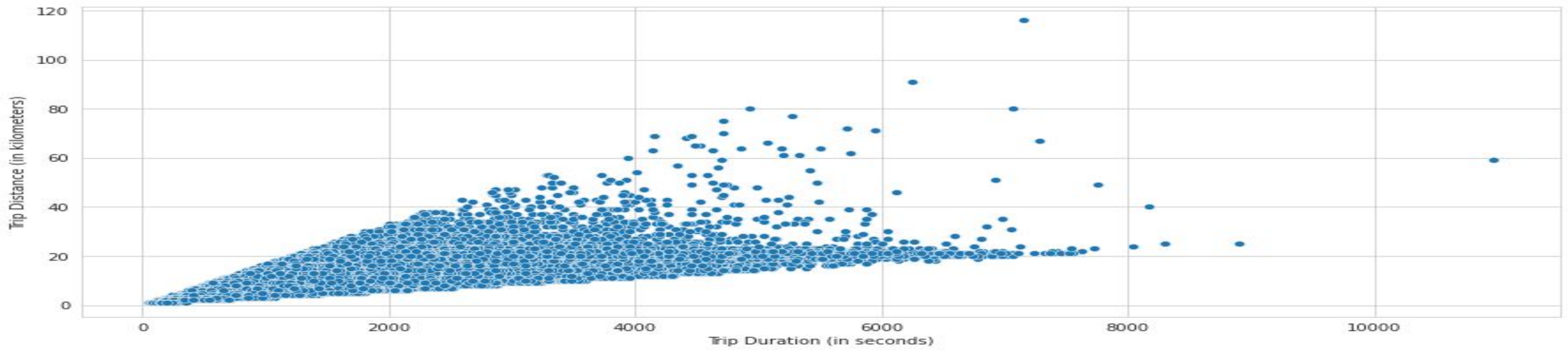

60000
50000
40000
30000
20000
10000
0
Count of Trips
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Hour of the day (24 Hour Format)

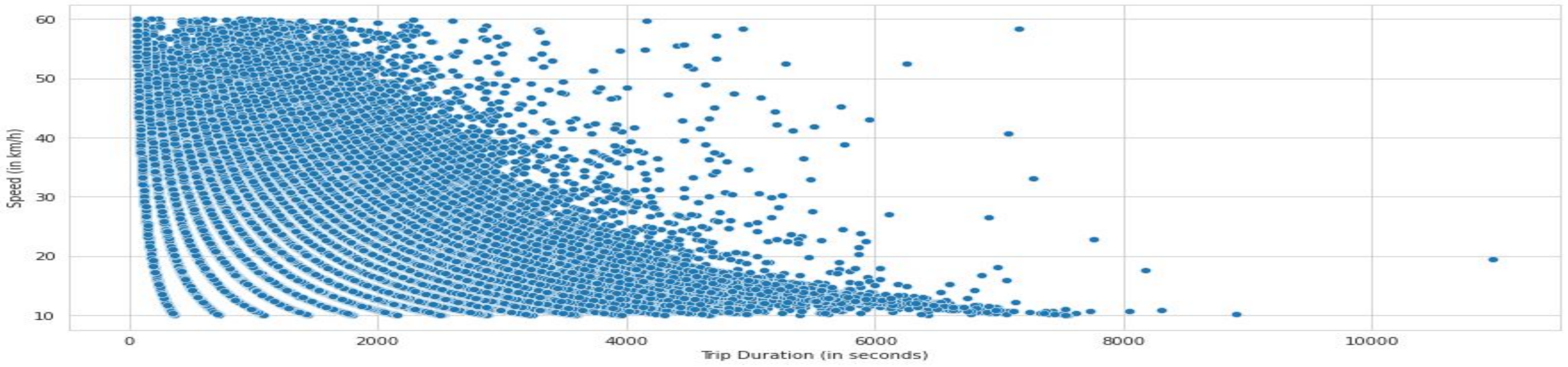

Count of Trips
700000
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
No. of passengers in one trip

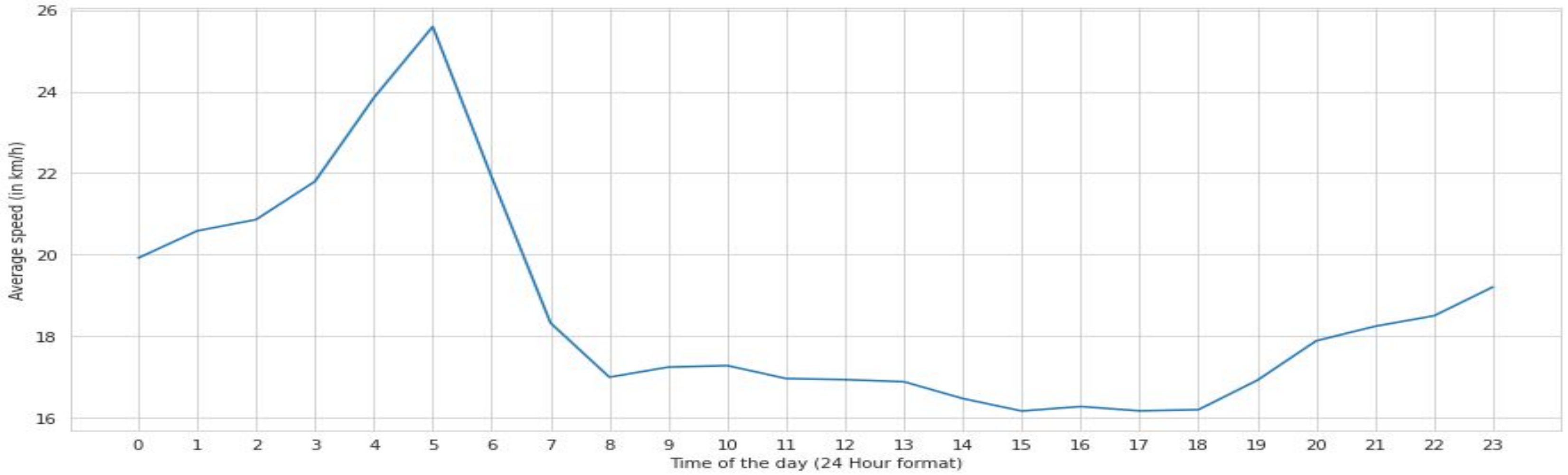

Density
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
0
2000
4000
6000
8000
10000
Trip Duration (in seconds)


Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
4
5
6
7
8
9
Trip Duration (in seconds)

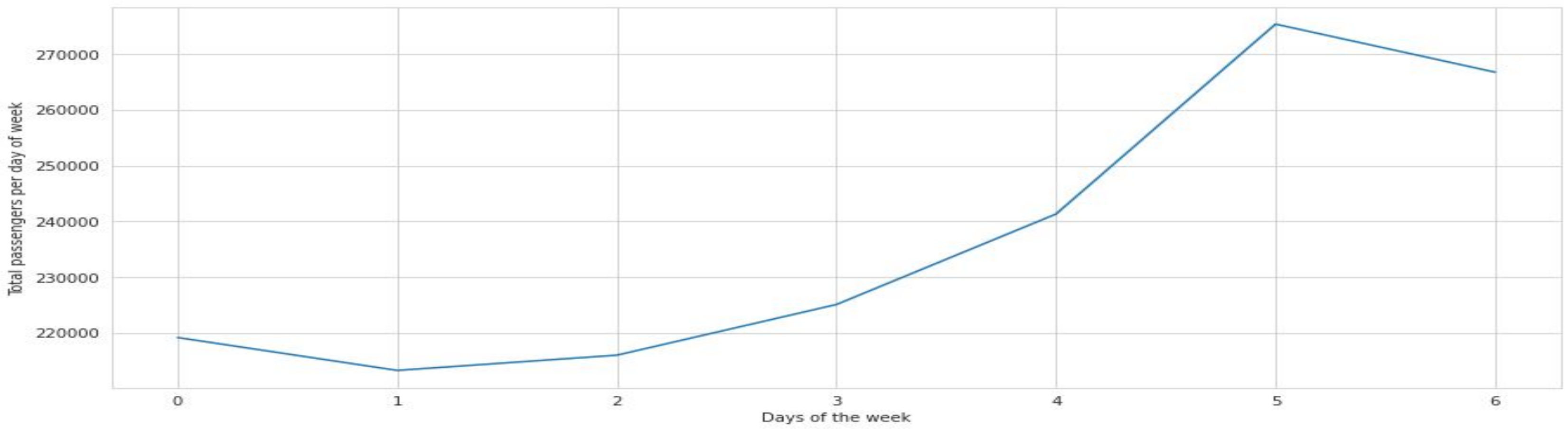
Bivariate Analysis







Average speed (in km/h)
26
24
22
20
18
16
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Time of the day (24 Hour format)



Total passengers per day of week
270000
260000
250000
240000
230000
220000
0 1 2 3 4 5 6
Days of the week

Model Building and Evaluation

For this problem, the model being chosen is that of a decision tree.

Further ensembling techniques such as Random Forest, Gradient Boosting and XGB Regressor have also been experimented with.

Metrics such as Root Mean Squared Error (RMSE) and R2 score have been used for evaluation.

The model and their scores are mentioned as follows :-

- Decision tree

  RMSE for train data is : 9.069802740810407e-06

  RMSE for test data is : 0.3260150044363

  r2 score for train data is : 0.9999999998590093

  r2 score for test data is : 0.8177274695508714

Model Building and Evaluation

- Random Forest

  RMSE for train data is : 0.08725229498355119

  RMSE for test data is : 0.22885095768822386

  r2 score for train data is : 0.9869518635821268

  r2 score for test data is : 0.9101844539884814

- Gradient Boosting

  RMSE for train data is : 0.2819623959856205

  RMSE for test data is : 0.2817815104829239

  r2 score for train data is : 0.8637372481460092

  r2 score for test data is : 0.8638332626167505

Model Building and Evaluation

- XGB Regressor

  RMSE for train data is : 0.28167803390903184

  RMSE for test data is : 0.28147208391128886

  r2 score for train data is : 0.864011954424334

  r2 score for test data is : 0.8641321500279132

As seen from the above models :-

- Decision tree overfits.
- Random forest is more consistent and also gives equal importance to features.
- Gradient Boosting and XGB Regressor have similar results.
- Due to the size of the data ensembling techniques are computationally expensive.

Hyperparameter Tuning

Here hyperparameter tuning is done with the help of Grid Search CV.

2 best models i.e XGB and Random Forest have been considered for hyper parameter tuning in order to fine tune and further improve the models.

The results of hyperparameter tuning are as follows :-

- XGB Regressor

  RMSE for train data is : 0.17833040894878405

  RMSE for test data is : 0.2199104111004795

  r2 score for train data is : 0.9454938406353761

  r2 score for test data is : 0.917065042415389

Hyperparameter Tuning

- Random Forest

  RMSE for train data is : 0.08578403189720486

  RMSE for test data is : 0.2271452012549775

  r2 score for train data is : 0.9873835424488463

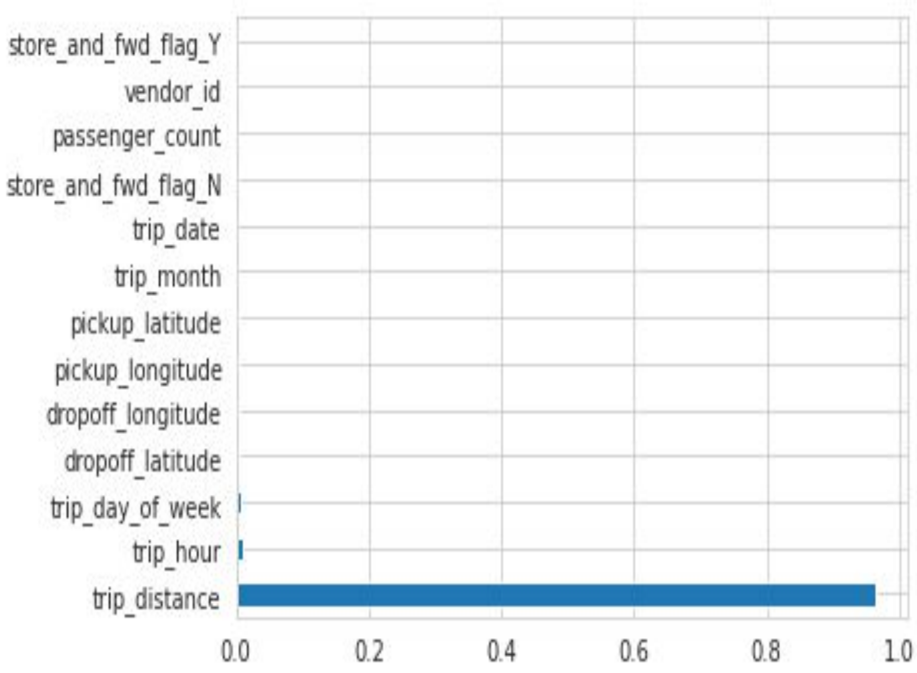
  r2 score for test data is : 0.9116239938873824

As seen above :-

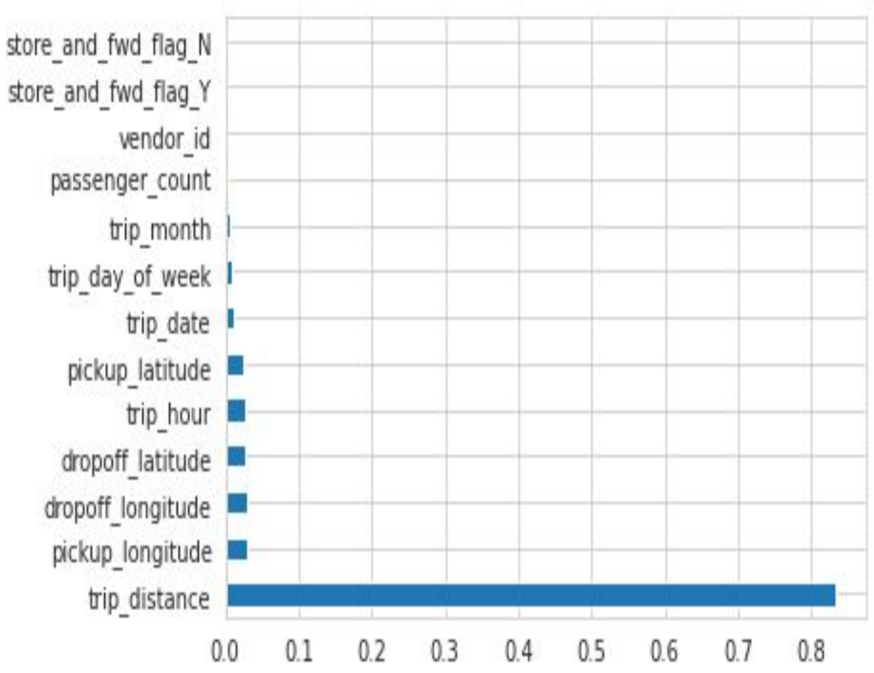
- XGB after hyperparameter tuning gives better results but lacks in giving importance to more features.
- Random forest has similar results to before but gives importance to more features.
- Both models are now giving similar results.
- Both models take time to compute.

Hyperparameter Tuning Feature Importance

XGB Regressor



Random Forest



Hyperparameter Tuning

From the results seen above we can understand that :-

- Due to the data being large, model fitting takes a long time.
- Decision tree being a greedy algorithm overfits.
- Ensembling techniques work well.
- Random Forest gives importance to more variables compared to other models.
- Random Forest before and after hyperparameter tuning performs consistently.
- Random Forest with 1 hyperparameter achieves similar accuracy of XGB with 3 hyperparameters.
- Random Forest can do even better with more n_ estimators.

Keeping in consideration the above points, I certainly feel Random Forest is working well compared to other models, though being computationally expensive it is providing more consistent results.

Conclusion

- In conclusion, this capstone project has given me immense understanding about real world problems. I have done my best to apply concepts that I have learnt over the weeks to solve the problem in hand.
- There can be various other models that could be used that may even perform better than the ones being used here. Hence, there are possibilities of improving the model further. Other features such as weather and traffic can be further added to improve the model.
- There were challenges in terms of new concepts, new fields of study and also working on a large dataset was time consuming. These challenges have brought in more interest to discover more ways of being efficient and also it has built up curiosity in me to learn more of what data science has to offer.

AI

THANK YOU!