EDA on NYC Taxi Records

Presented by

B Parashuram Achar

CONTENT

- **Introduction**
- **Problem Statement**
- **Understanding the Dataset**
- **EDA**
- **Challenges Faced**
- **Conclusion**

Introduction

- **Taxicabs are the only vehicles that have the right to pick up street-hailing and prearranged passengers anywhere in New York City.**
- **Increasing popularity of app-based taxi such as lyft or uber and there competitive pricing levels made user decisive to choose based on trip pricing and duration.**
- **In the present supervised Machine Learning (ML) – regression algorithm, will try to predict the over all trip duration of the average yellow taxi service of NYC Taxi and Limousine Commission (TLC).**
- **Such kind of model building for prediction, will help customers to select the taxi based on trip duration and driver to select optimum route to their destination.**
- **Thus, will further help service providers to improve their service more, helping customer.**

Problem Statement

- An upcoming taxi operation in NYC, you are tasked to use the 2023 taxi trip data to uncover insights that could help optimise taxi operations. The goal is to analyse patterns in the data that can inform strategic decisions to improve service efficiency, maximise revenue, and enhance passenger experience.

In the dataset there are over all *1458644 & 11*, of rows and columns

- 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

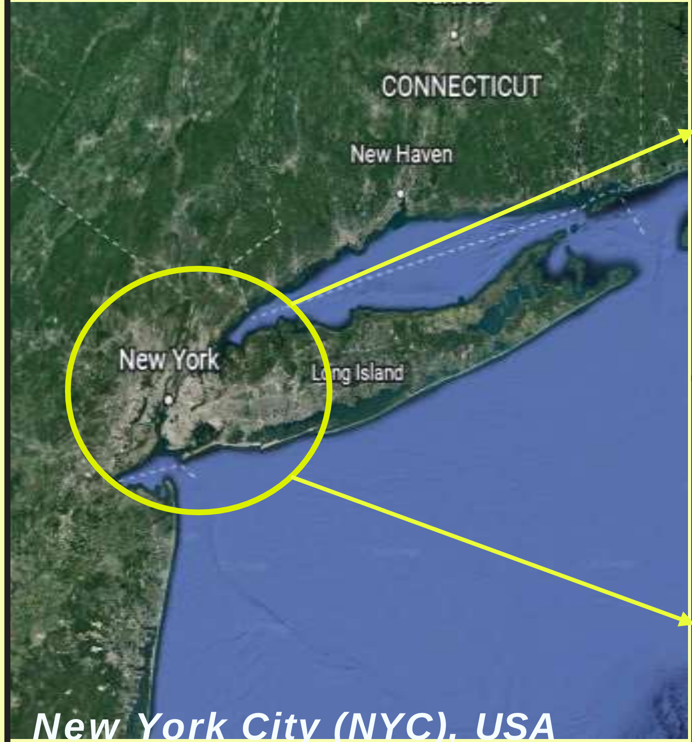
Unique values and data types of the different variables identified

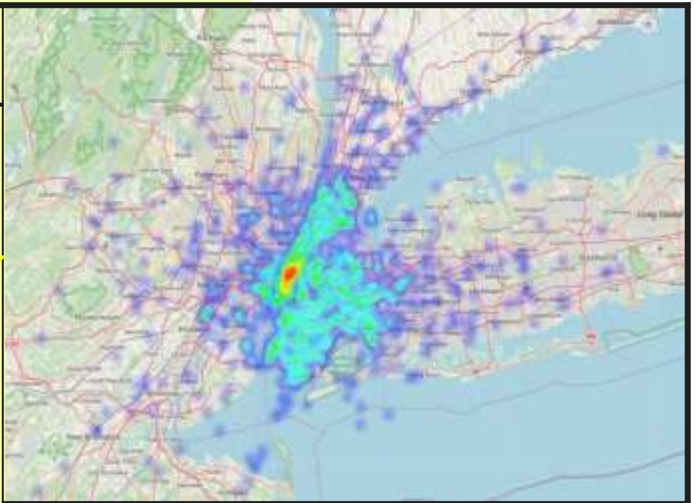
```
Total Unique Values in id - 1458644
Total Unique Values in vendor_id - 2
Total Unique Values in pickup_datetime - 1380222
Total Unique Values in dropoff_datetime - 1380377
Total Unique Values in passenger_count - 10
Total Unique Values in pickup_longitude - 23047
Total Unique Values in pickup_latitude - 45245
Total Unique Values in dropoff_longitude - 33821
Total Unique Values in dropoff_latitude - 62519
Total Unique Values in store_and_fwd_flag - 2
Total Unique Values in trip_duration - 7417
```

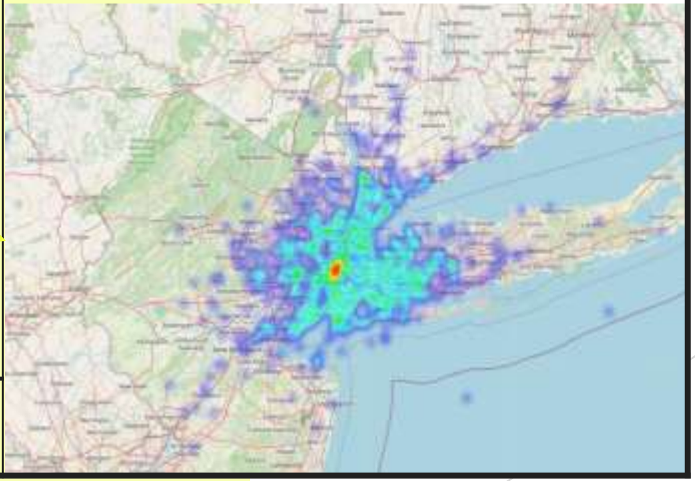
```
RangeIndex: 1458644 entries, 0 to 1458643
Data columns (total 11 columns):
 #   Column              Non-Null Count    Dtype
---  ------              --------------    -----
 0   id                  1458644 non-null  object
 1   vendor_id           1458644 non-null  int64
 2   pickup_datetime     1458644 non-null  object
 3   dropoff_datetime    1458644 non-null  object
 4   passenger_count     1458644 non-null  int64
 5   pickup_longitude    1458644 non-null  float64
 6   pickup_latitude     1458644 non-null  float64
 7   dropoff_longitude   1458644 non-null  float64
 8   dropoff_latitude    1458644 non-null  float64
 9   store_and_fwd_flag  1458644 non-null  object
 10  trip_duration       1458644 non-null  int64
dtypes: float64(4), int64(3), object(4)
memory usage: 122.4+ MB
```

Location Map of NYC, USA

Pickup Heatmap Location





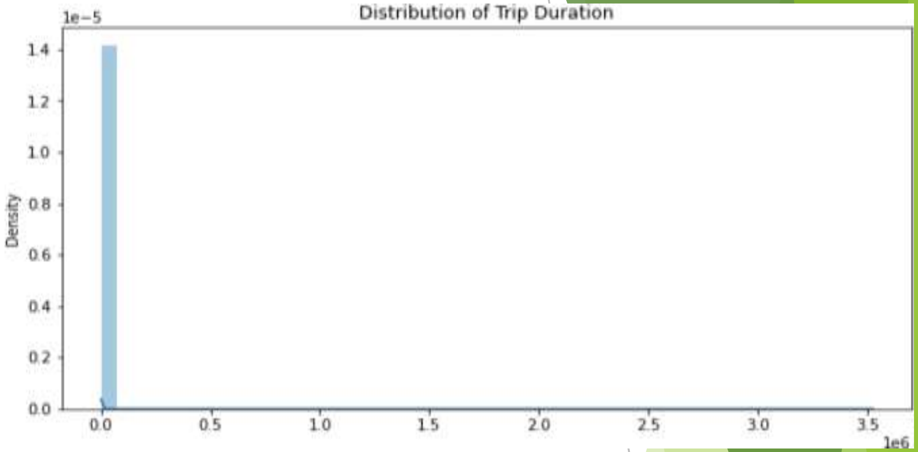


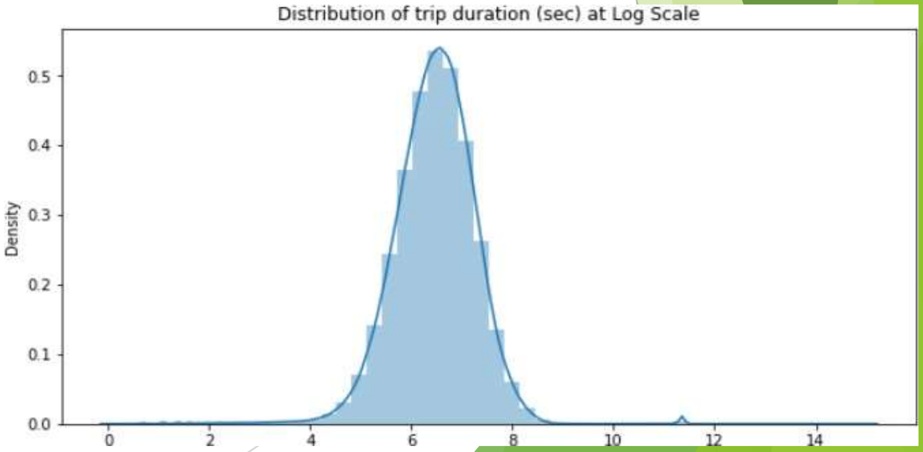
Drop-off Heatmap Location

Added features and its types in the list

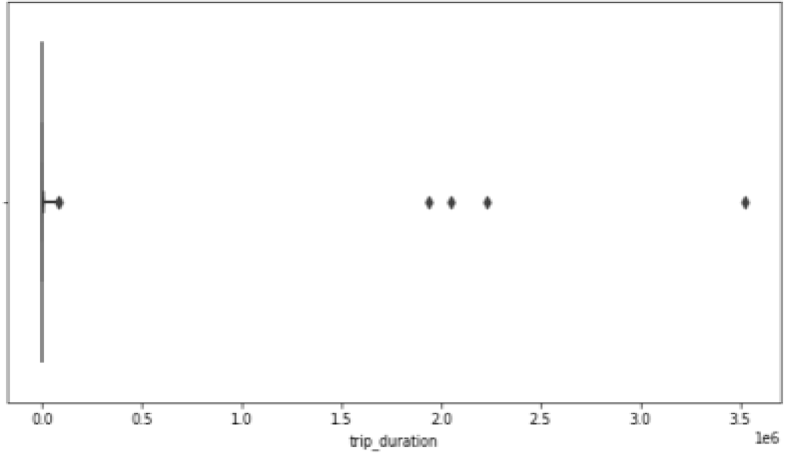
	index	0
0	id	object
1	vendor_id	int64
2	pickup_datetime	datetime64[ns]
3	dropoff_datetime	datetime64[ns]
4	passenger_count	int64
5	pickup_longitude	float64
6	pickup_latitude	float64
7	dropoff_longitude	float64
8	dropoff_latitude	float64
9	store_and_fwd_flag	object
10	trip_duration	int64
11	pickup_hour	int64
12	pickup_weekday	object
13	pickup_month	int64
14	pickup_day_num	int64
15	dropoff_weekday	object
16	distance	float64
17	speed	float64
18	pickuptime_of_day	object

Distribution of dependent variable after using log transform

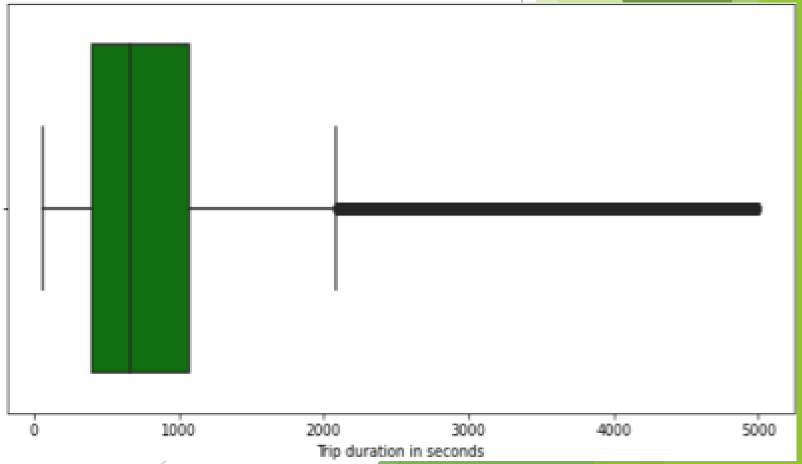




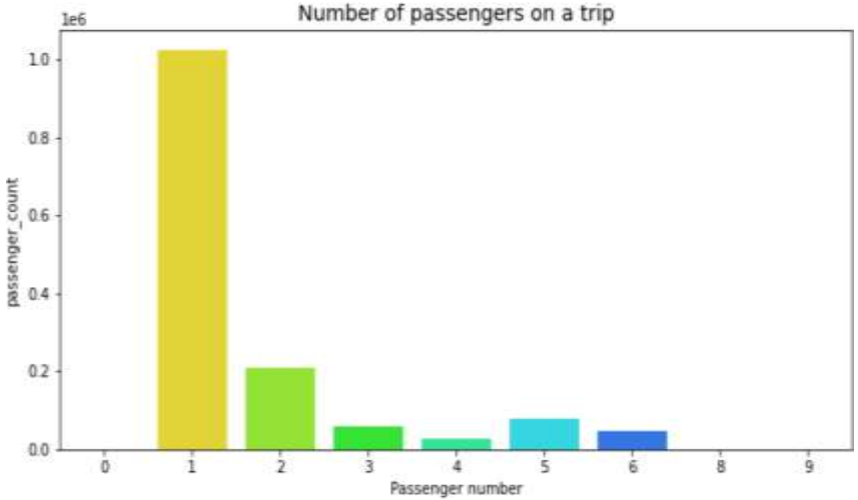
Dependent variable distribution



- **After treatment, most trip durations completed within 10-20 min and observed trips took 0-30 min (1800 seconds).**
- **Outliers present in the dependable variable**

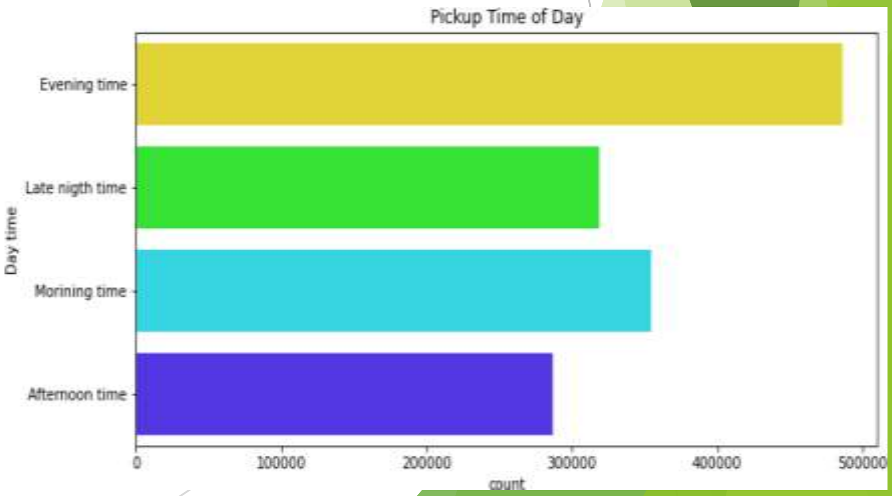


Distribution of different variables

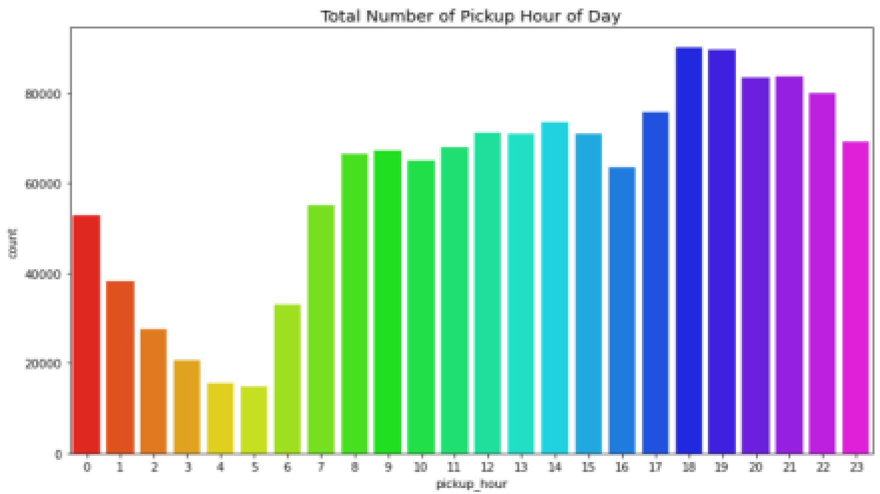


- **Number of passengers travelling during a trip**

- **Duration of a day time when maximum number of passenger travels**

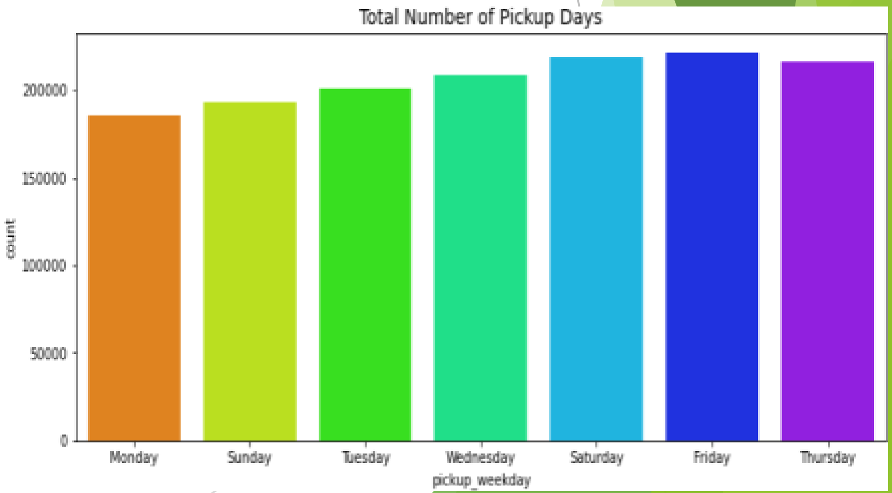


Distribution of different variables

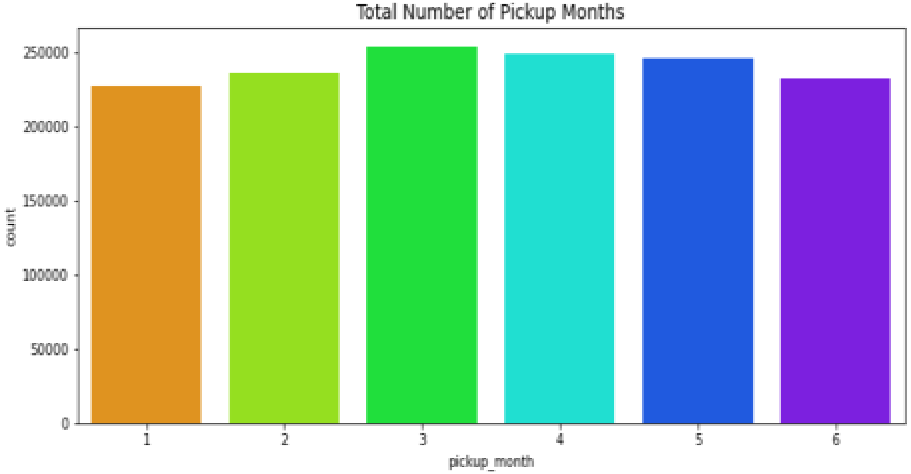


- Plot it can be clearly viewed that the timing between 6.00 pm to 7.00 pm in evening are the pick time for travelling.
- As this is the time when lot of working class people and market going people prefer to travel.

- Distribution of days when it is maximum travelling take place.

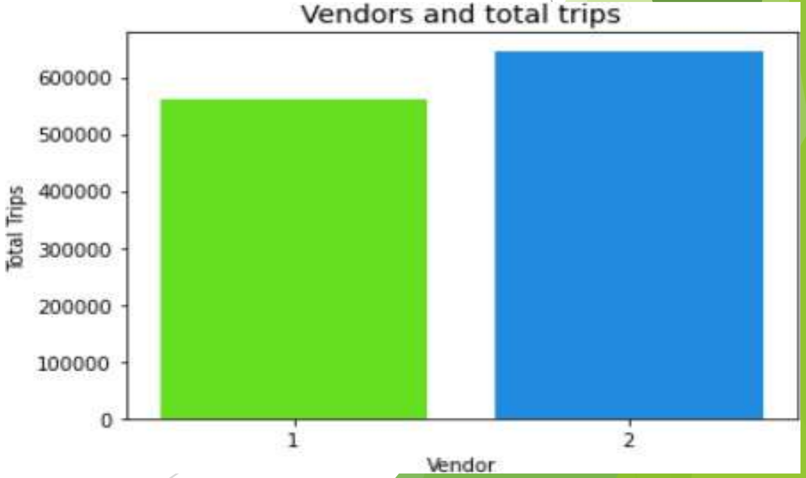


Distribution of different variables

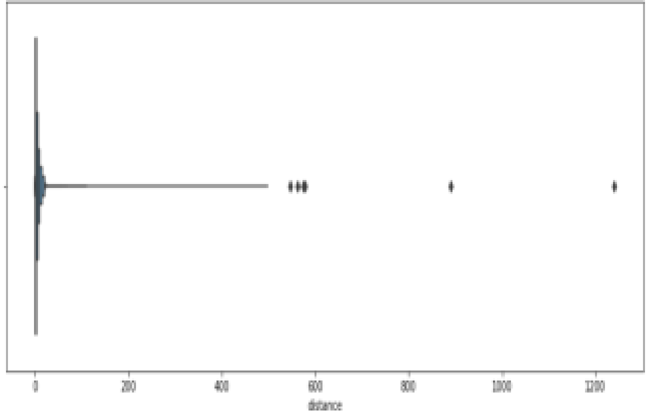


- **Number of trips per month in which it can be seen that not much change has been observed.**

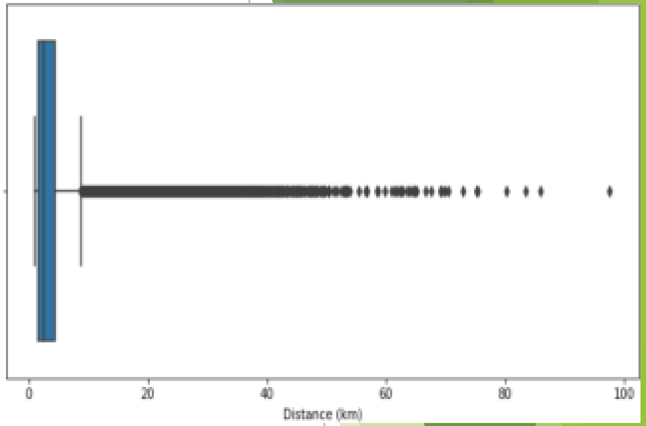
- **Vendor and total trip distribution just provides a small variation between two types.**

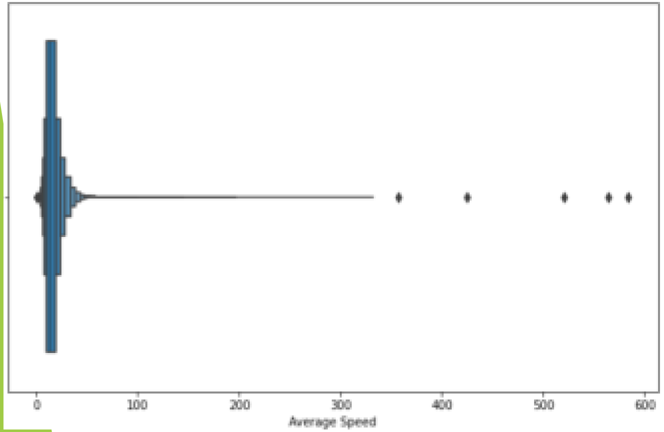


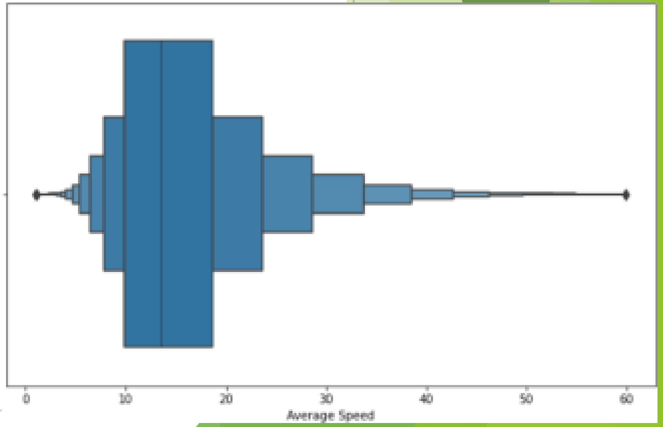
Distribution of different variables



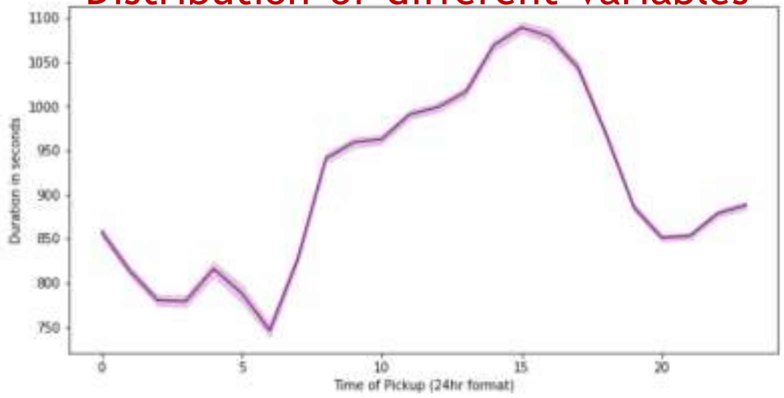
- First plot showing the outliers of speed and distance.
- And after treatment, the distance variable updated between 0 to 100 km.
- And for speed, its variable updated to near by 60 km/h (as per the traffic rule of NYC speed limit is 40 km/h).





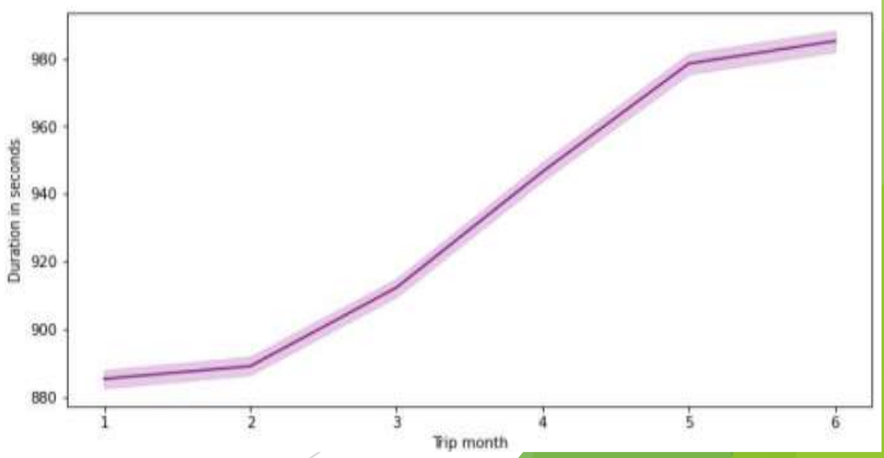


Distribution of different variables

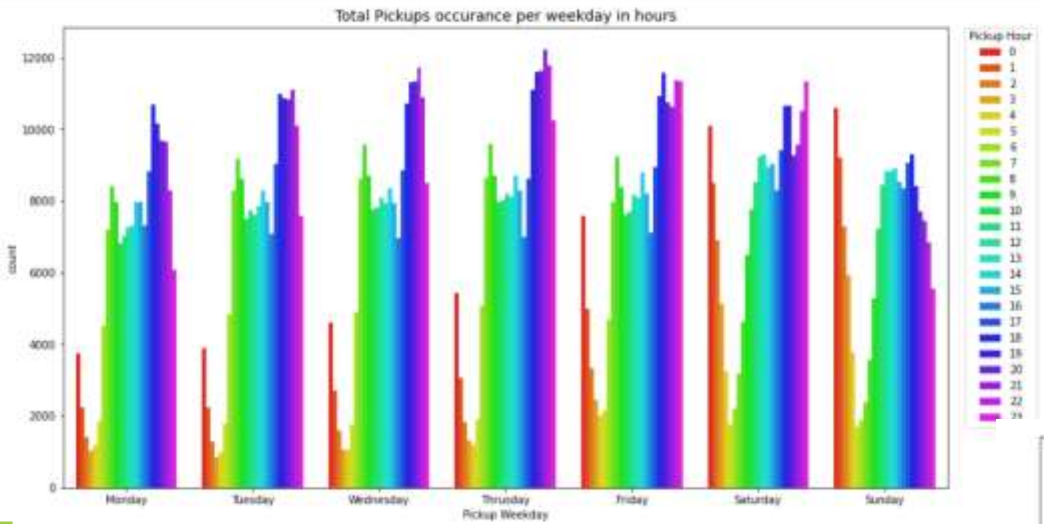


- **Trip duration reaches its pick around 3pm.**
- **And lowest at 6 am in early morning.**

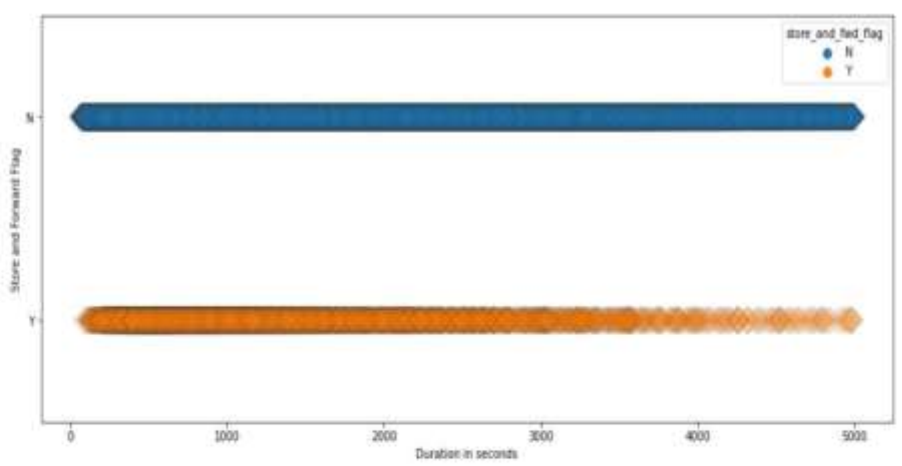
- **Showing the plot trip duration in following of 6 months in total.**
- **It is lowest at its starting month.**
- **But the picks up suddenly in the 5th month of the year.**



Distribution of different variables

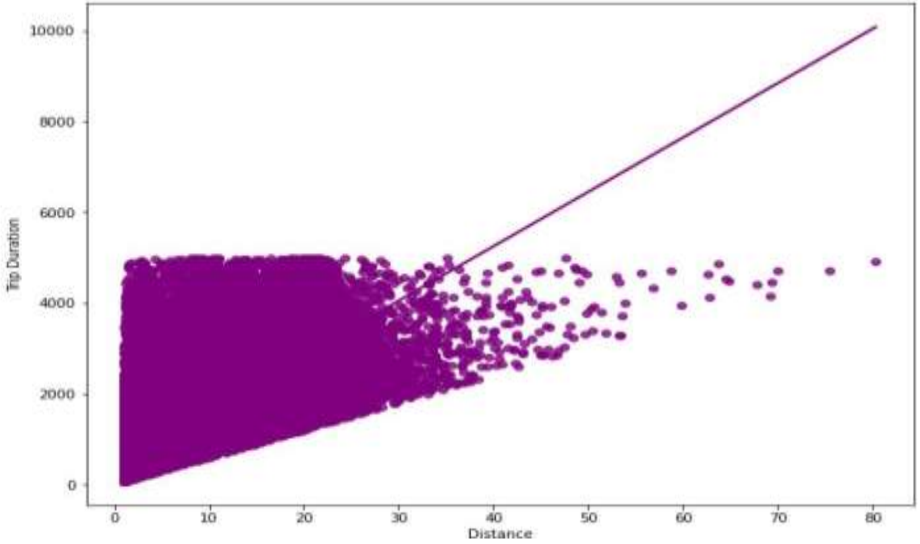


- **Plot shows, lowest trip number can be seen around 6am in early morning.**
- **Again in all weekdays maximum number of pick hours can be observed around between 3pm to 4pm, with some variation in all weekdays.**

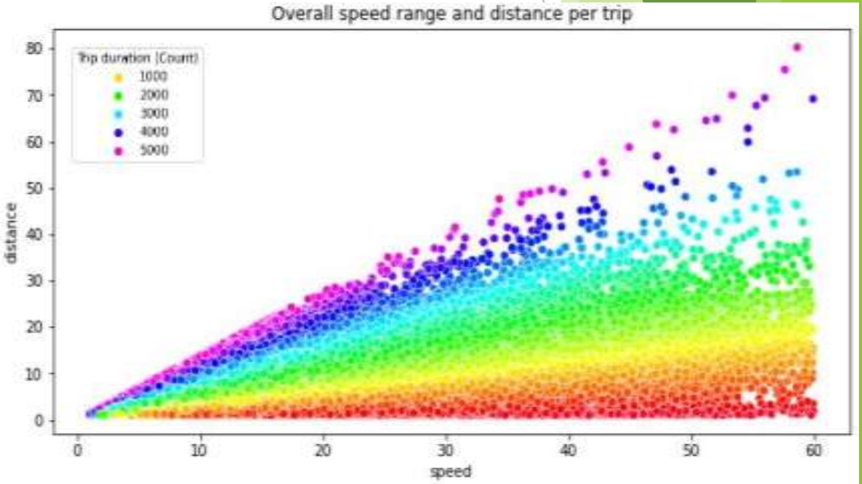


- **Not much difference between Y and N**

Distribution of different variables

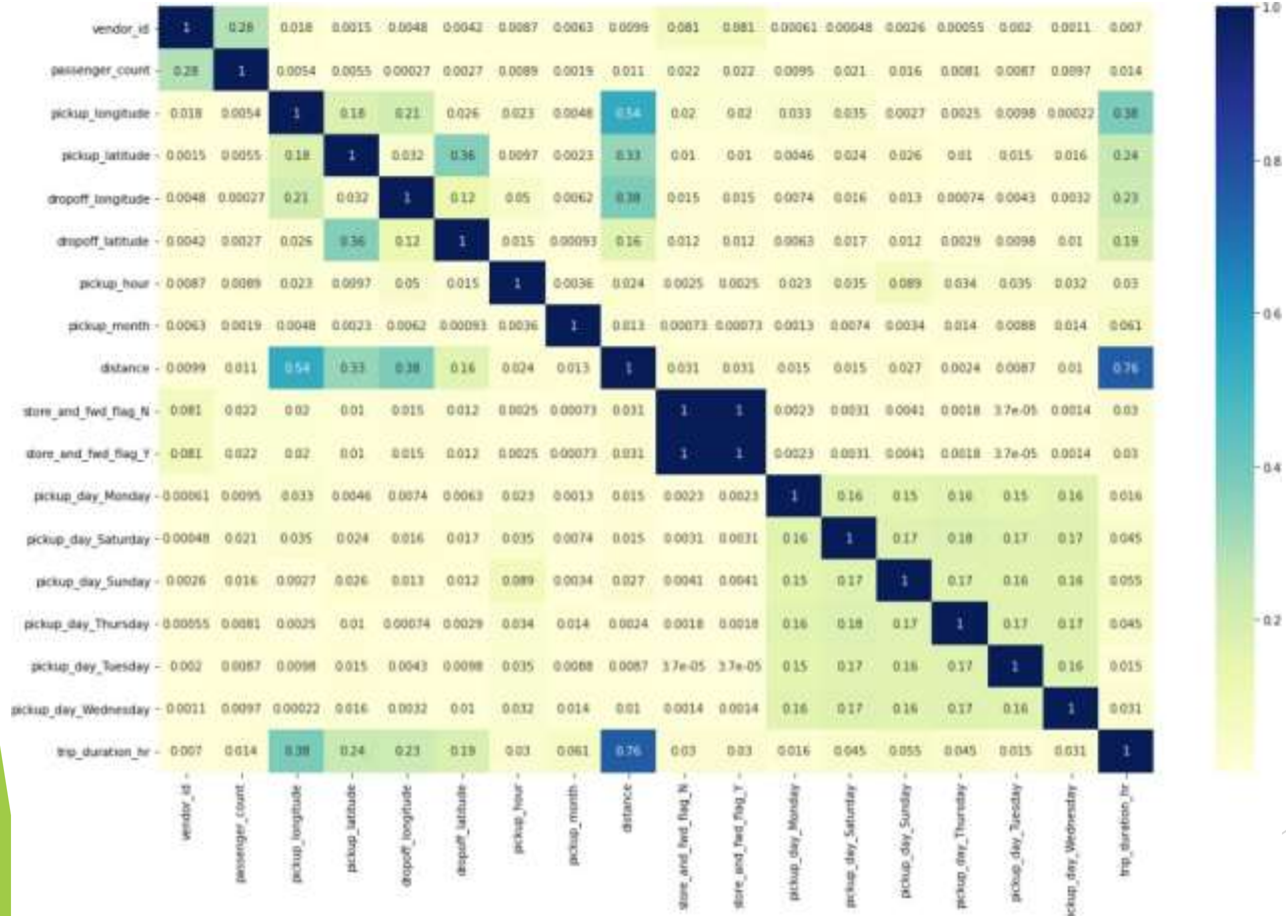


- **Regression line showing some linear relation between trip duration and distance.**



- **It can be observed at few values the speed at 60 km/hr travels the maximum distances.**

Feature Engineering



Correlation

- The plot does not show much correlation between different independent variables with dependent variable.
- Only distance and pickup_longitude shows correlation.
- Else not much correlation has been observed.

Evaluation Metrics & Types

- **Mean Absolute Error(MAE)**

MAE is a very simple metric which calculates the absolute difference between actual and predicted values.

- **Mean Squared Error(MSE)**

MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value.

- **Root Mean Squared Error(RMSE)**

As RMSE is clear by the name itself, that it is a simple square root of mean squared error.

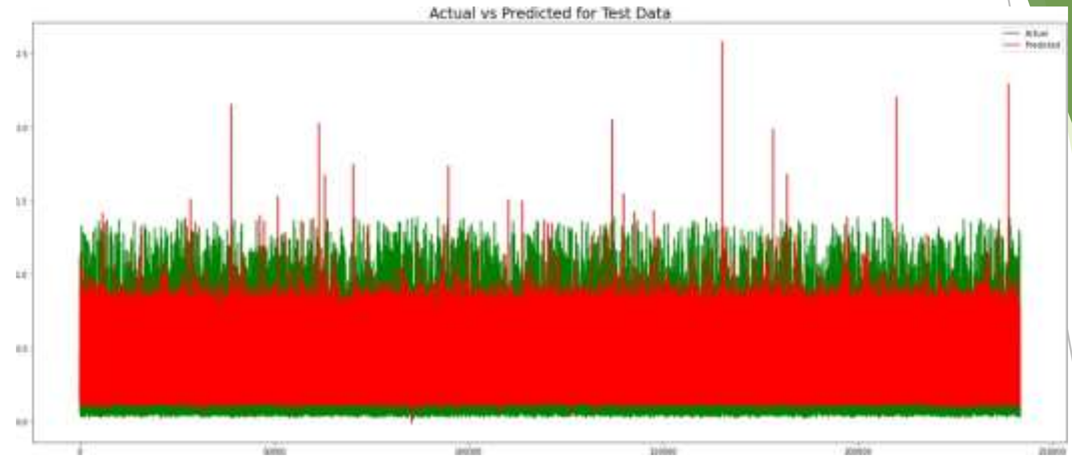
- **R Squared (R2)**

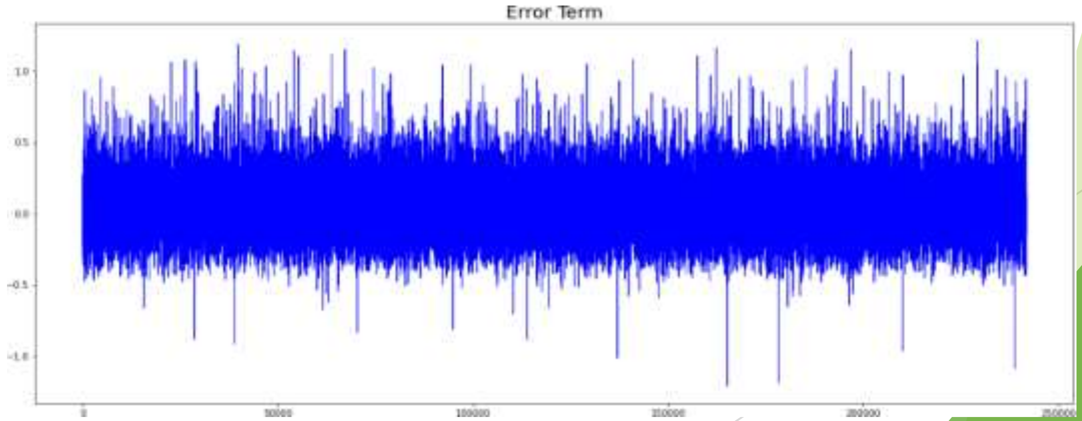
R-Squared is a statistical measure of fit that indicates how much variation of a dependent variable is explained by the independent variable(s) in a regression model. It's values range from 0 to 1and are commonly stated as percentages from 0% to 100%.

- **Adjusted R2**

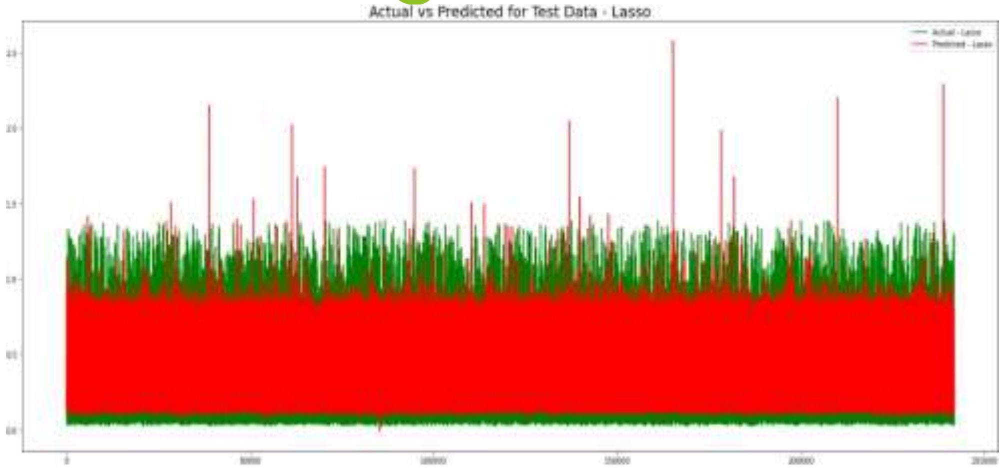
Adjusted R2 is a corrected goodness-of-fit (model accuracy) measure for linear models. It identifies the percentage of variance in the target field that is explained by the input or inputs.

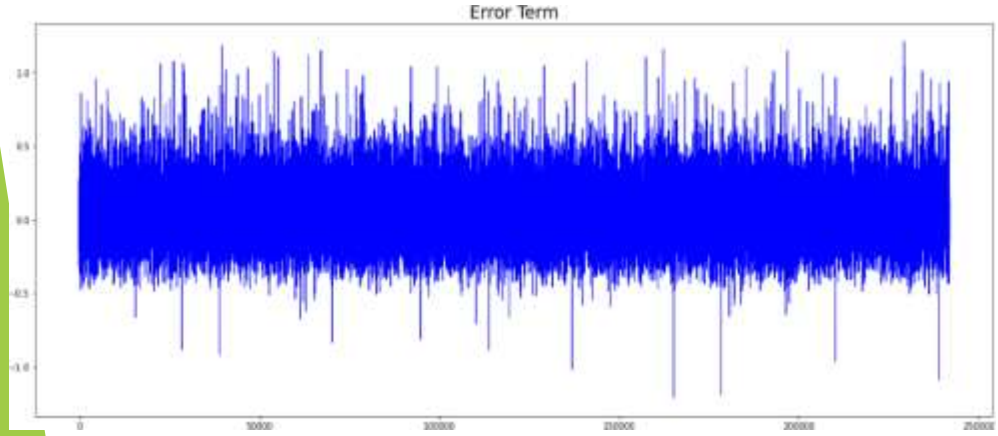
Linear Regression





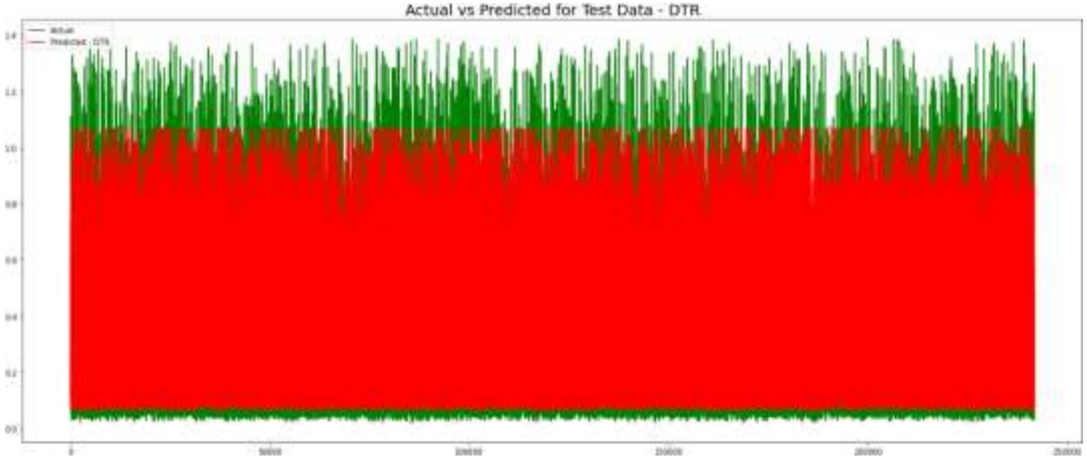
Lasso Regression

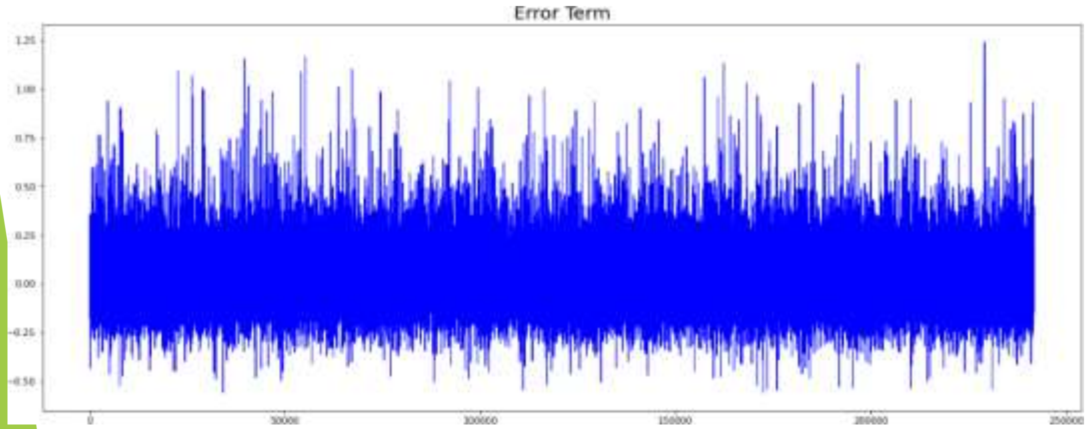




Cross Validation used

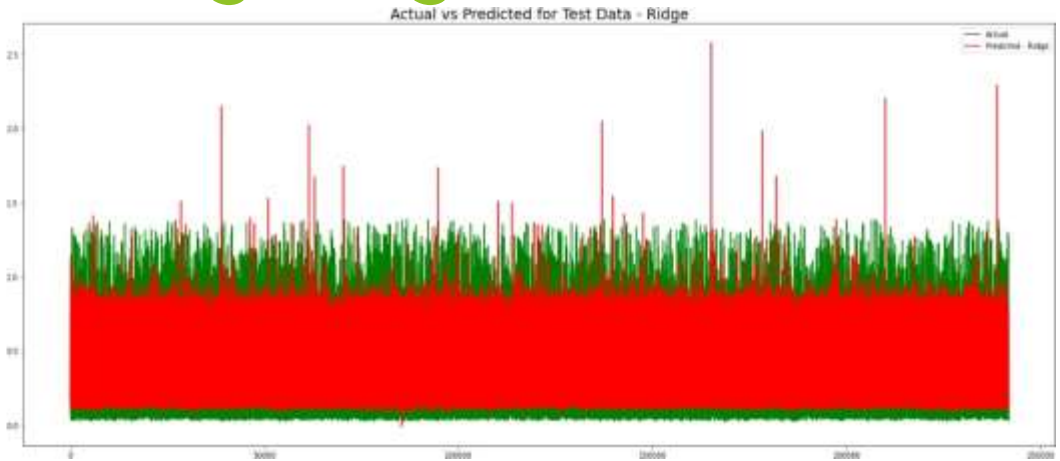
Decision Tree Regressor

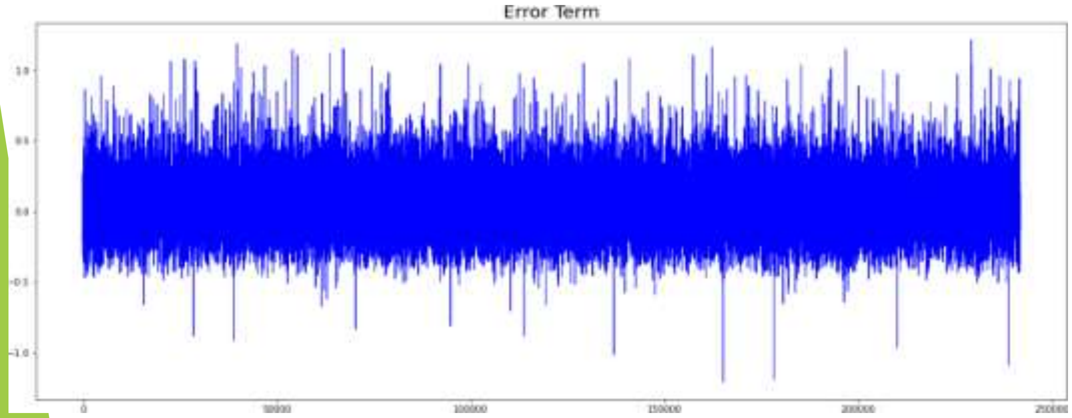




- **GridSearch CV**
- **Hyperparameter tuning**

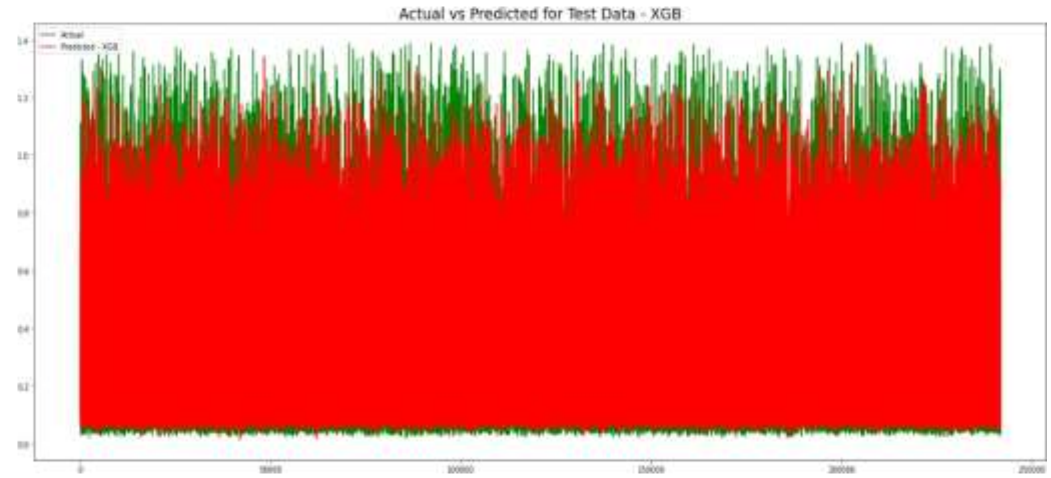
Ridge Regression

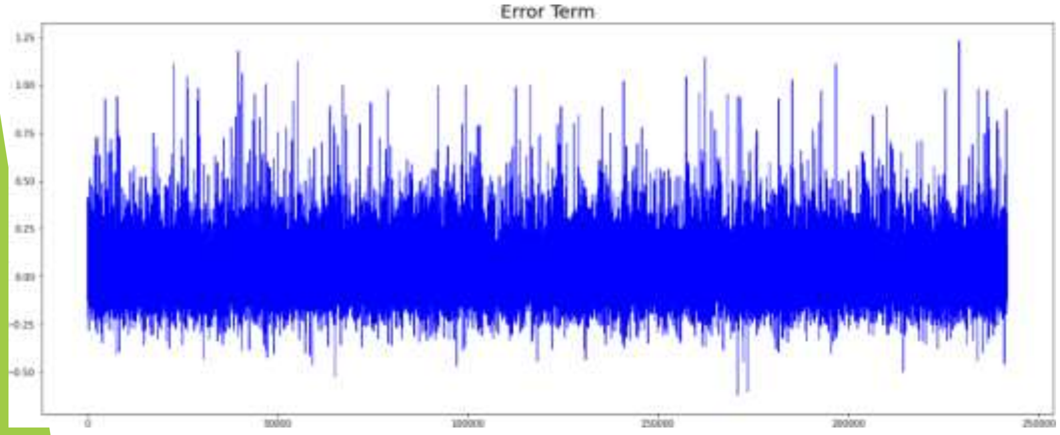




- **GridSearch CV**

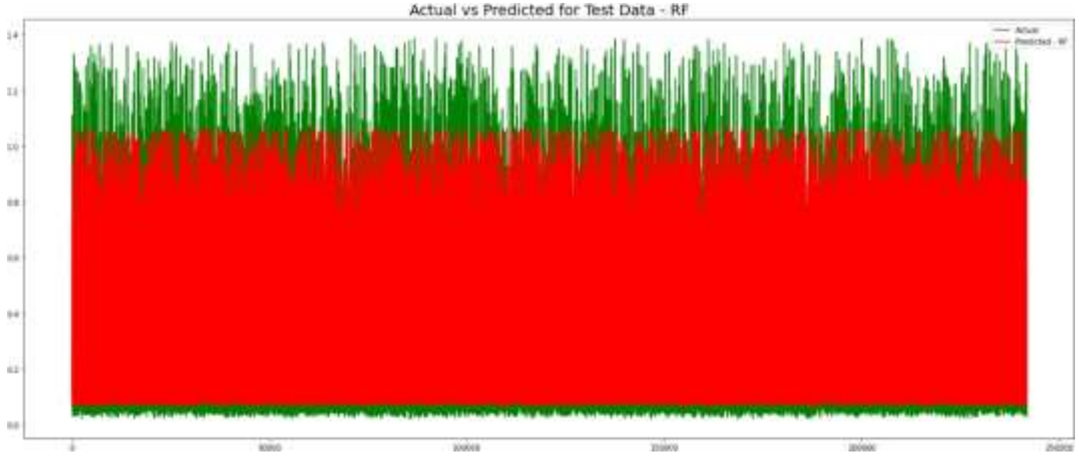
XGBoost Regressor

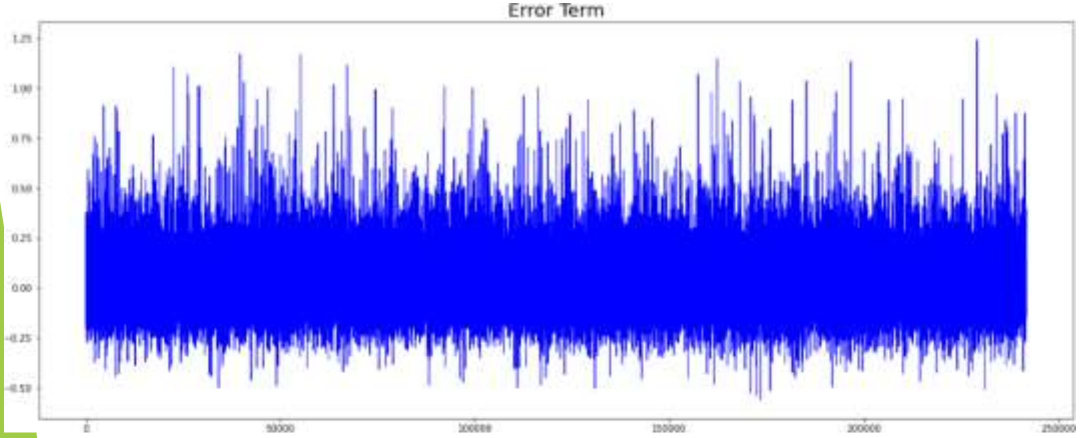




- GridSearch CV
- Hyperparameter Tuning

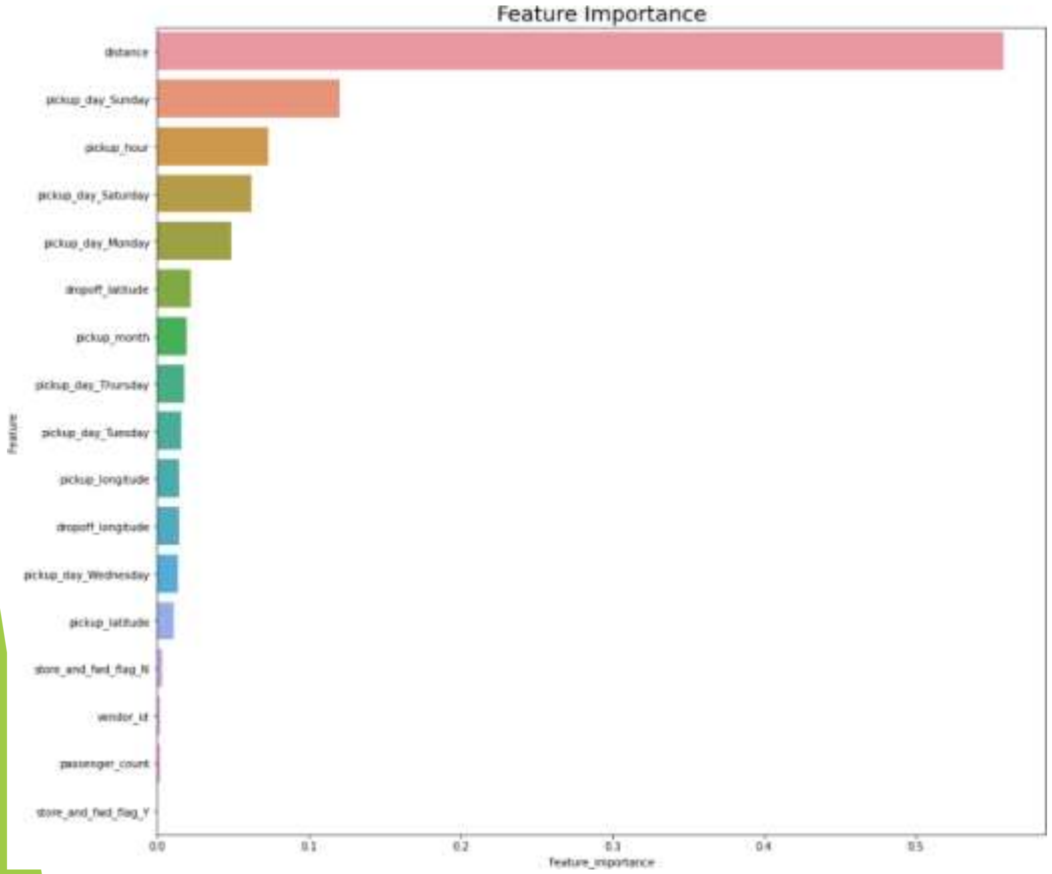
Random Forest Regressor





- GridSearch CV
- Hyperparameter Tuning

Feature Importance



- **Here, we can see among all the features, distance feature performed best in XGBoost model.**
- **And stands out to be the most important feature during prediction.**

Model Comparison

	Model Name	Train MSE	Train RMSE	Train R^2	Train Adjusted R^2
0	Linear Regression	0.012791	0.113097	0.595977	0.595970
1	Lasso Regression	0.012791	0.113097	0.595977	0.595970
2	Ridge Regression	0.012791	0.113097	0.595977	0.595970
3	DecisionTree Regressor	0.008673	0.093132	0.726034	0.726029
4	XGBoost Regressor	0.005282	0.072681	0.833144	0.833142
5	Random Forest Regressor	0.008334	0.091293	0.736744	0.736739

Train set analysis

	Model Name	Test MSE	Test RMSE	Test R^2	Test Adjusted R^2
0	Linear Regression	0.012752	0.112926	0.595680	0.595651
1	Lasso Regression	0.012752	0.112926	0.595681	0.595653
2	Ridge Regression	0.012752	0.112926	0.595681	0.595653
3	DecisionTree Regressor	0.008806	0.093841	0.720792	0.720773
4	XGBoost Regressor	0.006079	0.077967	0.807264	0.807250
5	Random Forest Regressor	0.008477	0.092073	0.731215	0.731196

Test set analysis

Challenges Faced

- **Handling the very large dataset**
- **Feature Engineering**
- **Computational time during running of model**
- **Optimization of models**

Conclusion

- Firstly, their is not much difference between the train and test values of Linear Regression, Lasso Regression, Ridge Regression during the MSE, RMSE.
- But the model performance of the above mentioned regression models
- suddenly increases for R^2 and Adjusted R^2 value.
- Hyperparameter tuning also did not help much in improving the value.
- The model performance for the Decision Tree Regressor, XGBoost Regressor
- and Random Forest Regressor finally showed some good performance in both
- train and test data for MSE, RMSE.
- Among all the used models for ML regression analysis, XGBoost Regressor showed the best performance with 80.72% in test R^2 value and 83.31% in train R^2.

Thank You