

Capstone Project

NYC Taxi Trip Duration Prediction

- 1. Vilas Sonawane**
- 2. Bhavika Gaurkar**

Point for Discussion

- **About NYC TAXI**
- **Project Road Map**
- **Business Problem**
- **Purpose of the Project**
- **Data Pipeline**
- **Data Summary**
- **Data Cleaning**
- **Feature Engineering**
- **Exploratory Data Analysis**
 - i) **Univariate Analysis**
 - ii) **Bivariate Analysis**
- **Feature Selection**
- **Preparing Dataset for Modelling**
- **Machine Learning Models**
- **Model Validation and Selection**
- **Conclusion**

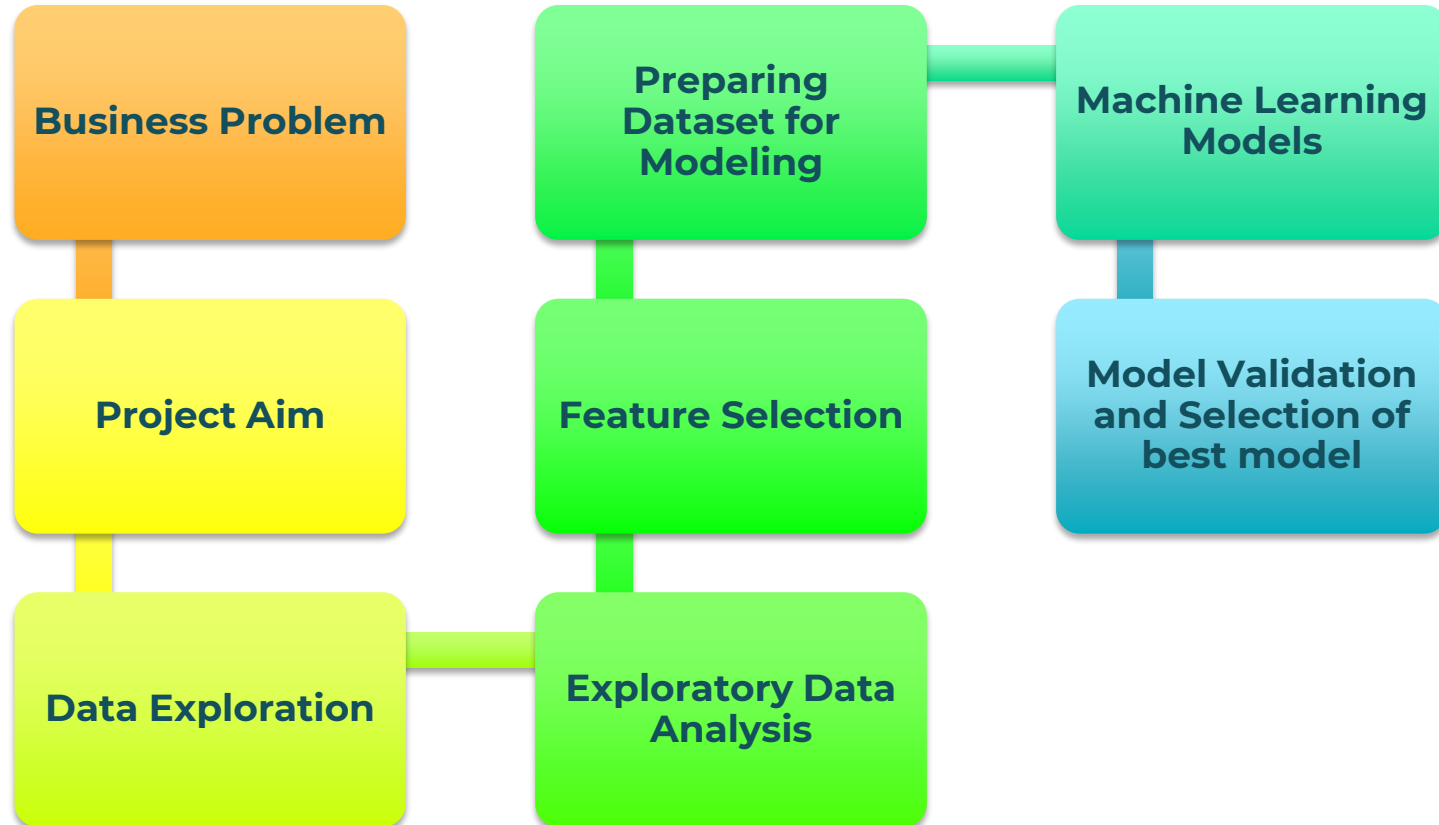
The New York City Taxi and Limousine Commission

The New York City Taxi and Limousine Commission (TLC), created in 1971, is the agency responsible for licensing and regulating New York City's Medallion (Yellow) taxi cabs, for-hire vehicles (community-based liveries, black cars and luxury limousines), commuter vans, and paratransit vehicles.

Over 200,000 TLC licensees complete approximately 1,000,000 trips each day.



Project Road Map



Business Problems



The New York City Taxi and Limousine Commission want to predict trip duration of their rides?

Purpose of The Project



The main purpose of Our project is to build Machine Learning model which can predicts the trip duration of taxi in New York City.

- **Data Processing:** In this part we have explore dataset and identified inconsistency in dataset if any and take necessary action on it wherever it was necessary. Since there were some columns which were not directly important so we have converted them into proper format , So we have also created some new features based on exited feature in dataset.
- **EDA:** In this part we explored the data and identified outlier and inconsistent data and removed outlier and modified data whenever required and obtained some useful insights and trends from the data.
- **Data Preparation:** After cleaning data we have prepared data for implementation of regression models by removing features having high multicollinearity between each other. Then selected dependent features and normalized data and make it ready for application of machine learning model.
- **Machine Learning Regression Modelling:** After preparation of data we have applied different regression model on the dataset. Applying the model is not an easy task. It's also an iterative process. We have started with simple regression model, then slowly used complex models for better performance.

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
Y=store and forward; N=not a store and forward trip

trip_duration

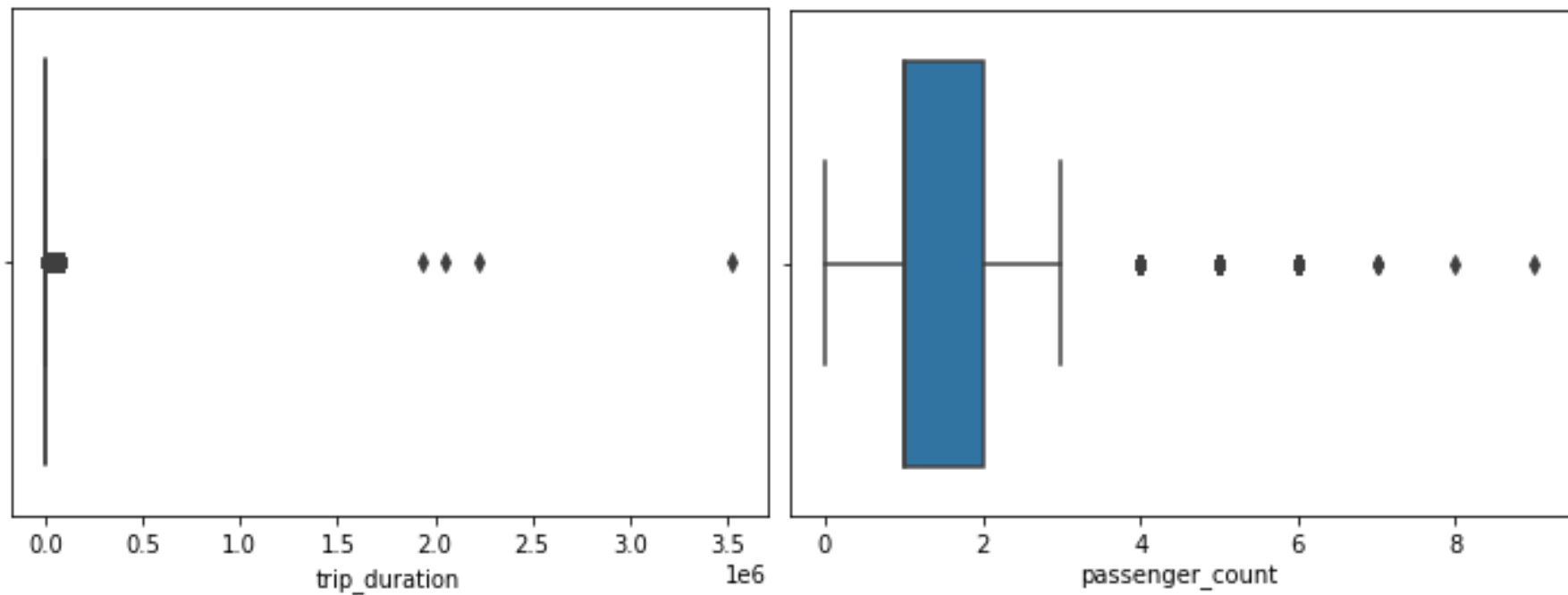
duration of the trip in seconds

Dealing with Outliers

The data also has some outliers. **Trip duration and Passenger count** attribute has an outlier and we have rearranged the trip duration with in the **range of 30sec to 3440 sec** and minimum passenger count is 1. Here in the below table, we can easily identify the presence of outliers (highlighted) for column trip duration and passenger count

	vendor_id	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	trip_duration
count	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06
mean	1.534950e+00	1.664530e+00	-7.397349e+01	4.075092e+01	-7.397342e+01	4.075180e+01	9.594923e+02
std	4.987772e-01	1.314242e+00	7.090186e-02	3.288119e-02	7.064327e-02	3.589056e-02	5.237432e+03
min	1.000000e+00	0.000000e+00	-1.219333e+02	3.435970e+01	-1.219333e+02	3.218114e+01	1.000000e+00
25%	1.000000e+00	1.000000e+00	-7.399187e+01	4.073735e+01	-7.399133e+01	4.073588e+01	3.970000e+02
50%	2.000000e+00	1.000000e+00	-7.398174e+01	4.075410e+01	-7.397975e+01	4.075452e+01	6.620000e+02
75%	2.000000e+00	2.000000e+00	-7.396733e+01	4.076836e+01	-7.396301e+01	4.076981e+01	1.075000e+03
max	2.000000e+00	9.000000e+00	-6.133553e+01	5.188108e+01	-6.133553e+01	4.392103e+01	3.526282e+06

Dealing with Outliers



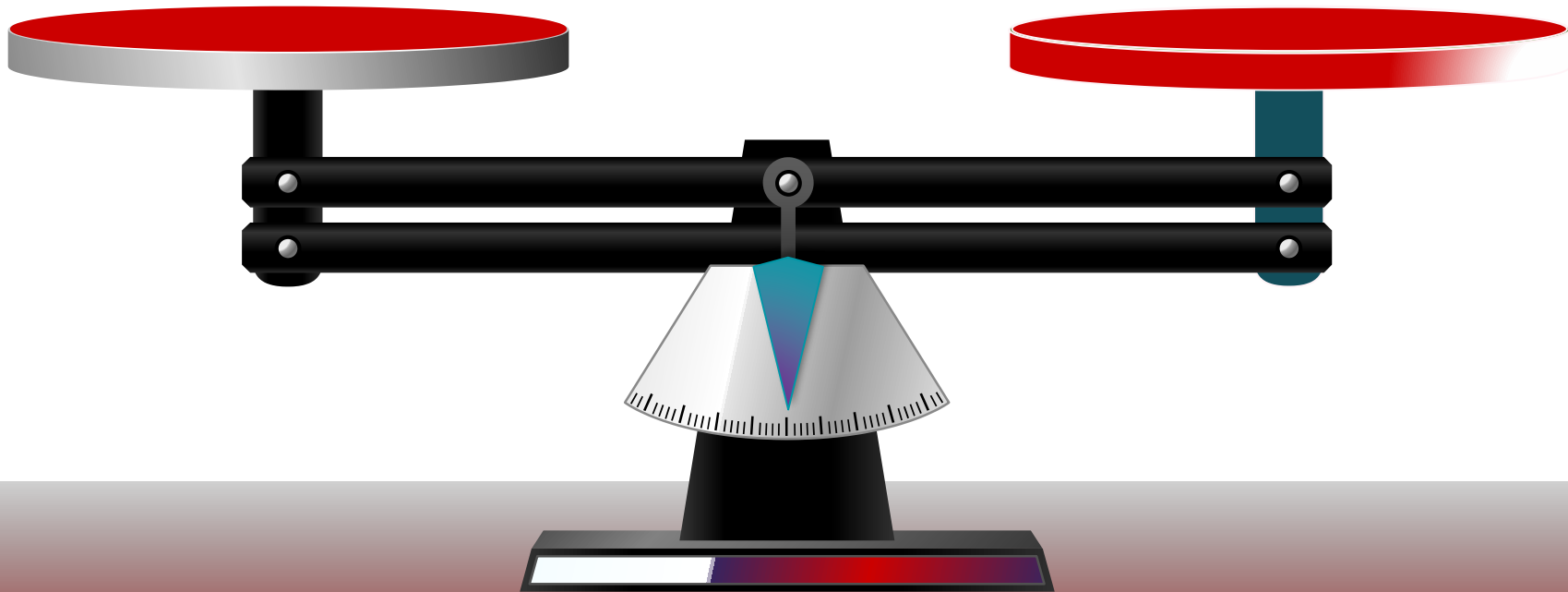
Feature Engineering

We have created the following features :

- **Pickup day & Dropoff day** : Contains the name of the weekdays.
- **Pickup day no & Dropoff day no**: Convert weekday into number start at Monday = 0 and end at Sunday = 6.
- **Pickup hour & Dropoff hour** : Extracting hour.
- **Pickup month & Dropoff month** : Extract month in number
- **Distance** : Calculate the distance with the help of latitude and longitude (km).
- **Speed** : Calculated speed distance/trip duration (km/hr).
- **Day part** : Dividing 24 hours into four parts
 - **Morning** (from 6:00 am to 11:59 pm)
 - **Afternoon** (from 12 noon to 3:59 pm)
 - **Evening** (from 4:00 pm to 9:59 pm)
 - **Late Night** (from 10:00 pm to 5:59 am)

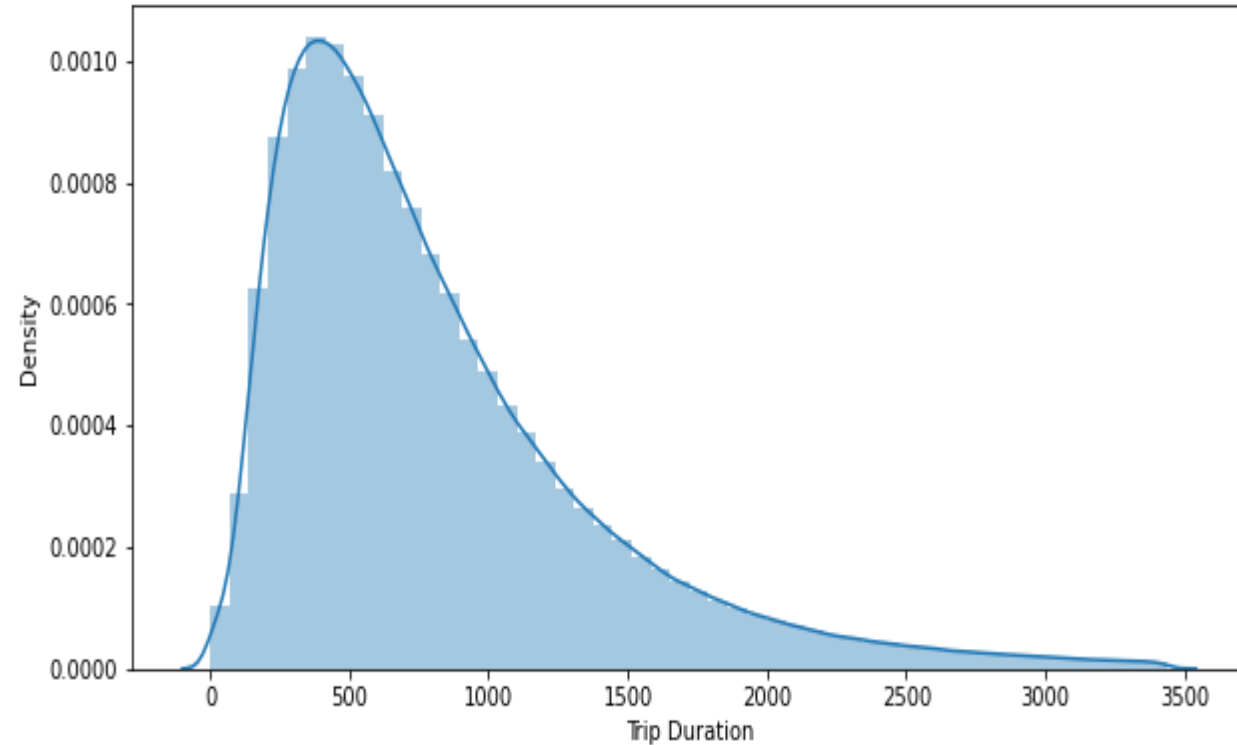
Univariate Analysis

Bivariate Analysis



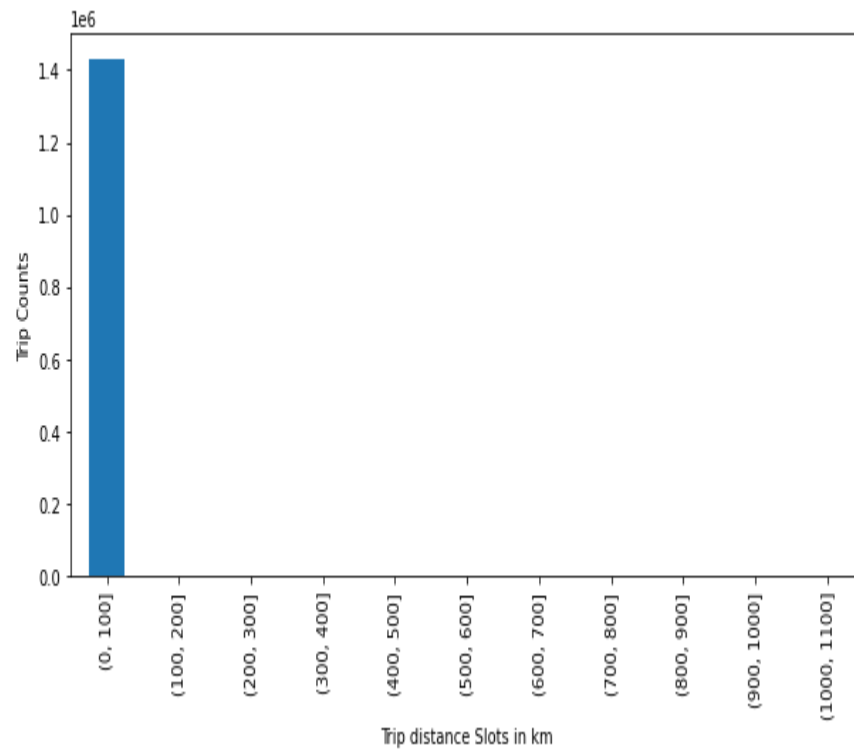
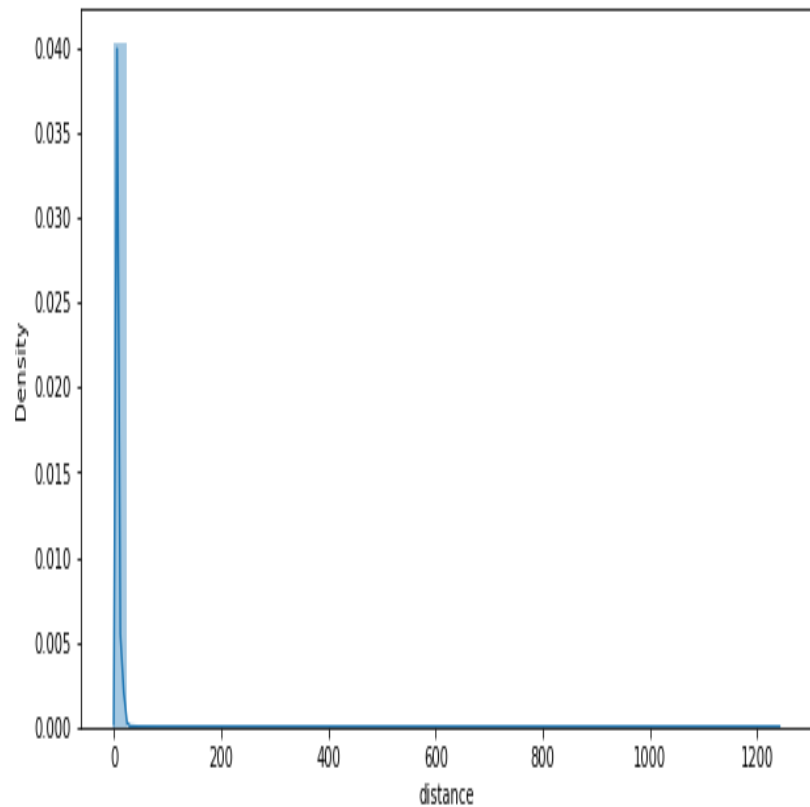
EDA

Trip Duration

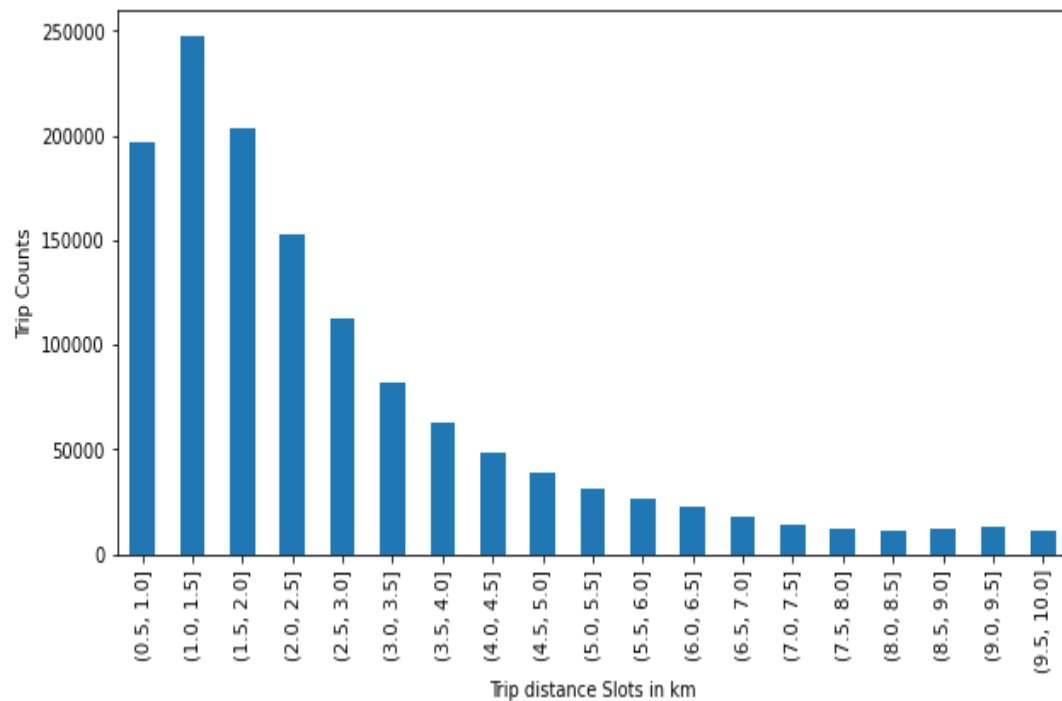
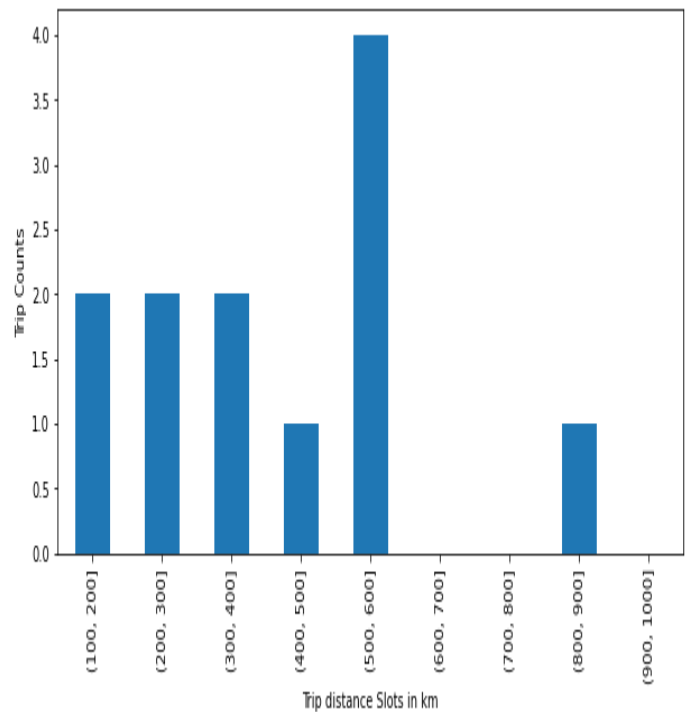


- We can see that major chunk (99th percentile) of trip duration is completed in 3440 seconds i.e. nearly 1 hour.
- There are very few trip which have duration greater than 3500 seconds.
- There are some durations with as low as 1 second. which points towards trips with 0 km distance.
- We should get rid the outliers for the sake of data consistency. (Trip duration greater than 3440 seconds and also trip duration less than 30 seconds)

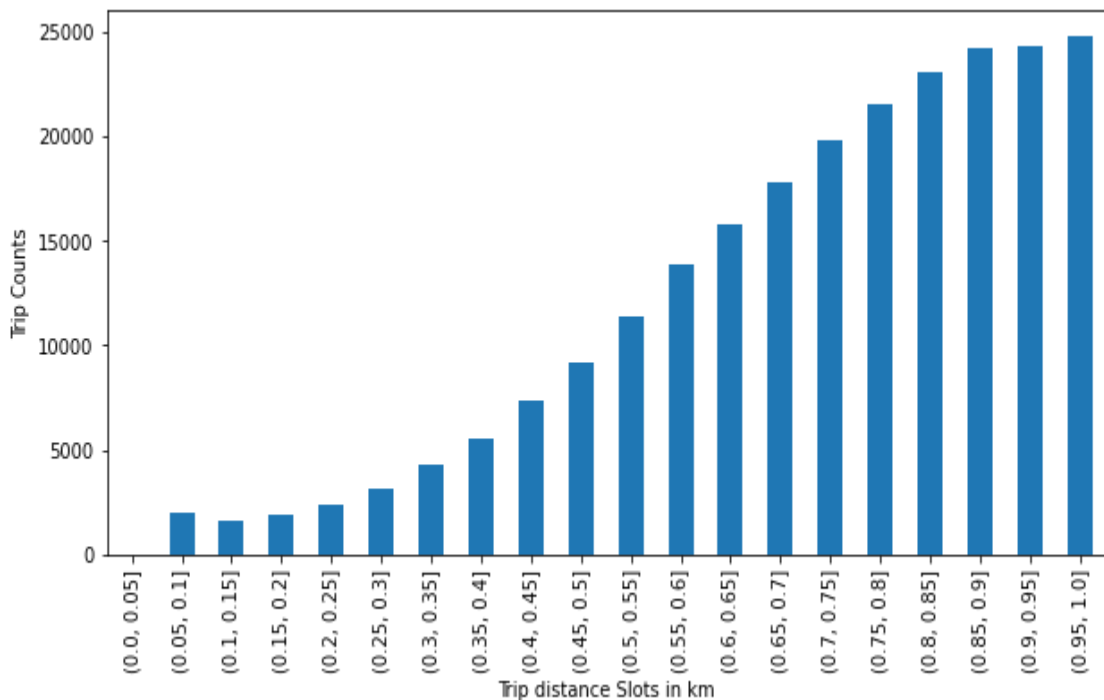
Distance



Distance

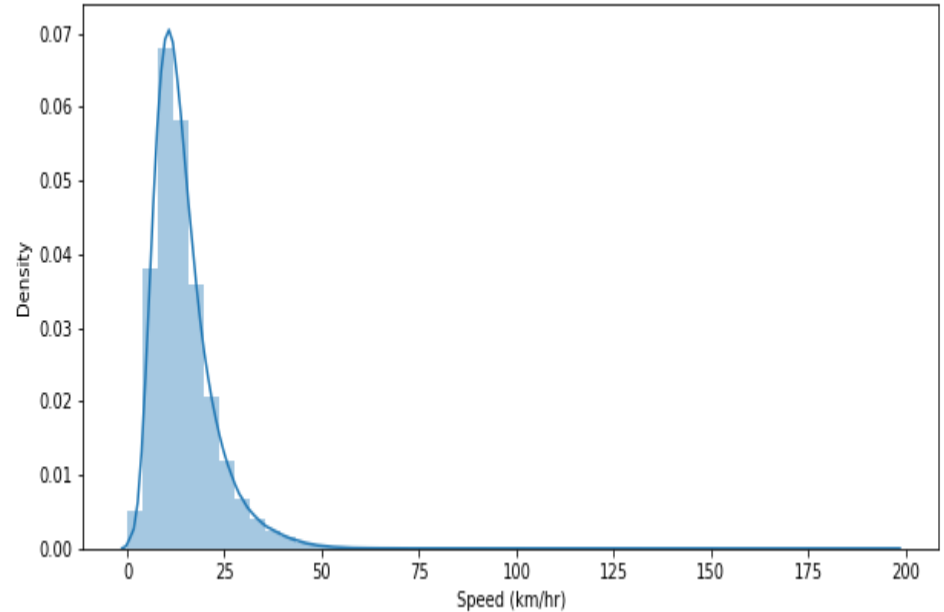
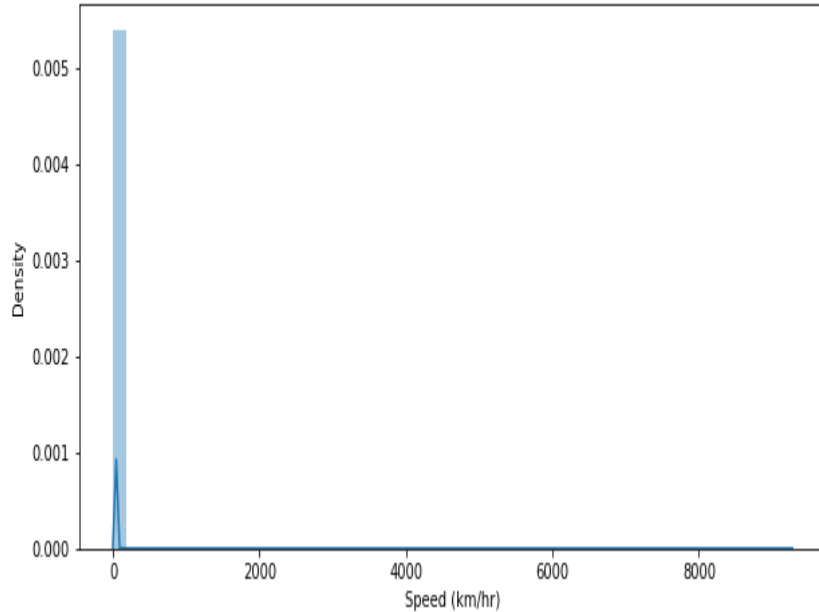


Distance



- There are **16 trips with more than 100 km** distance.
- There are **5887 trips with 0 km** distance. so this trip will unnecessary influence our model so we have dropped those rows.
- **The possible reasons for 0 km trips can be:**
 - The drop-off location couldn't be tracked.
 - The driver deliberately took this ride to complete a target ride number.
 - The passengers or driver cancelled the trip due to some issue.
 - Due to some technical issue in software, etc.

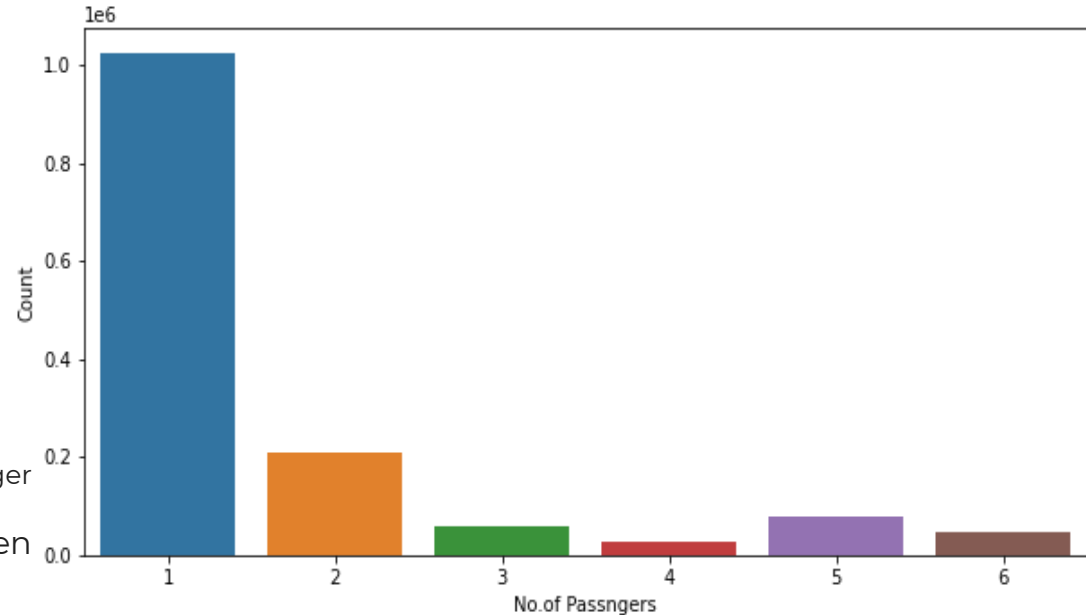
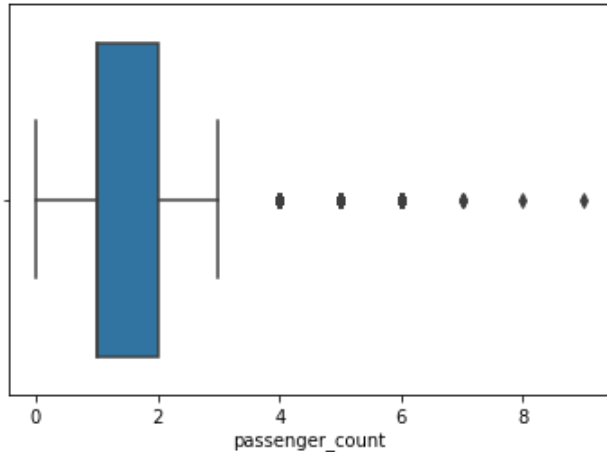
Speed



- There are some trips that were done at a **very high speed 8000 km/hr** + which is quite impossible. So we have applied some speed limit for dataset.
- As per the rule in NYC, the speed limit is 25 mph(approx. 40km/h) in New York City.
- **Mostly trips are done at a speed range of 5-25 km/hr.**

Passenger count

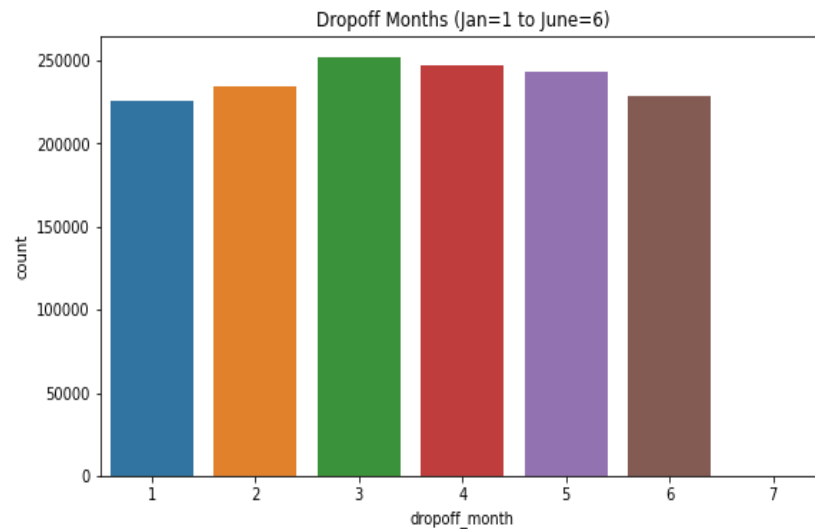
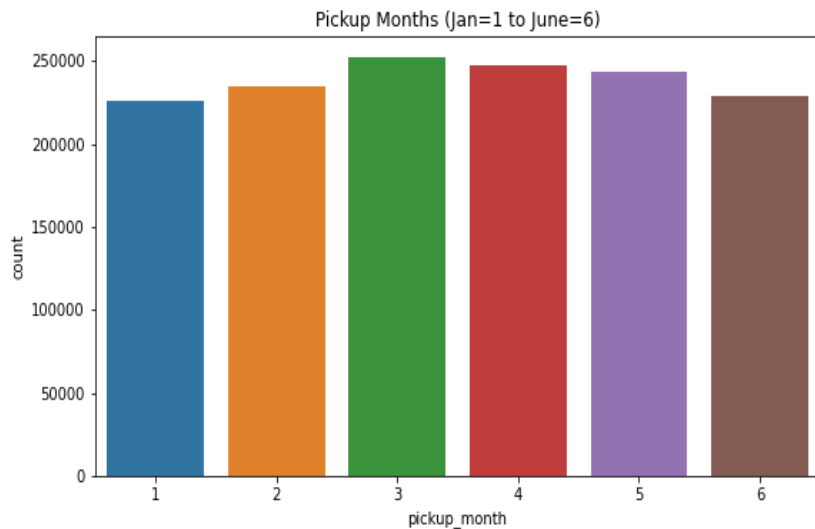
No of passenger	1	2	5	3	6	4	0	7	9	8
Trip counts	1024120	207672	77220	59209	47775	28009	59	3	1	1



Let us remove the rows which have 0,7, 8 and 9 passenger count which are very less.

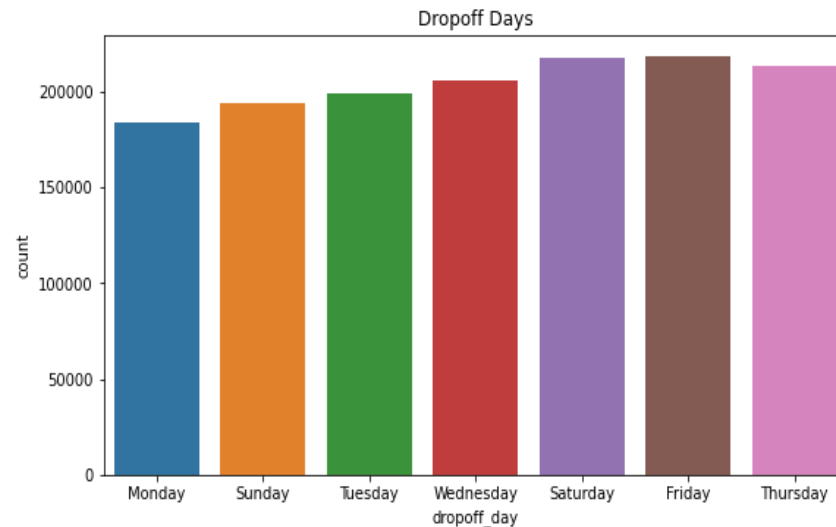
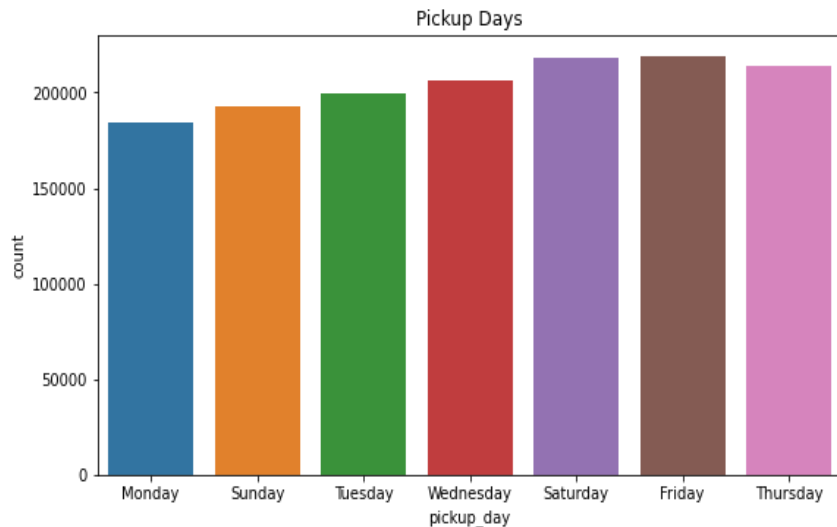
- We see the highest amount of trips was taken by a single passenger.
- The instance of large group of people travelling together is rare.

Pickup month & Drop up month



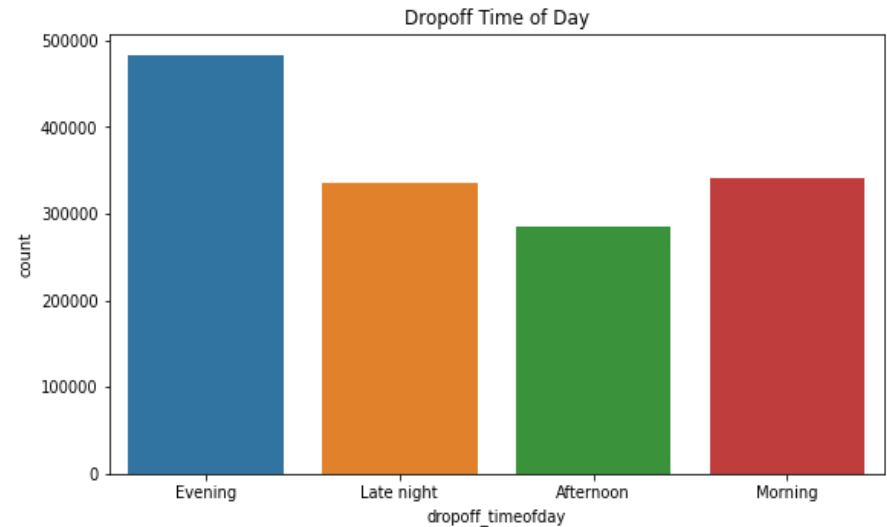
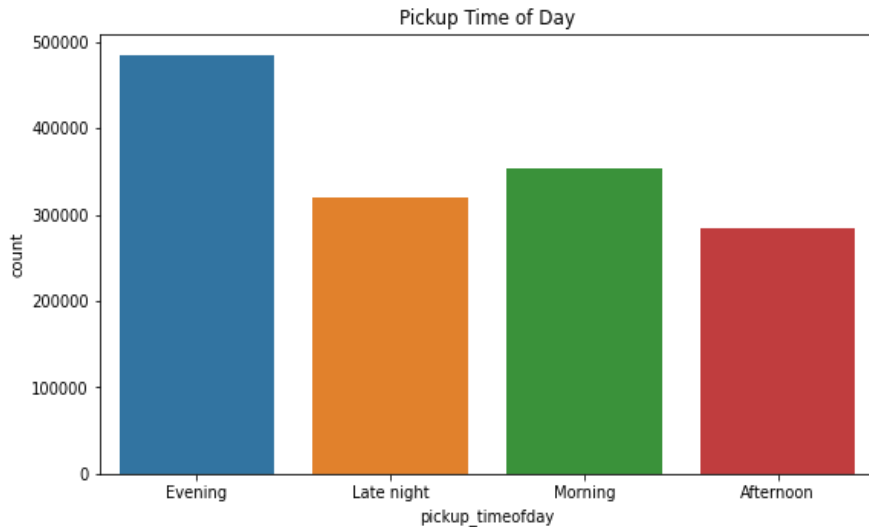
There is not much difference in the number of trips across months.

Pickup day & Drop up day



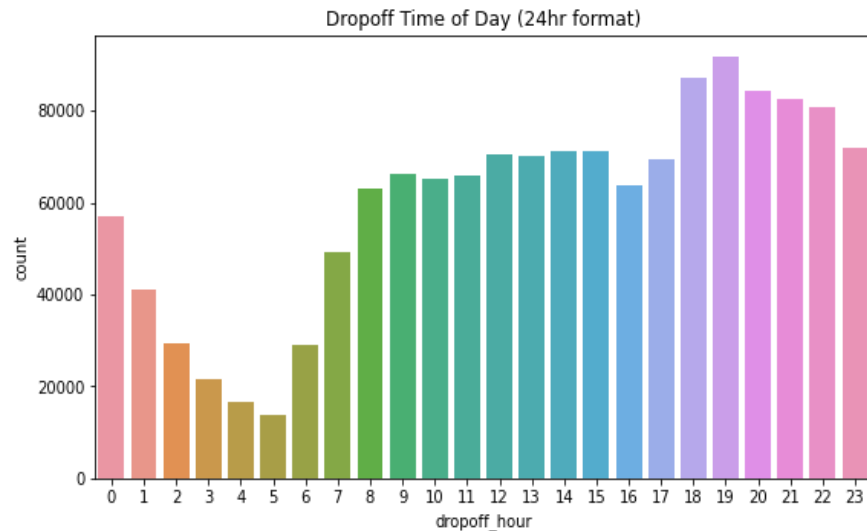
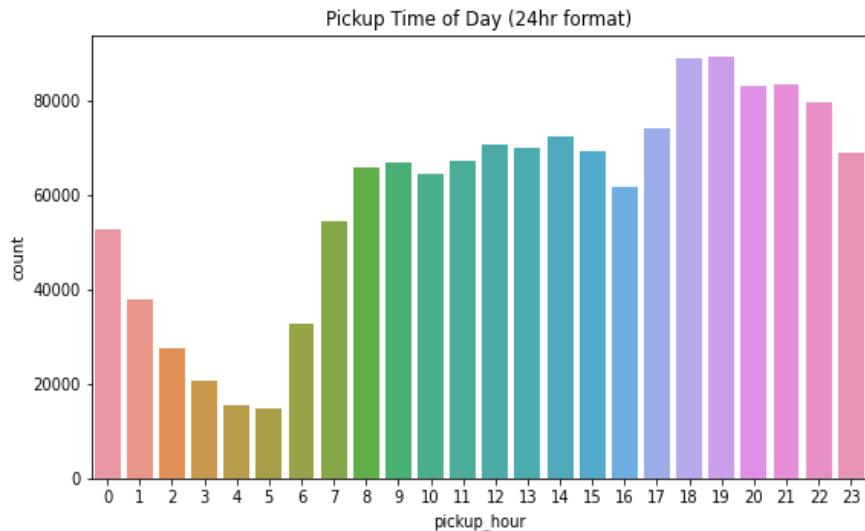
We see Fridays are the busiest days followed by Saturdays.
That is probably because it's weekend.

Pickup time of day & Dropoff time of day



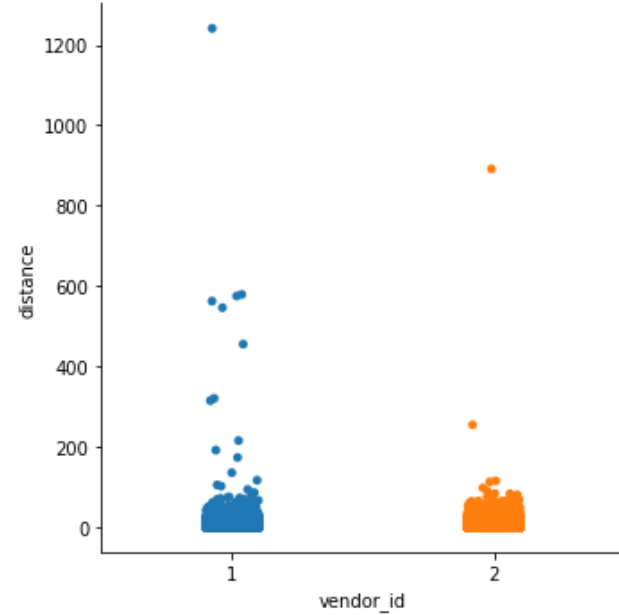
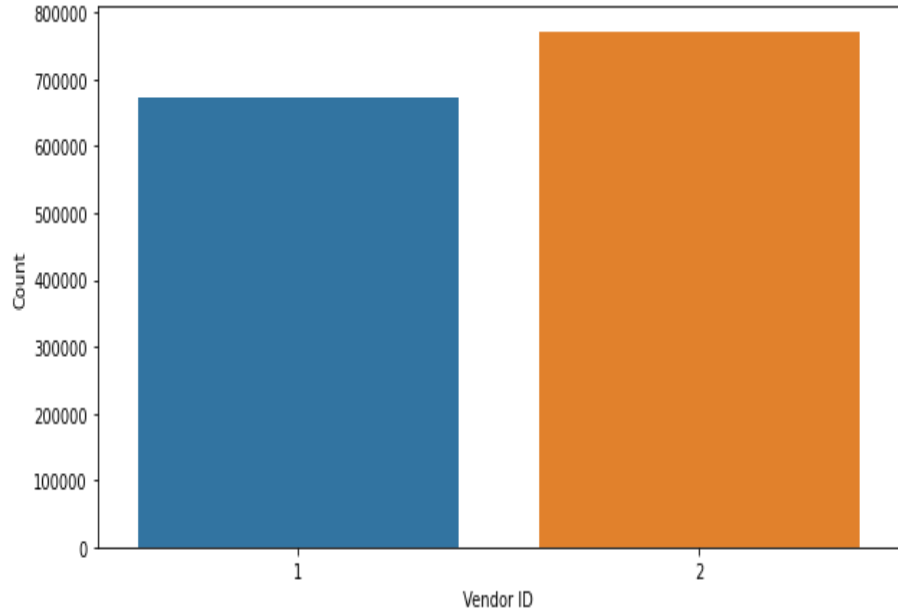
As we saw above, evenings are the busiest.

Pickup hour & Drop up hour



We see the busiest hours are 6:00 pm to 7:00 pm which makes sense as this is the time for people to return home from work.

Vendor id



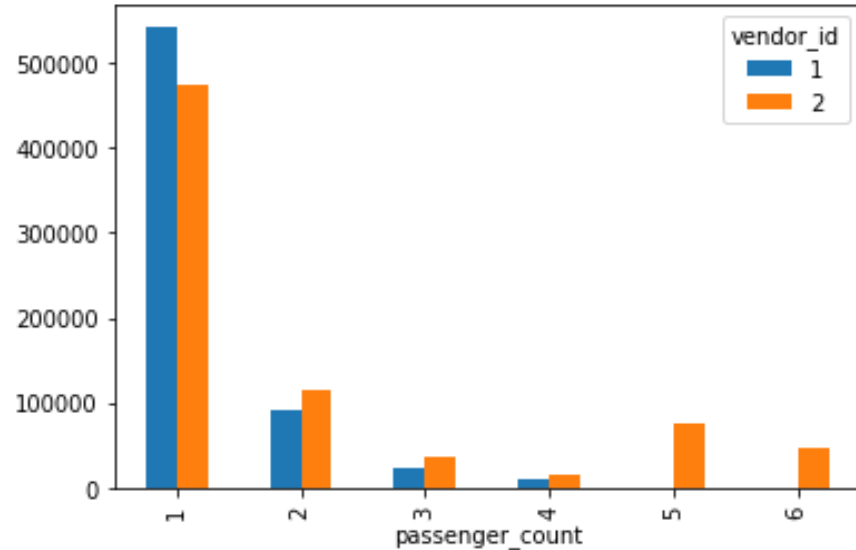
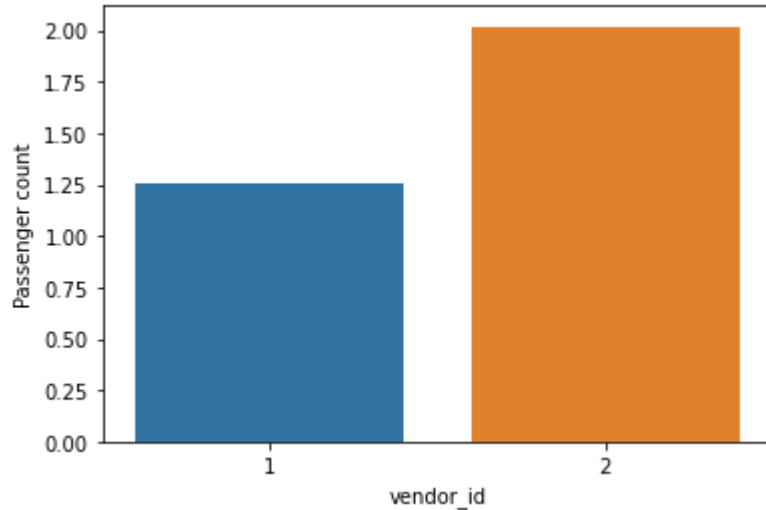
So vender 2 is more famous as compare to vender 1 & Vender 1 is taking more long trips as compare to vender 1

EDA

Bivariate Analysis

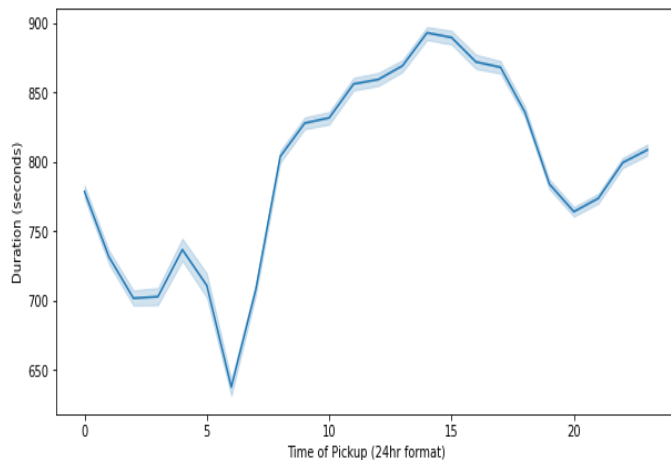


Passenger Count Per Vendor

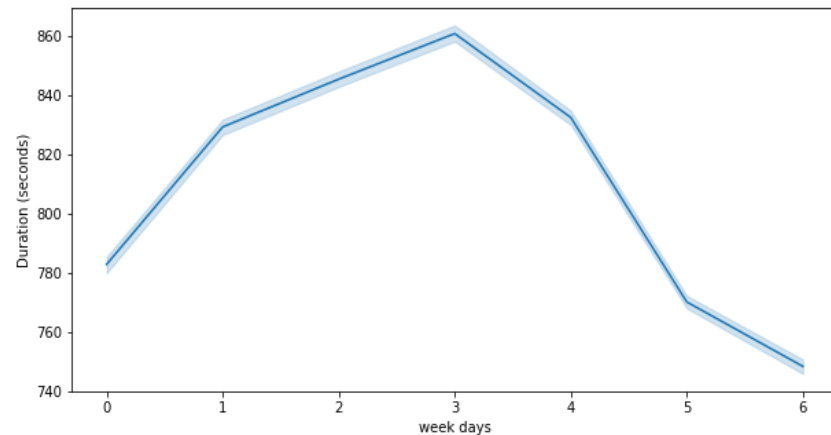


- It seems that vendor 2 trips generally consist of 2 passengers as compared to the vendor 1 with 1 passenger.
- We can see that the big car served by the vendor 2 vendor 1 has more mini vans
- the vendor 2 has greater market share as compare to vendor 1

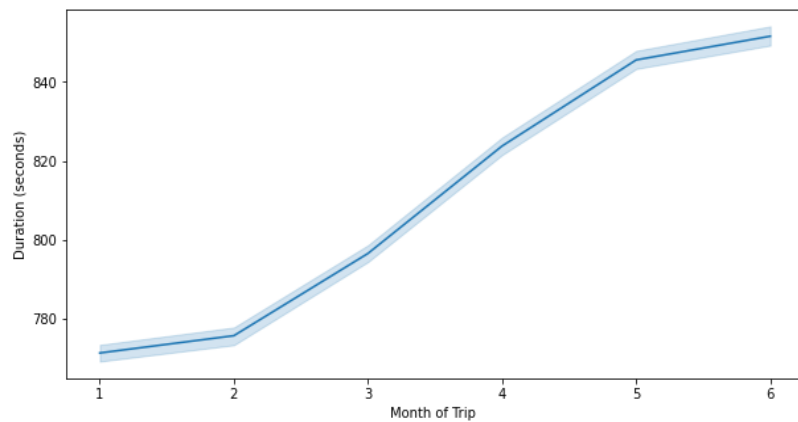
Trip Duration per hour



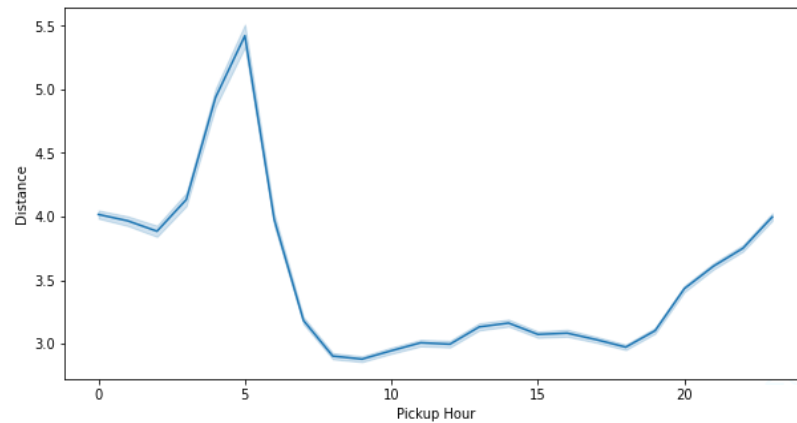
Trip duration per weekday



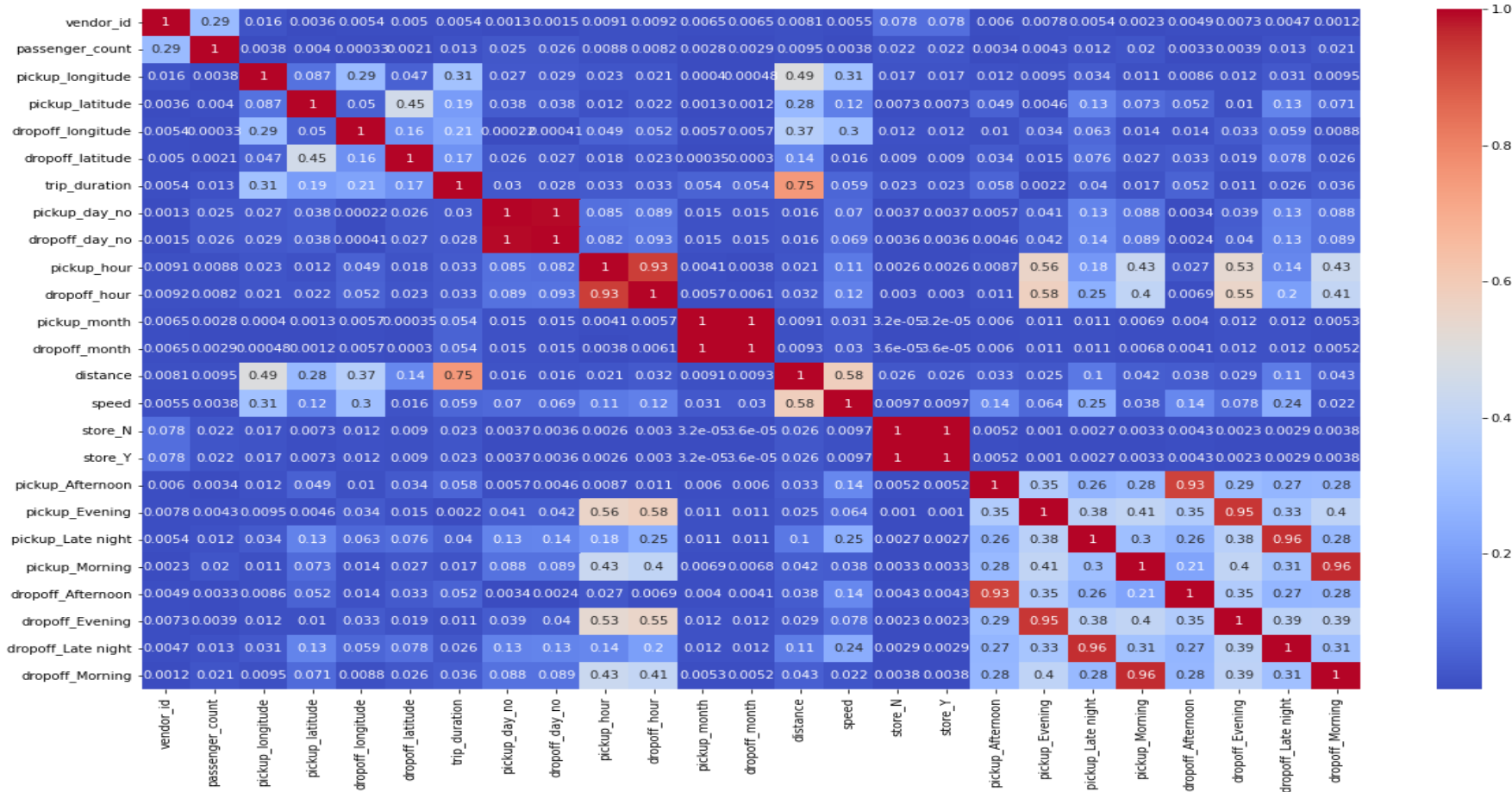
Trip duration per month



Distance and Hour

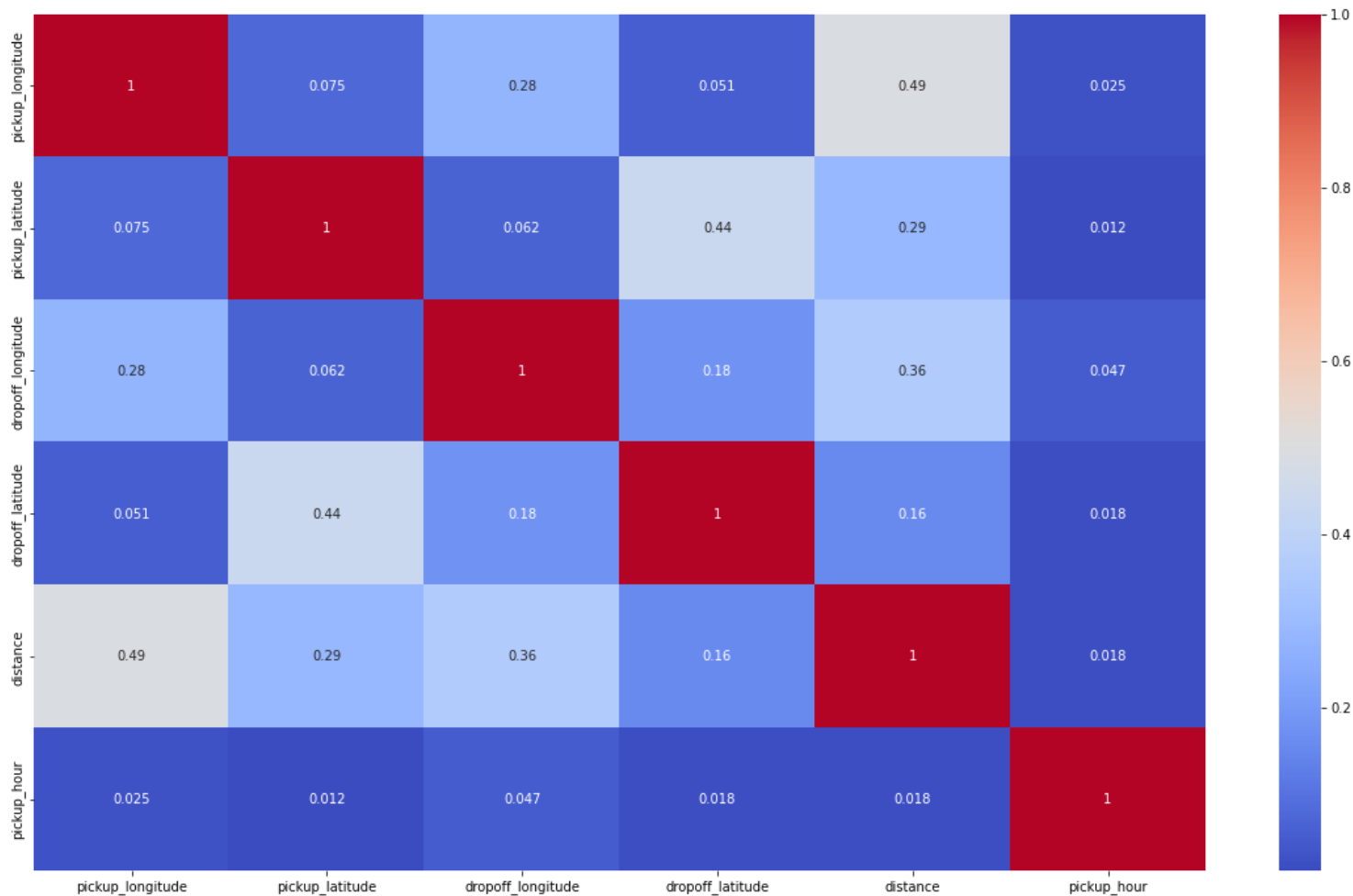


Feature Correlation



Lets Drop he features which are having high collinearity with each other (More than 0.4)

Feature Correlation Between Dependent Features



Feature Selection



Preparing Dataset for Modeling



```
[ ] independent_variables=['pickup_longitude', 'pickup_latitude', 'dropoff_longitude', 'dropoff_latitude', 'distance', 'pickup_hour']

    dependent_variables = 'trip_duration'

[ ] # Create the data of independent variables
    X = df_corr[independent_variables]

    # Create the dependent variable data
    y = df_corr[dependent_variables]

[ ] from sklearn.preprocessing import StandardScaler
    # Transforming data
    scaler = StandardScaler()
    X = scaler.fit_transform(X)
```

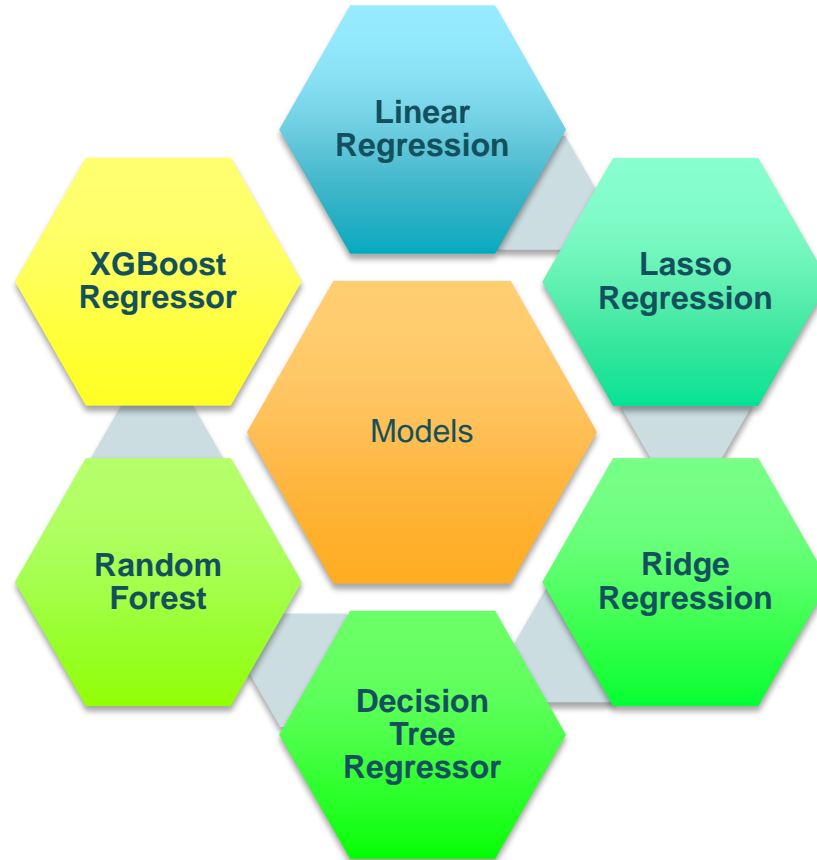
Splitting the data in train and test sets

```
[ ] #Importing RFE and Linear Regression
    from sklearn.model_selection import train_test_split

    # Splitting the dataset into the Training set and Test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.35, random_state=0)
```

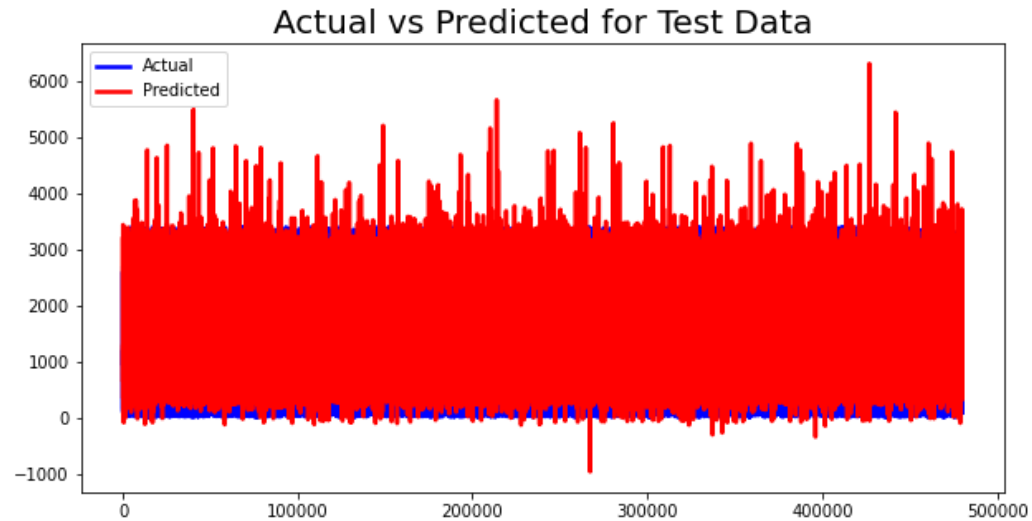
- Normalizing the Dataset using **Standard Scaling** Technique.
- It is because **Standard Scaling** adjusts the value of mean and standard deviation between **0's and 1's** , which tend to work better for optimization techniques like Gradient descent.

Machine Learning Models



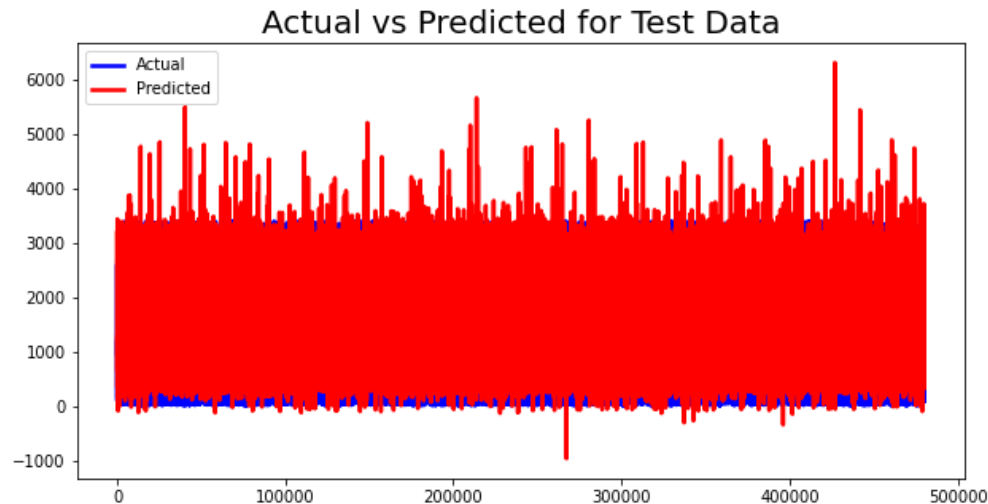
Linear Regression

Error Matrices	Train Data	Test Data
MSE	120625.236	121458.094
R squared Value	0.631030	0.626488
Adjusted R squared Value	0.631027	0.626483



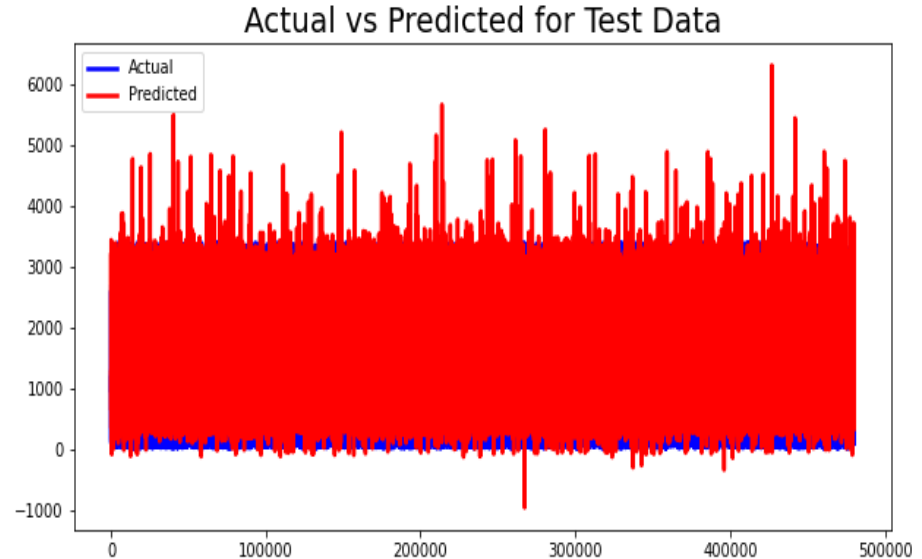
Lasso Regression

Error Matrices	Train Data	Test Data
MSE	120625.237	121458.077
R squared Value	0.631030	0.626488
Adjusted R squared Value	0.631028	0.626483



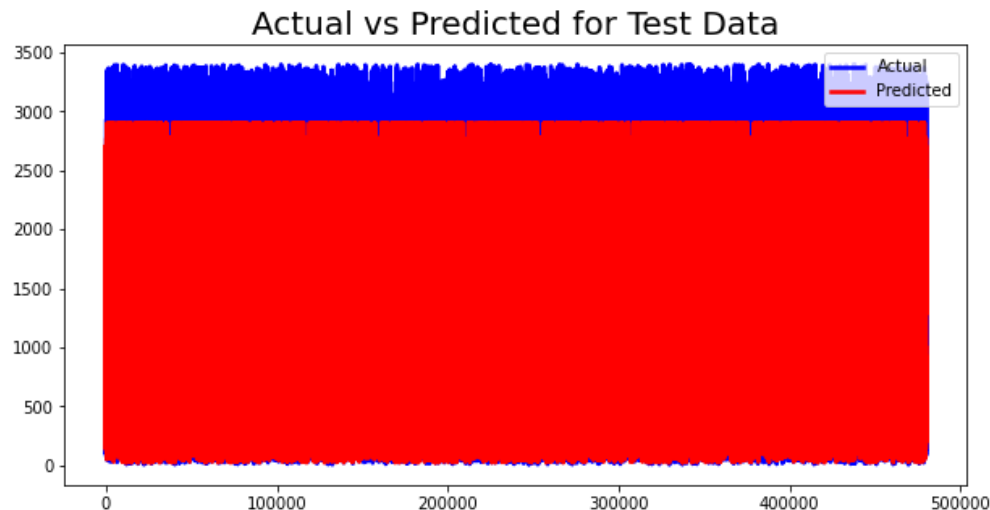
Ridge Regression

Error Matrices	Train Data	Test Data
MSE	120625.2374	121458.062
R squared	0.631030	0.626488
Adjusted R squared	0.631028	0.626483



Decision Tree Regression

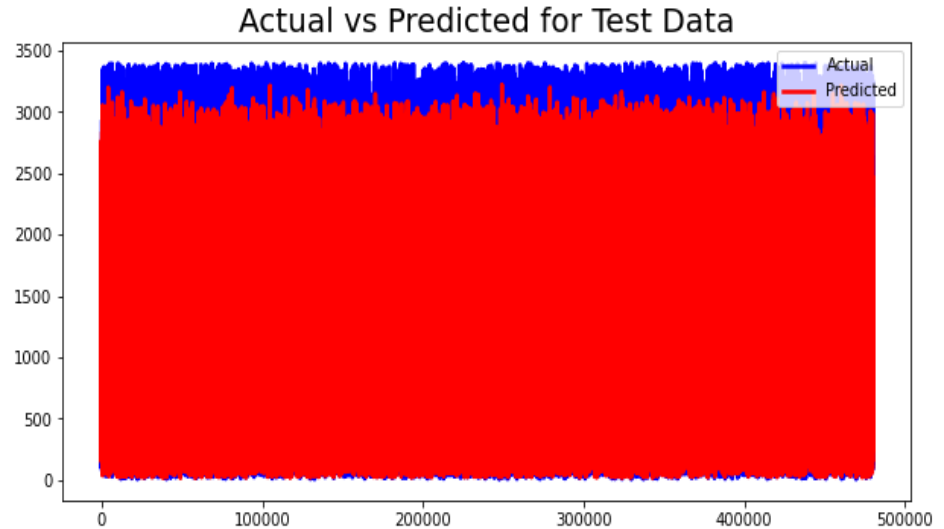
Error Matrices	Train Data	Test Data
MSE	74874.505	76851.041
R squared	0.770973	0.763665
Adjusted R squared	0.770972	0.763662



- The best fit alpha value is found out to be : {max depth: 10, min samples leaf: 22, min samples split: 30}
- The R2 score using the same alpha is : 0.766297121693419

Random Forest Regression

Error Matrices	Train Data	Test Data
MSE	72197.2638 39	74063.6063 95
R squared	0.779162	0.772237
Adjusted R squared	0.779161	0.772234

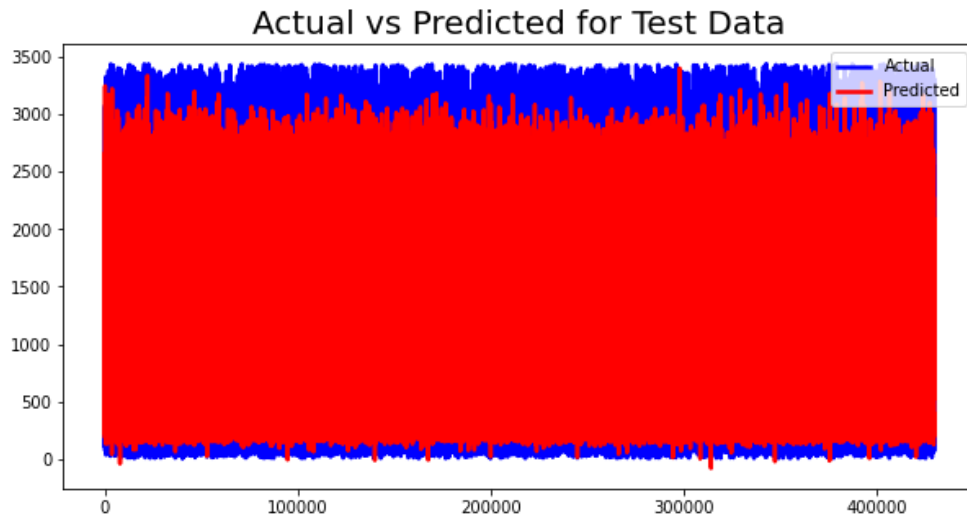


The best fit alpha value is found out to be :

(n_estimators = 100, max_depth = 10, min_samples_split = 5 , n_jobs = -1)

XGBooster Regression

Error Matrices	Train Data	Test Data
MSE	52591.28905	59863.998
R squared	0.839133	0.815904
Adjusted R squared	0.839132	0.815902



The best fit alpha value is found out to be : {learning rate: 0.2, max depth: 8, min samples: 4, n estimators: 200, scoring = r2, verbose =1)

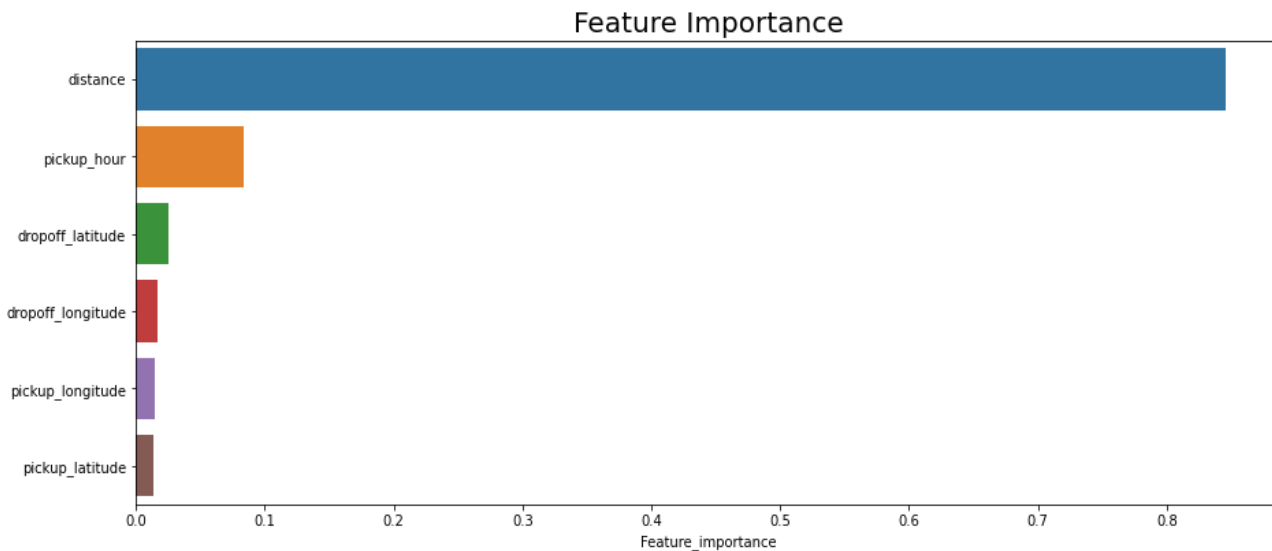
The R2 score using the same alpha is : 0.8151701094551514

Model Validation and Selection

	Model Name	Train MSE	Test MSE	Train R ²	Test R ²	Train Adjusted R ²	Test Adjusted R ²
0	Linear Regression	120625.236922	121458.094909	0.631030	0.626488	0.631028	0.626483
1	Lasso Regression	120625.237961	121458.077369	0.631030	0.626488	0.631028	0.626483
2	Ridge Regression	120625.237419	121458.062872	0.631030	0.626488	0.631028	0.626483
3	DecisionTree Regressor	74874.505055	76851.041027	0.770973	0.763665	0.770972	0.763662
4	Random Forest	72197.263839	74063.606395	0.779162	0.772237	0.779161	0.772234
5	XGBoost Regressor	52591.289051	59863.998326	0.839133	0.815904	0.839132	0.815902

Feature Importance

	Features	Feature_importance
0	pickup_longitude	0.014439
1	pickup_latitude	0.013807
2	dropoff_longitude	0.016493
3	dropoff_latitude	0.025443
4	distance	0.846257
5	pickup_hour	0.083561



Conclusion

We can see that MSE and R2 and Adjusted R2 which are the metrics used to evaluate the performance of regression model of Linear Regression, Lasso, Ridge, Decision Tree, Random Forest and XGBooster Regressor.

The Linear models don't show good performance on our training and testing environment. There is a significant improvement in the RMSE score for the tuned XGBooster regressed over the Random Forest regression when trained on the feature selection group.

The variance score of the XGB Regression more efficient than other Regression Models.

XGBooster Regression model proved to be much more efficient in predicting the output but it takes much more time to train it over the large dataset with more complexity as compared to the other Regression models.

Based on results we have obtained we can conclude that XGBooster Regressor is the best models as compare to the other models to predict the trip duration for a NYC Taxi.



Thank You