Capstone Project
NYC Taxi Trip Duration

Individual
Yaman Saini

Point for Discussion

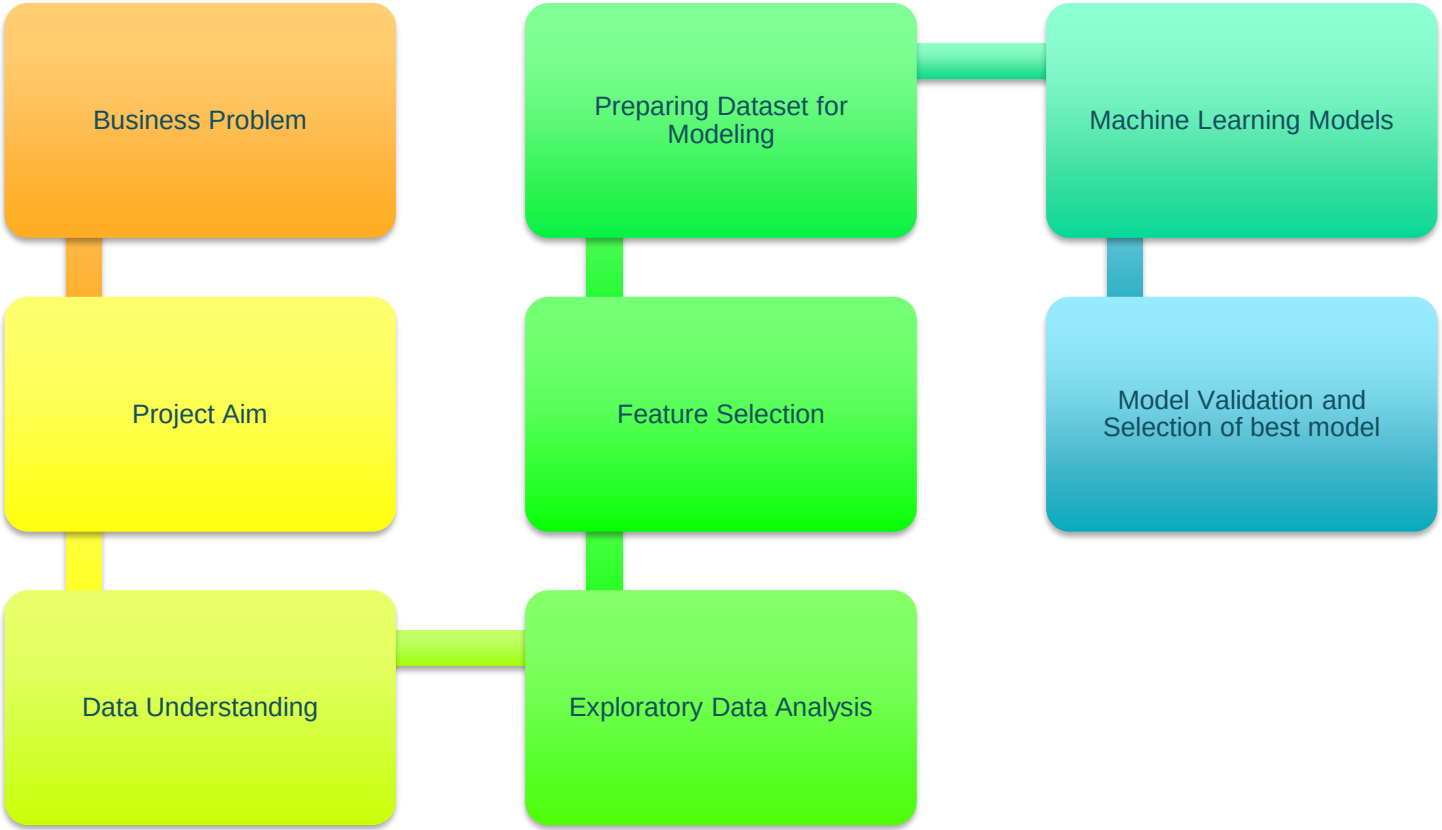
- About TLC
- Project Road Map
- Business Problem
- Purpose of the Project
- Data Pipeline
- Data Summary
- Dealing with outliers
- Feature Engineering
- Exploratory Data Analysis
    - i) Univariate Analysis
    - ii) Bivariate Analysis
- Feature Selection
- Preparing Dataset for Modelling
- Machine 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 the project is to build model which predicts the trip duration of taxi in New York City.

Data Pipeline

- **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

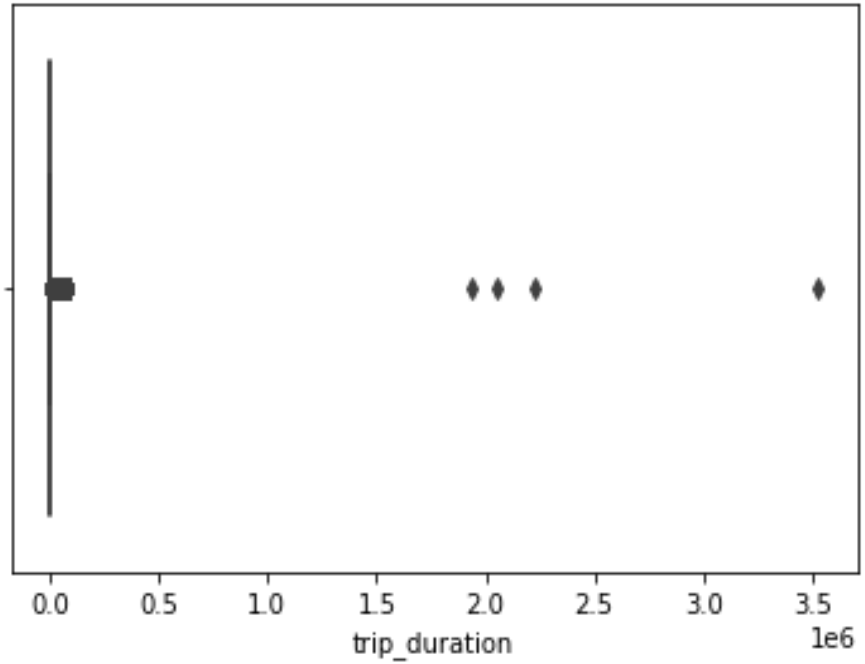
Id	a unique identifier for each trip
vendor_id	a code indicating the provider associated with the trip record
pickup_datetime	date and time when the meter was engaged
dropoff_datetime	date and time when the meter was disengaged
passenger_count	the number of passengers in the vehicle (driver entered value)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

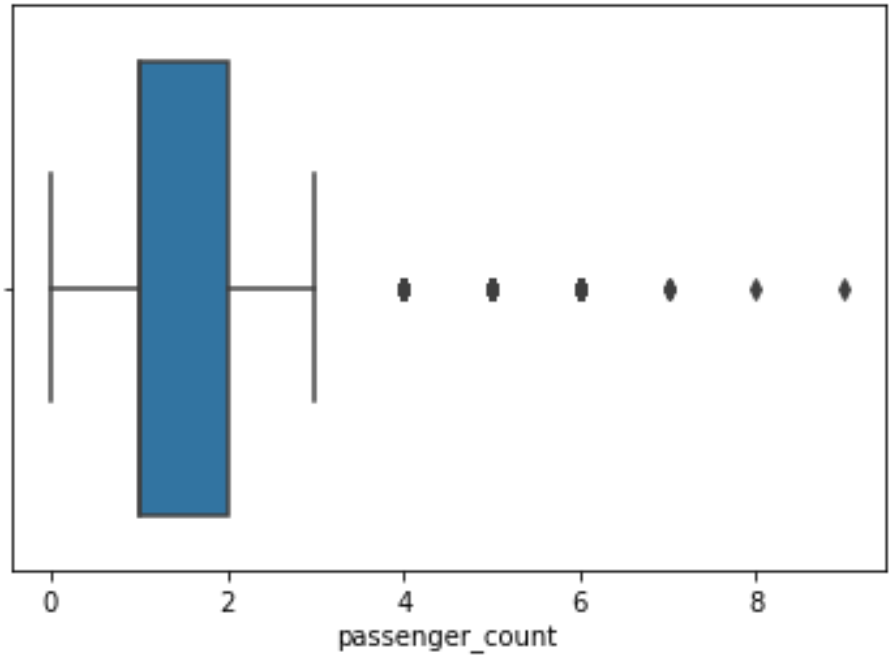
Dealing with Outliers

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





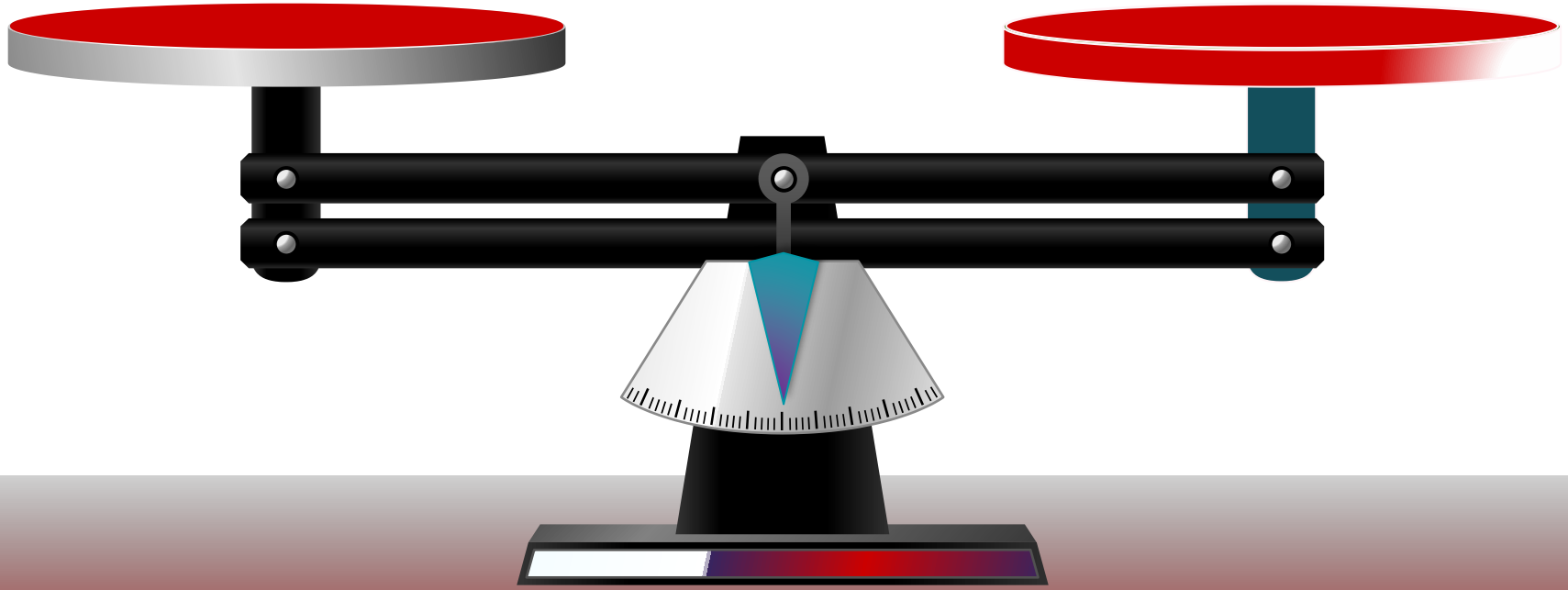
Feature Engineering

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 Nigh** (from 10:00 pm to 5:59 am)

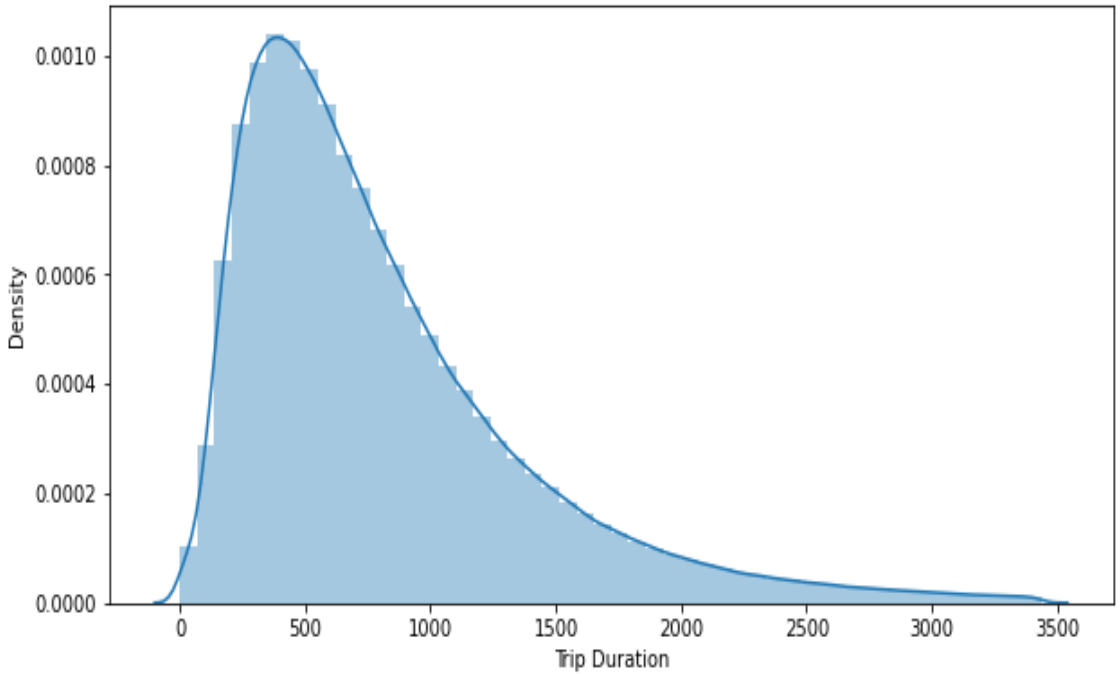
Univariate Analysis

Bivariate Analysis



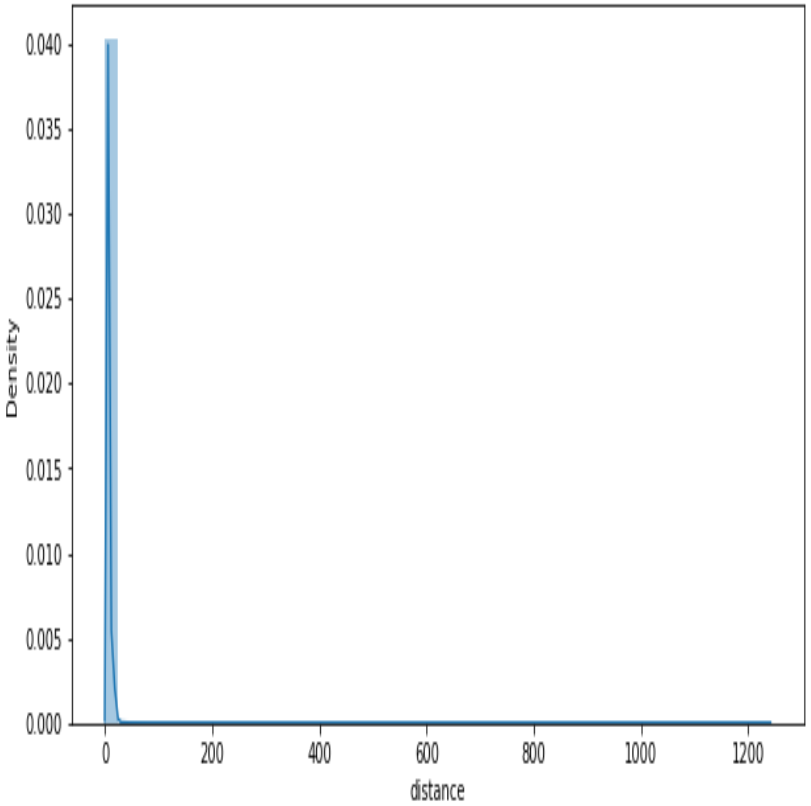
EDA

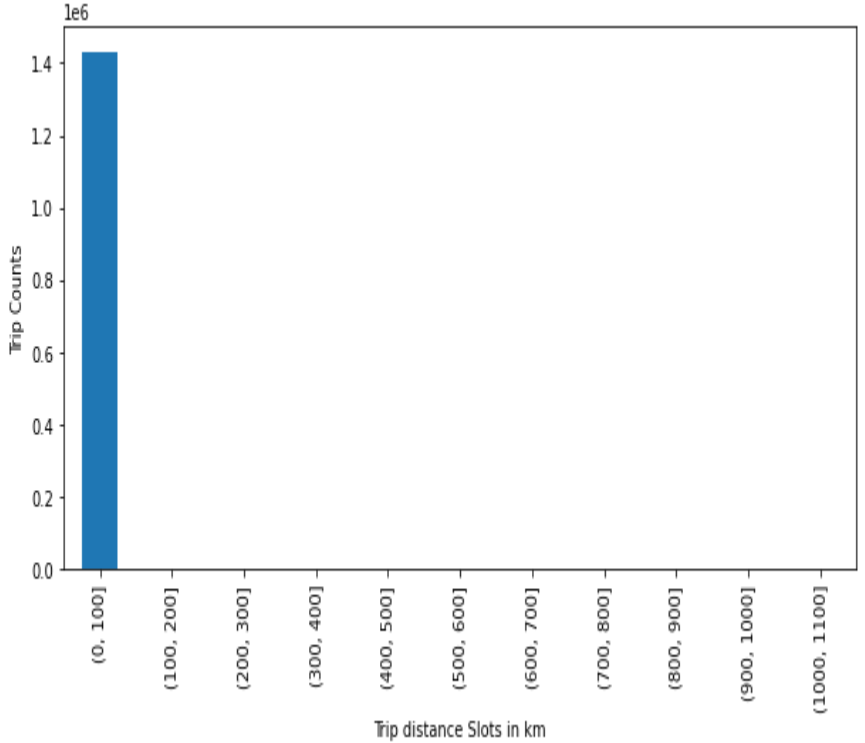
Trip Duration.



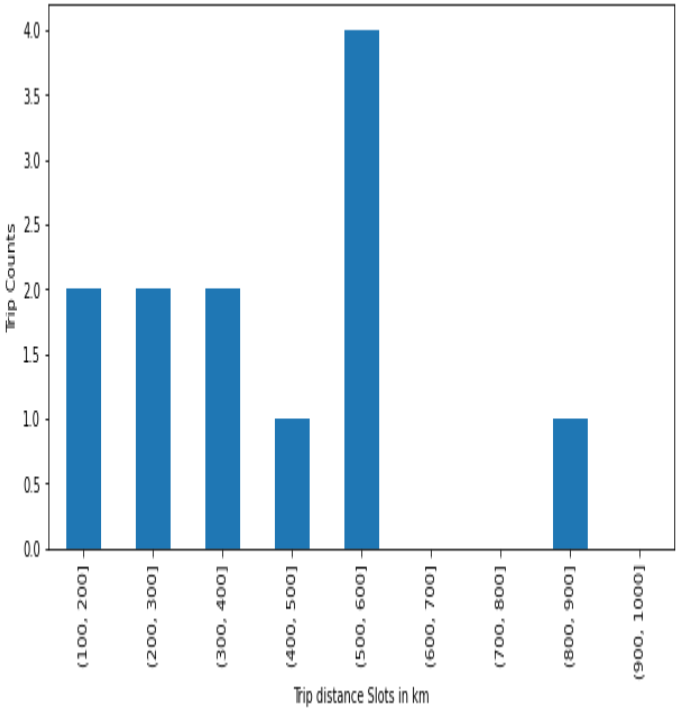
- We can see that major chunk (99th percentile) of trip duration is completed in 3440 seconds i.e. nearly 1 hour.
- There are very few trip which have duration greater than 5000 seconds.
- There are some durations with as low as 1 second. which points towards trips with 0 km distance.
- We should get rid the outliers for the sake of data consistency. (Trip duration greater than 5000 seconds and also trip duration less than 30 seconds)

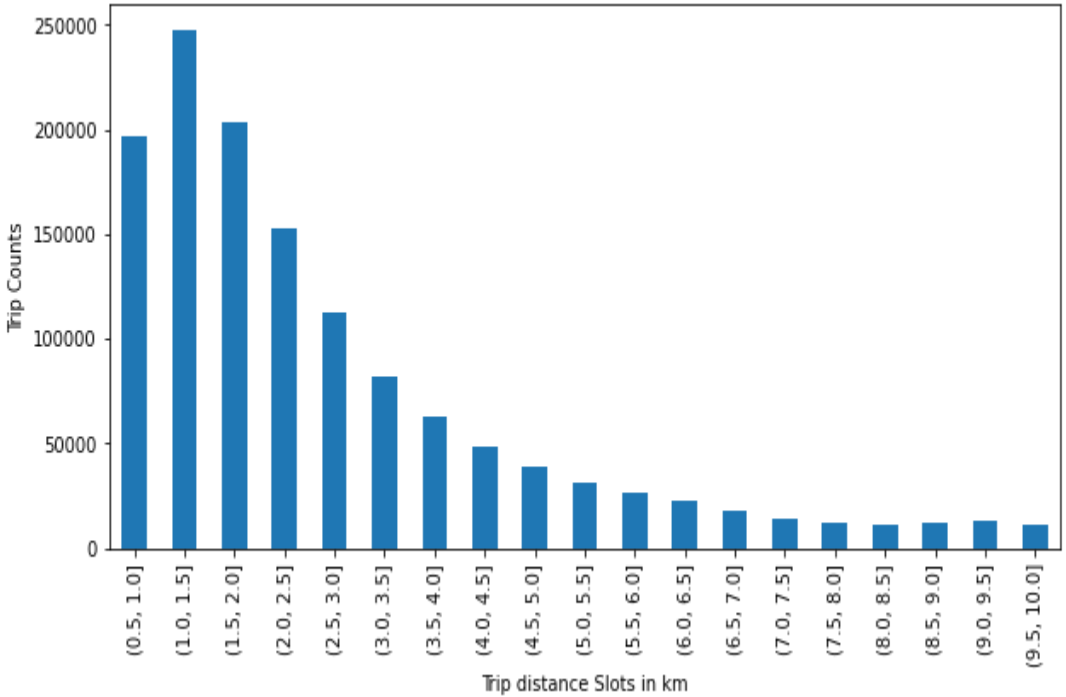
Distance


Density
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
200
400
600
800
1000
1200
distance

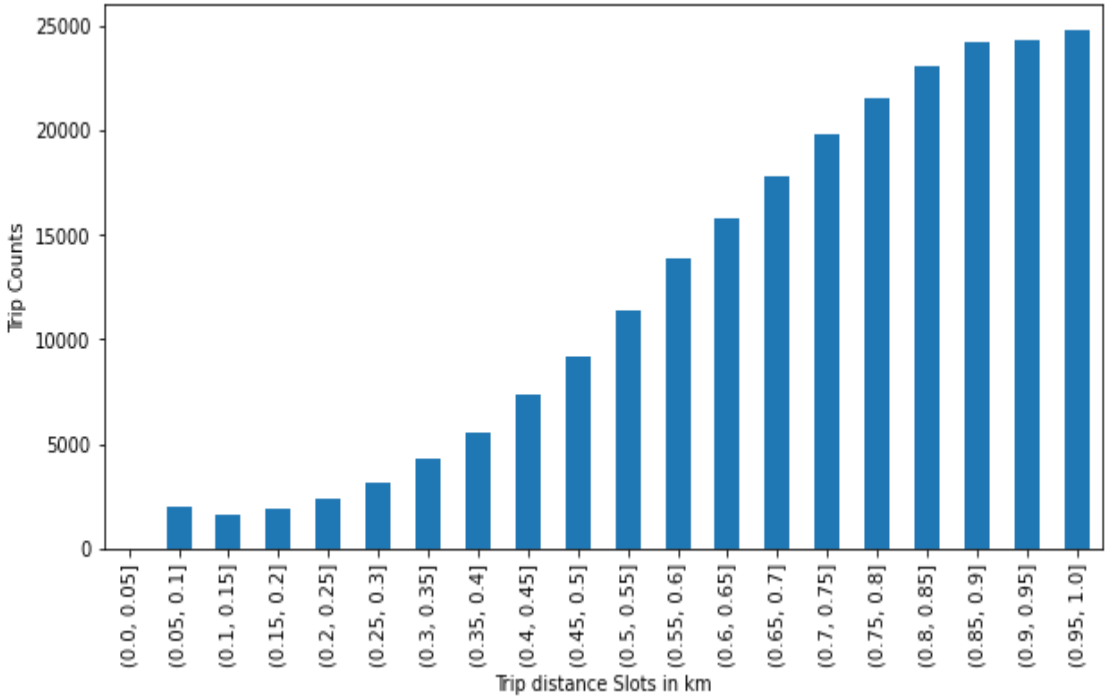

1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Trip Counts
(0, 100]
(100, 200]
(200, 300]
(300, 400]
(400, 500]
(500, 600]
(600, 700]
(700, 800]
(800, 900]
(900, 1000]
(1000, 1100]
Trip distance Slots in km

Distance


Trip Counts
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
(100, 200]
(200, 300]
(300, 400]
(400, 500]
(500, 600]
(600, 700]
(700, 800]
(800, 900]
(900, 1000]
Trip distance Slots in km


Trip Counts
250000
200000
150000
100000
50000
0
(0.5, 1.0]
(1.0, 1.5]
(1.5, 2.0]
(2.0, 2.5]
(2.5, 3.0]
(3.0, 3.5]
(3.5, 4.0]
(4.0, 4.5]
(4.5, 5.0]
(5.0, 5.5]
(5.5, 6.0]
(6.0, 6.5]
(6.5, 7.0]
(7.0, 7.5]
(7.5, 8.0]
(8.0, 8.5]
(8.5, 9.0]
(9.0, 9.5]
(9.5, 10.0]
Trip distance Slots in km

Distance

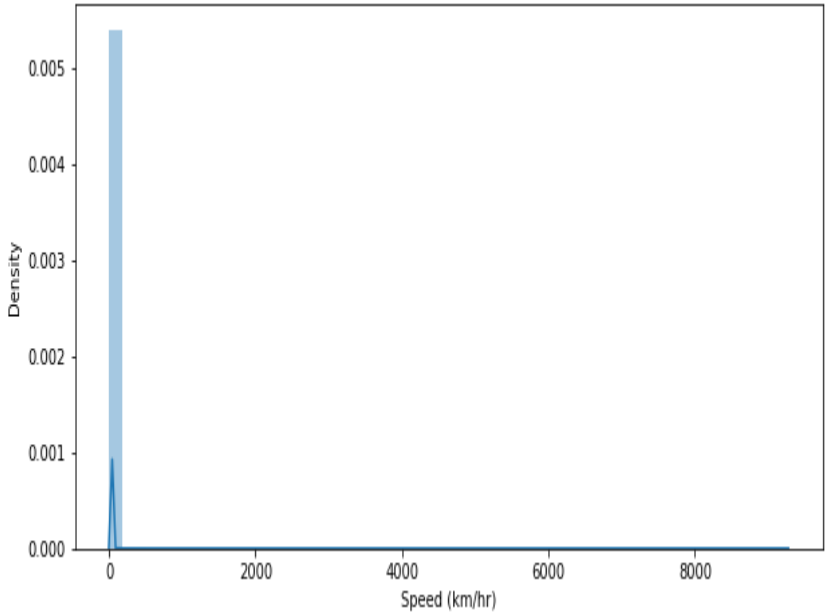


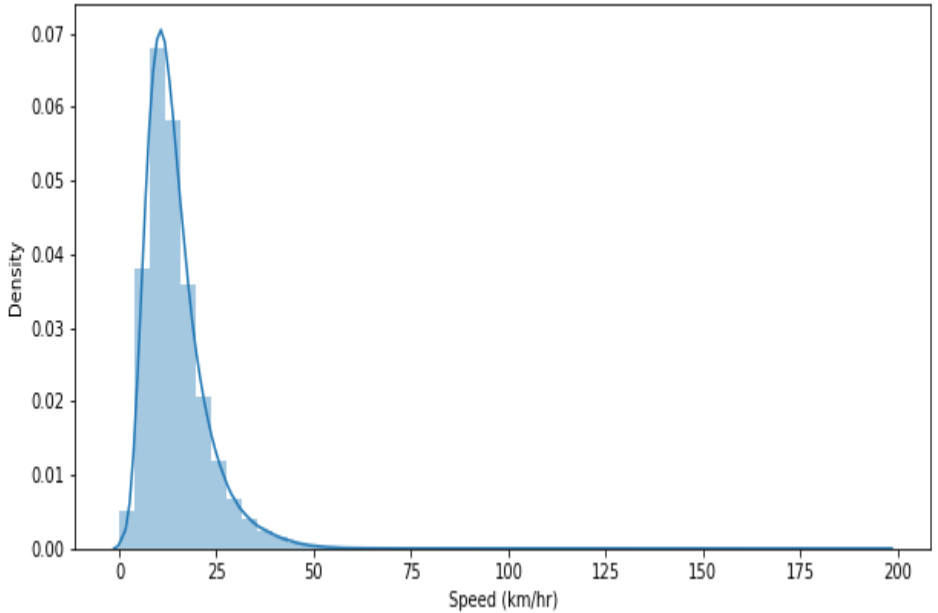
- There are **16 trips with more than 100 km** distance.
- There are **5887 trips with 0 km** distance. so this trip will unnecessary influence our model that we are dropping this rows

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

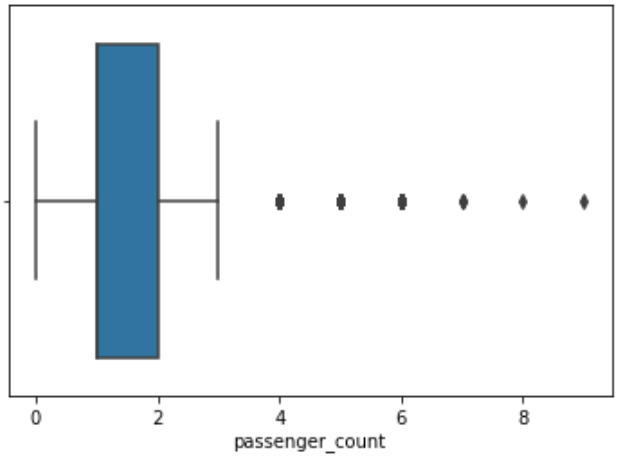


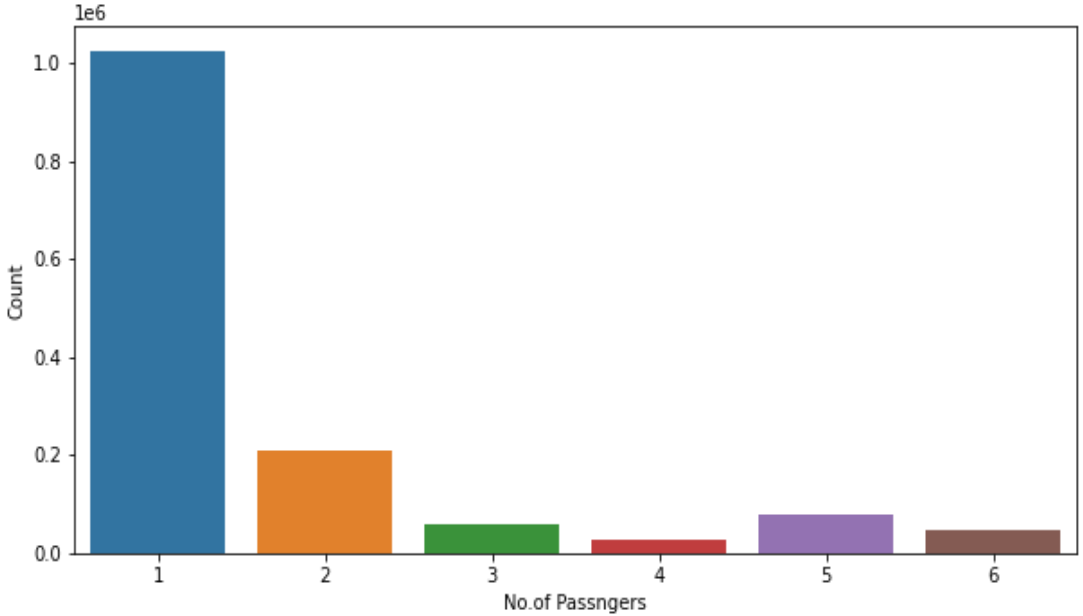


- There are some trips that were done at a **very high speed 8000 km/hr** + which is quite impossible. So we have applied some speed limit for dataset.
- As per the rule in NYC, the speed limit is 25 mph(approx. 40km/h) in New York City.
- **Mostly trips are done at a speed range of 5-25 km/hr.**

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

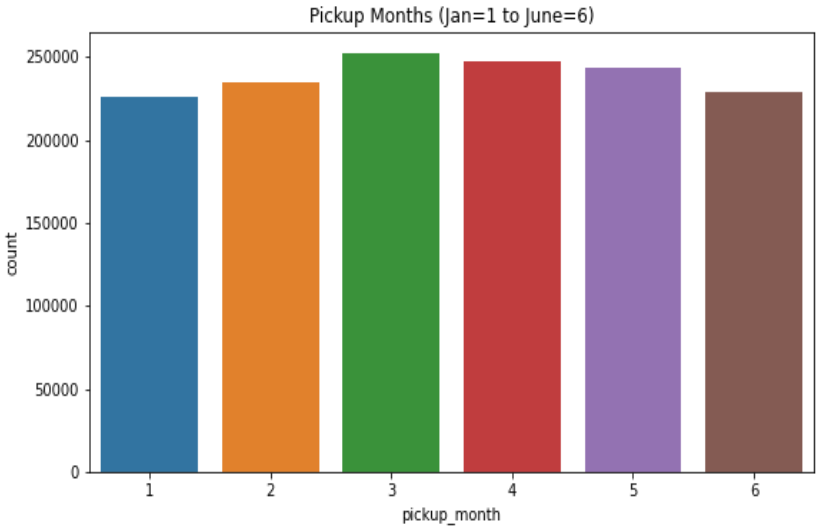


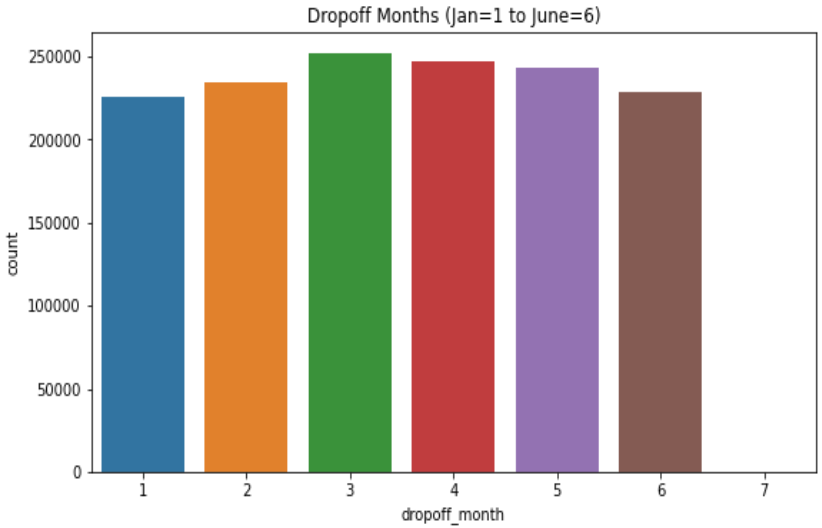


Let us remove the rows which have 0, 8, 7 and 9 passenger count

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

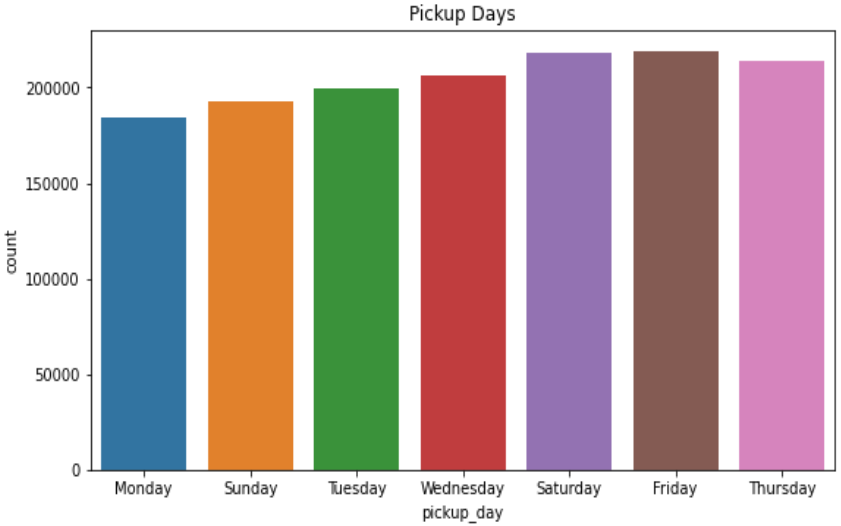
Pickup month & Dropup month

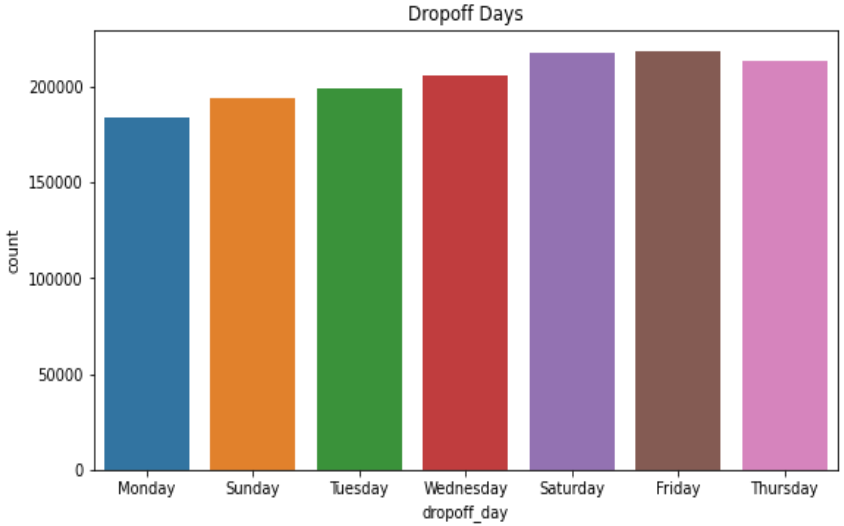




There is not much difference in the number of trips across months.

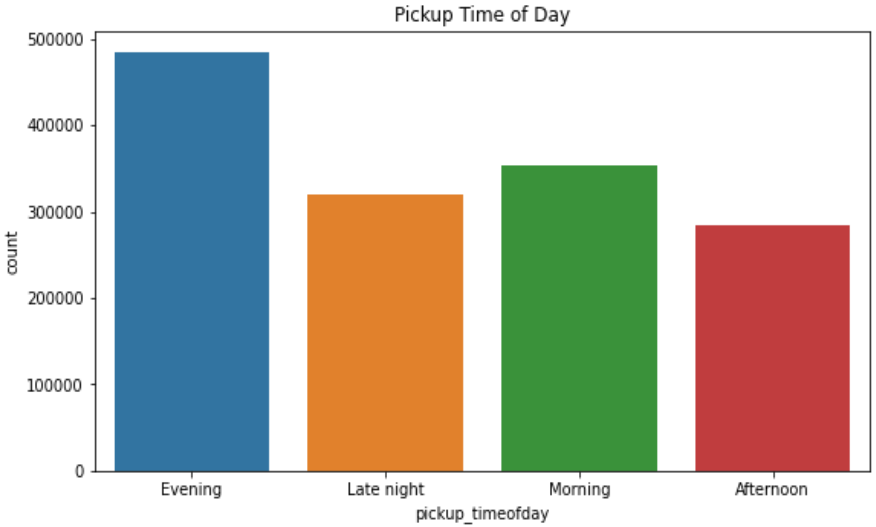
Pickup day & Dropup day

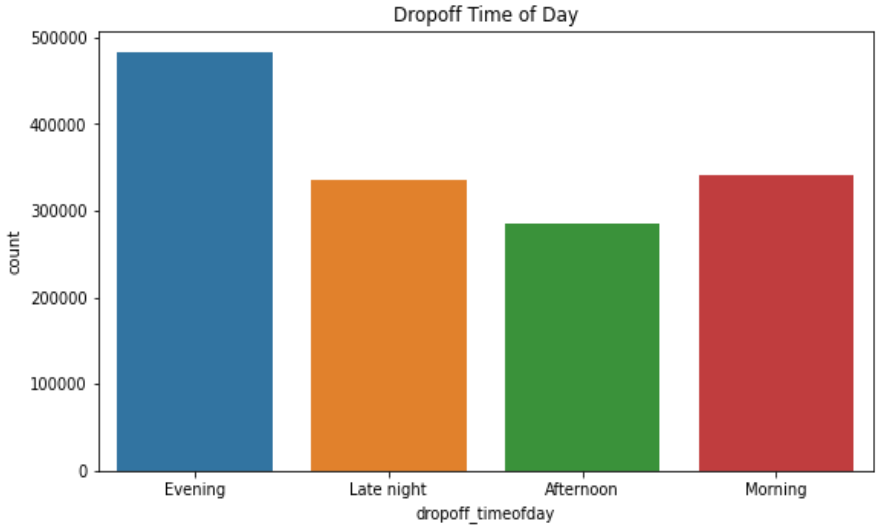




We see Fridays are the busiest days followed by Saturdays. That is probably because it’s weekend.

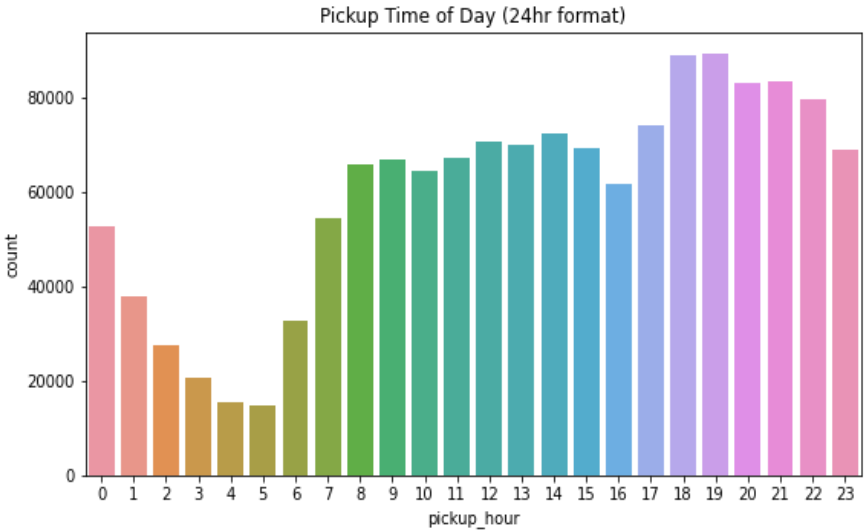
Pickup time of day & Dropoff time of day

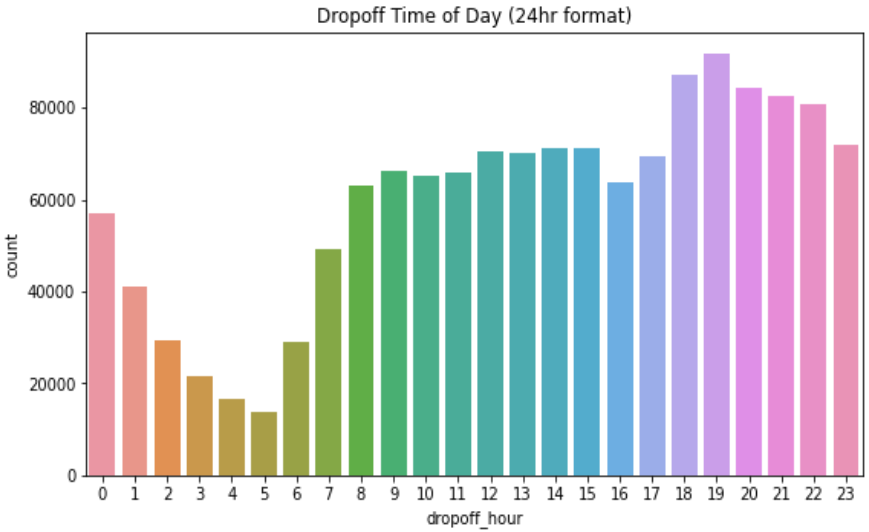




As we saw above, evenings are the busiest.

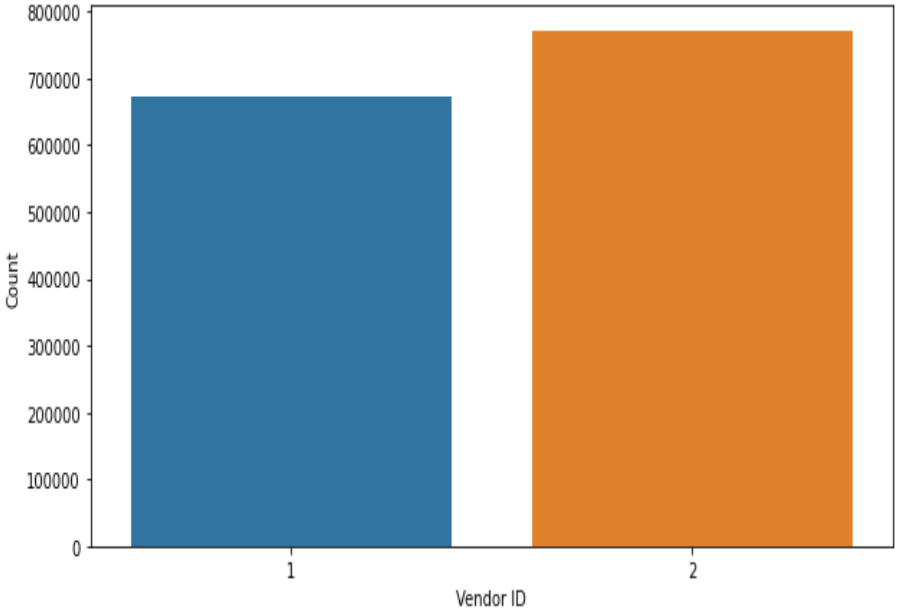
Pickup hour & Dropup hour

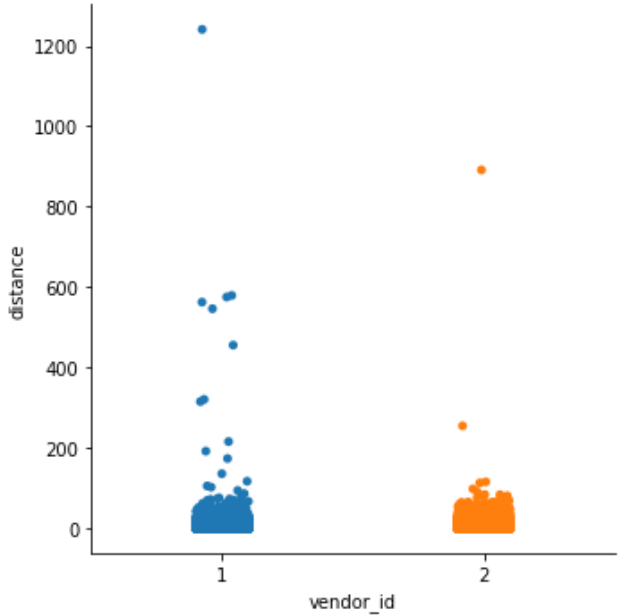




We see the busiest hours are 6:00 pm to 7:00 pm which makes sense as this is the time for people to return home from work.

Vendor id

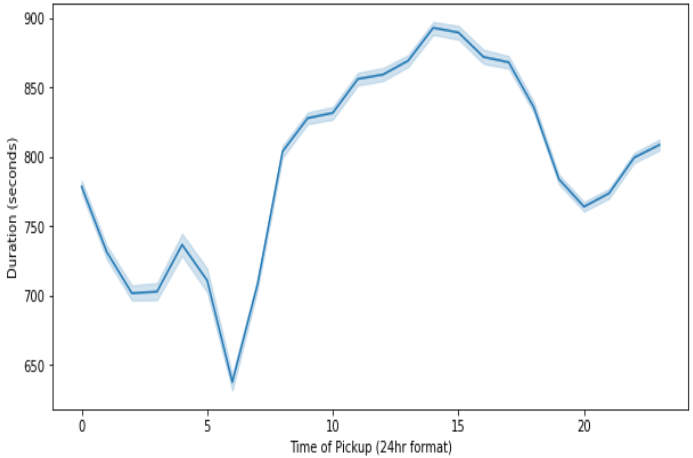




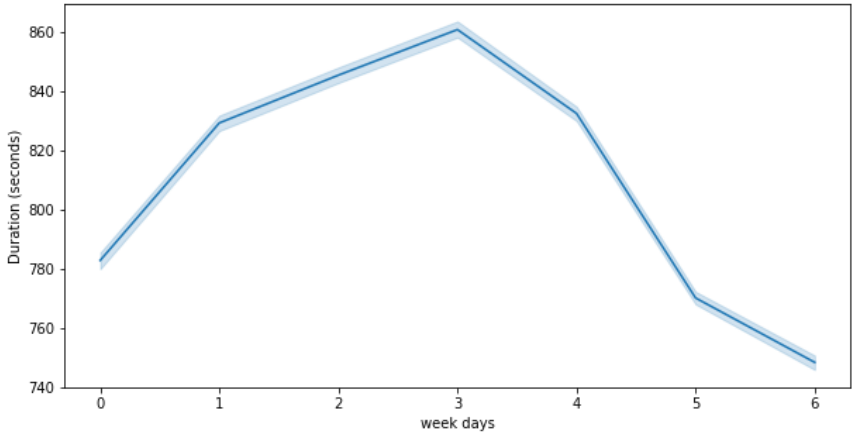
So vender 2 is more famous as compare to vender 1 & Vender 1 is taking more long trips as compare to vender 2

EDA
Bivariate Analysis

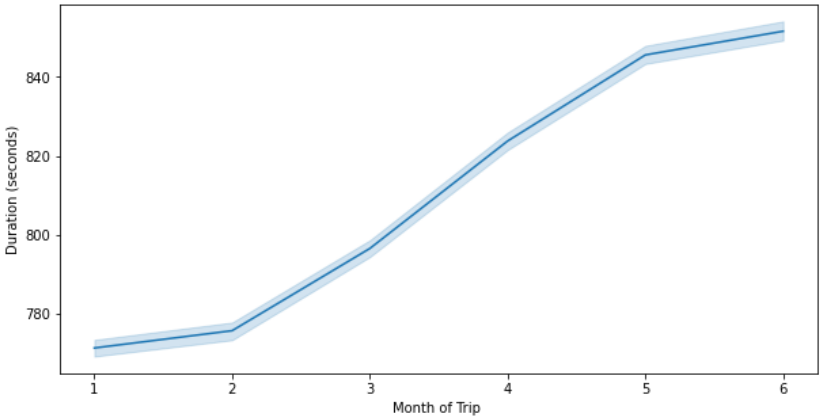
Trip Duration per hour



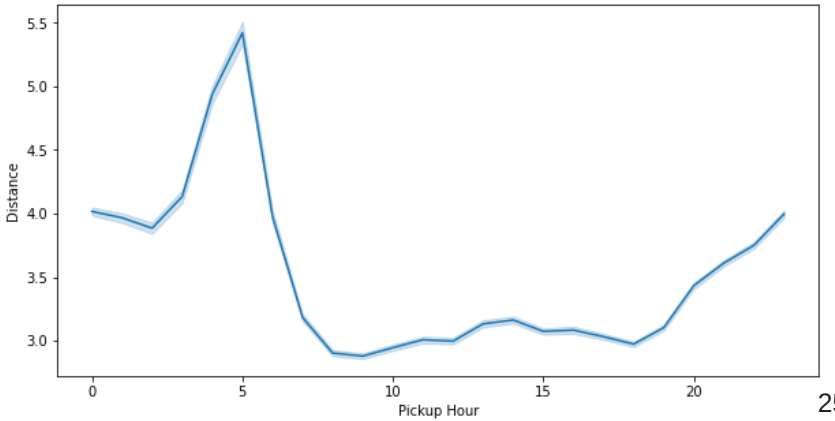
Trip duration per weekday



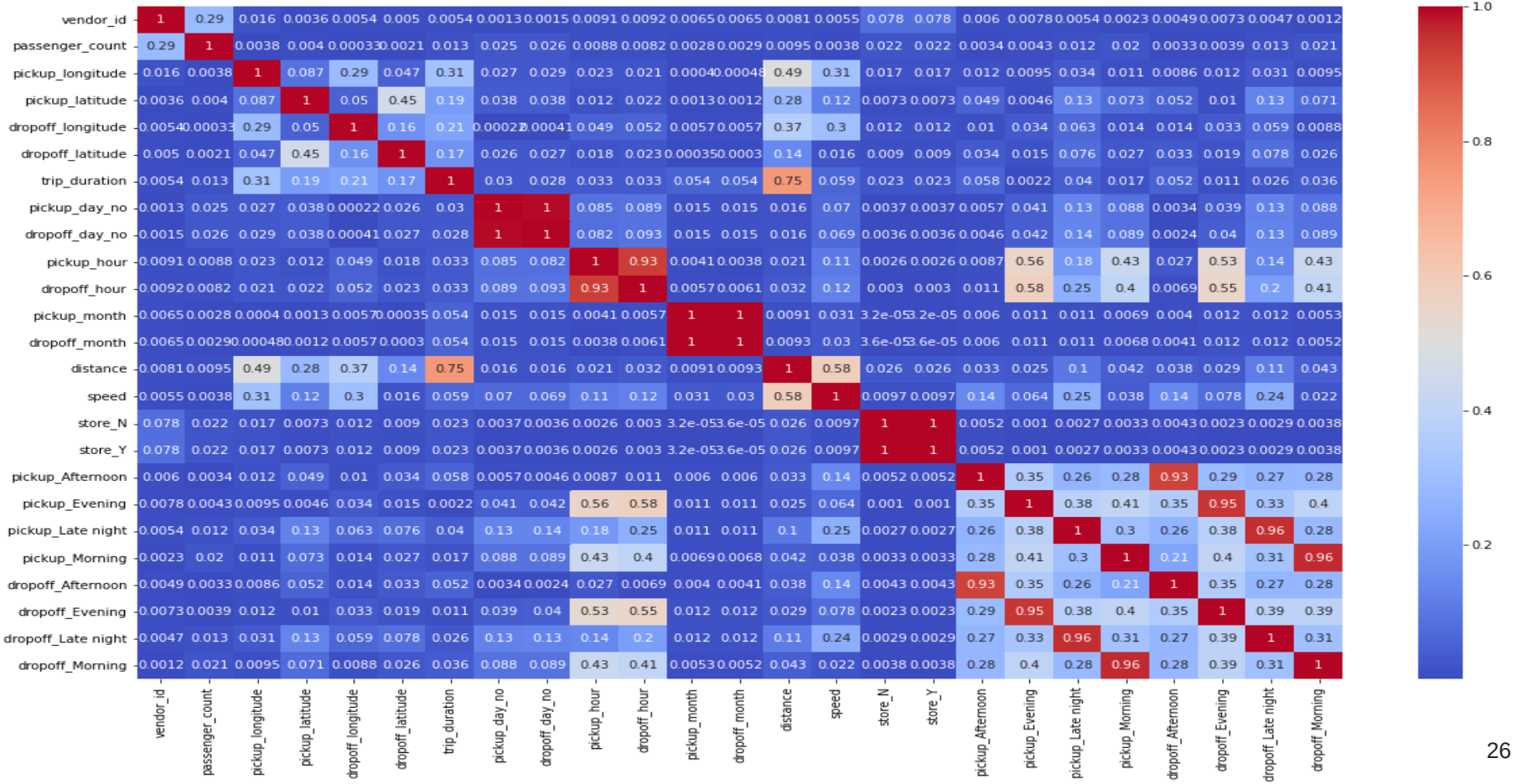
Trip duration per month

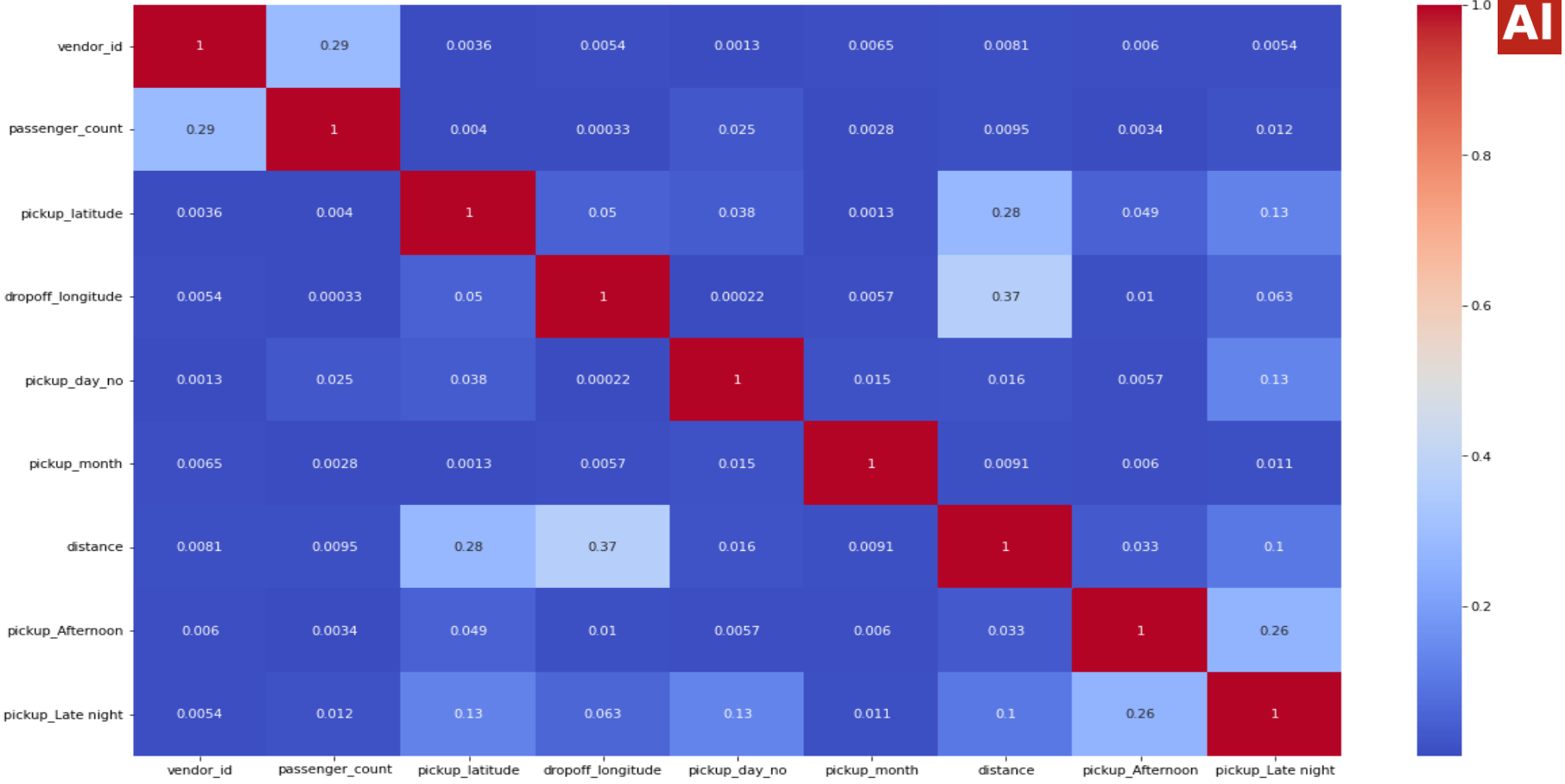


Distance and Hour



Feature Correlation





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

Feature Selection

Independent variables

- Pickup latitude
- Pickup longitude
- Dropoff latitude
- Dropoff longitude
- Distance
- Pickup hour

Dependent variables

- Trip duration

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)
```

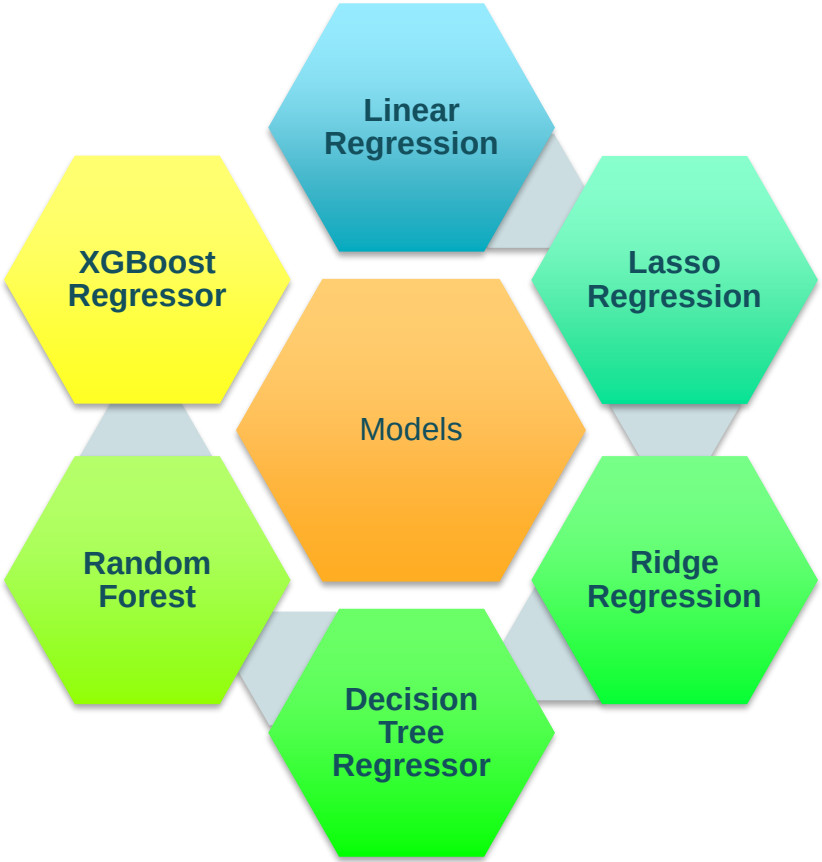
Splitting the data in train and test sets

```
[ ] #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)
```

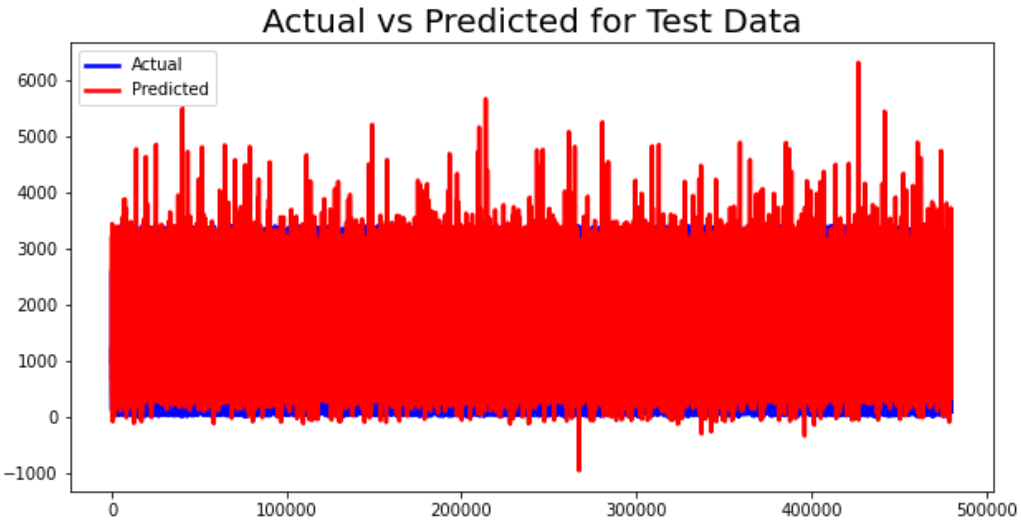
Normalizing the Dataset using Standard Scaling Technique.
It is because Standard Scaling adjusts the value of mean and standard deviation between 0's and 1's , which tend to work better for optimization techniques like Gradient descent.

Machine Learning Models



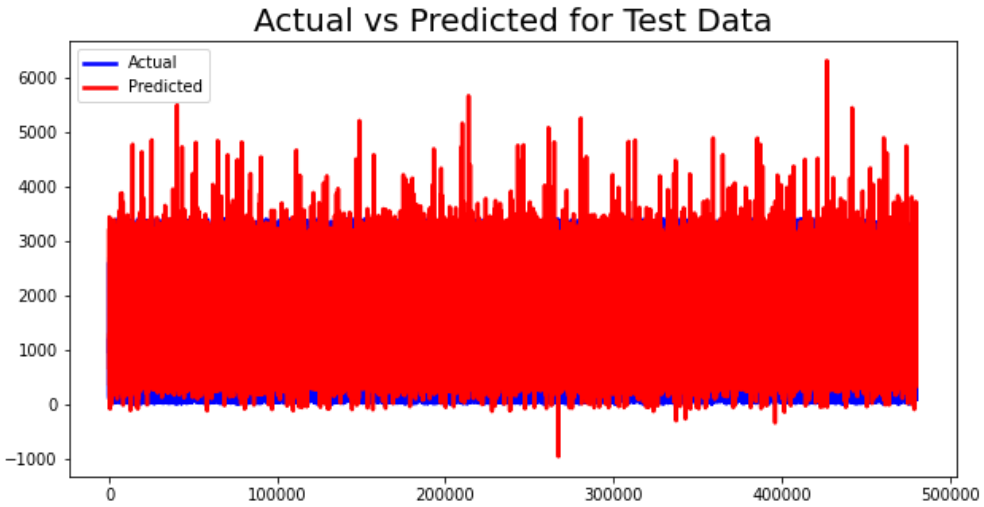
Linear Regression

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



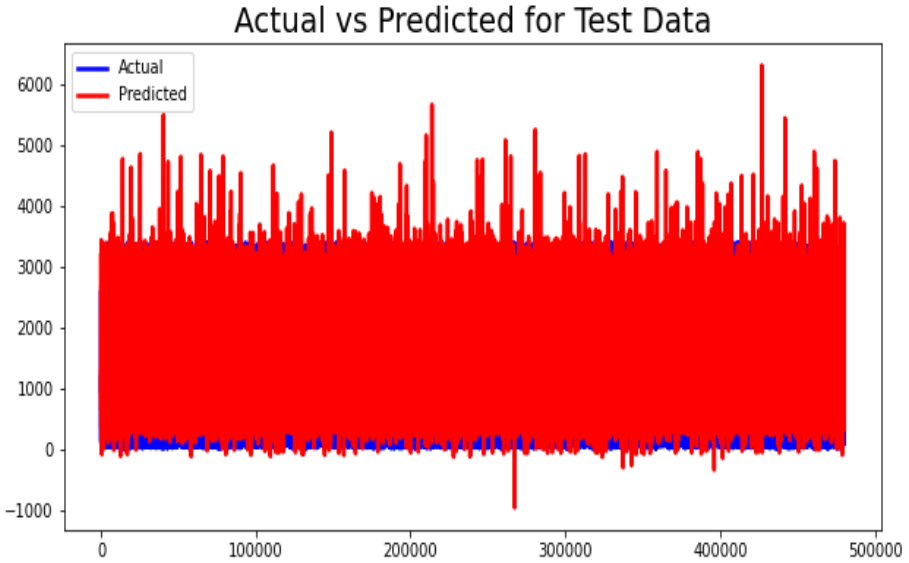
Lasso Regression

Error Matrices	Train Data	Test Data
MSE	120625.237	121458.077
R squared	0.6310	0.62648
Adjusted R squared	0.6310	0.62648



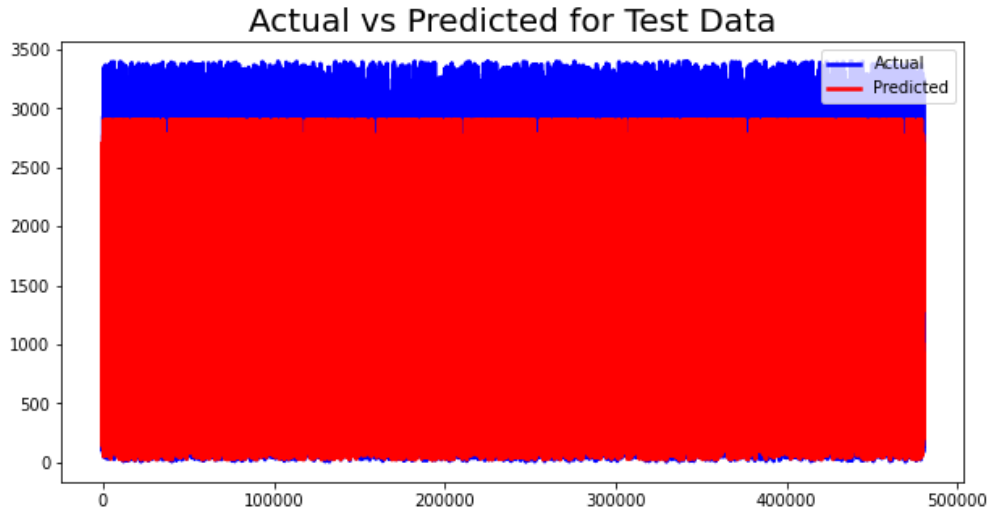
Ridge Regression

Error Matrices	Train Data	Test Data
MSE	120625.2374	121458.062
R squared	0.63103	0.62648
Adjusted R squared	0.63103	0.62648



Decision Tree Regression

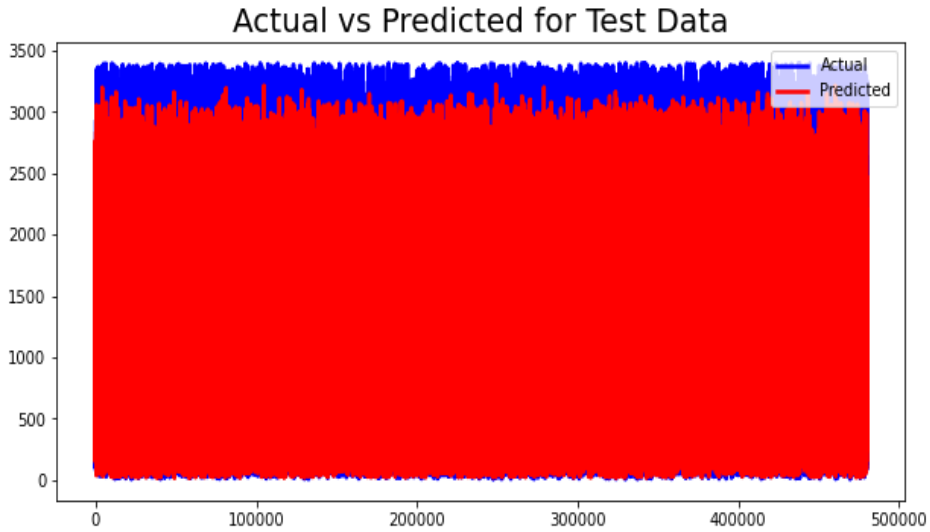
Error Matrices	Train Data	Test Data
MSE	74874.505	76851.041
R squared	0.77097	0.7636
Adjusted R squared	0.77097	0.7636



- 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

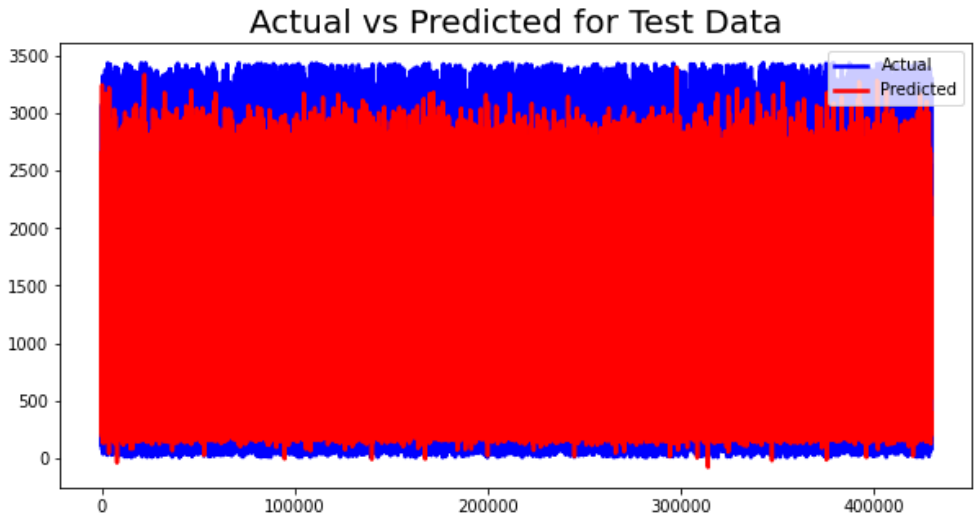
Error Matrices	Train Data	Test Data
MSE	9197.6164	63003.2482
R squared	0.9718	0.8062
Adjusted R squared	0.9718	0.8062



The best fit alpha value is found out to be : (n_estimators=40, n_jobs= -4)

XGBooster Regression

Error Matrices	Train Data	Test Data
MSE	52591.28905	59863.998
R squared	0.8391	0.8159
Adjusted R squared	0.8391	0.8159



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)

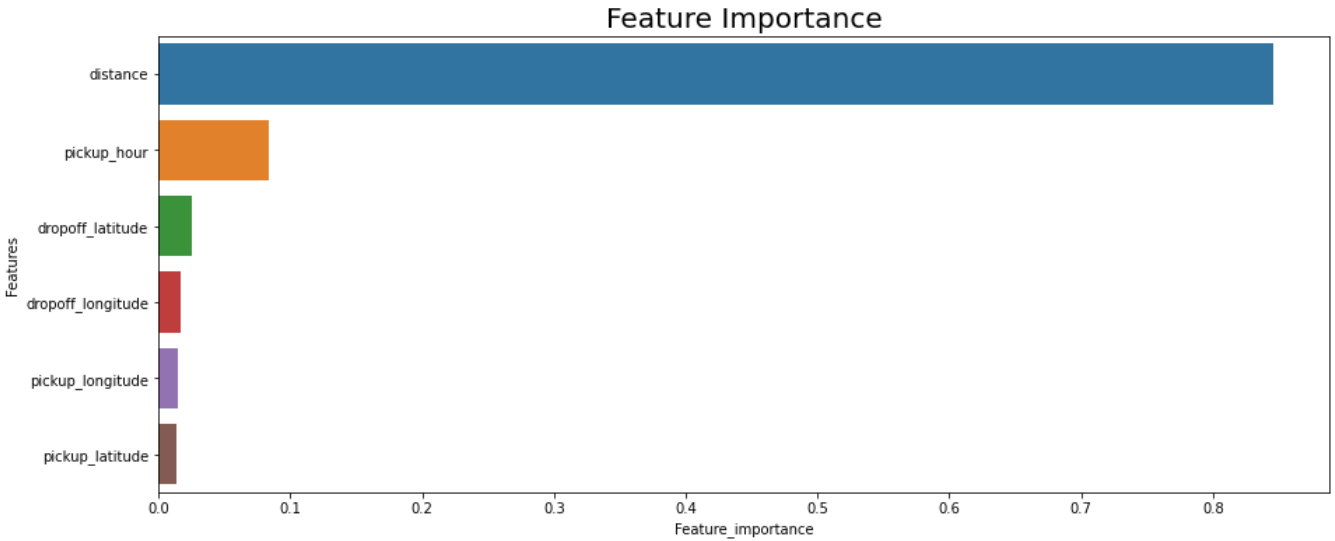
The R2 score using the same alpha is : `0.8151701094551514`

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	Ramdom Forest	9197.616442	63003.248293	0.971866	0.806250	0.971866	0.806248
5	XGBoost Regressor	52591.289051	59863.998326	0.839133	0.815904	0.839132	0.815902

Important Feature

	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 **XGBoost Regressor.**
- The Linear models don't show good performance on our training and testing environment.
- From above table we can conclude that **XGBoost Regressor** gives us **R2=81%.** Which is the best model as compare to the other models to predict the trip duration for a NYC taxi.

Thank You