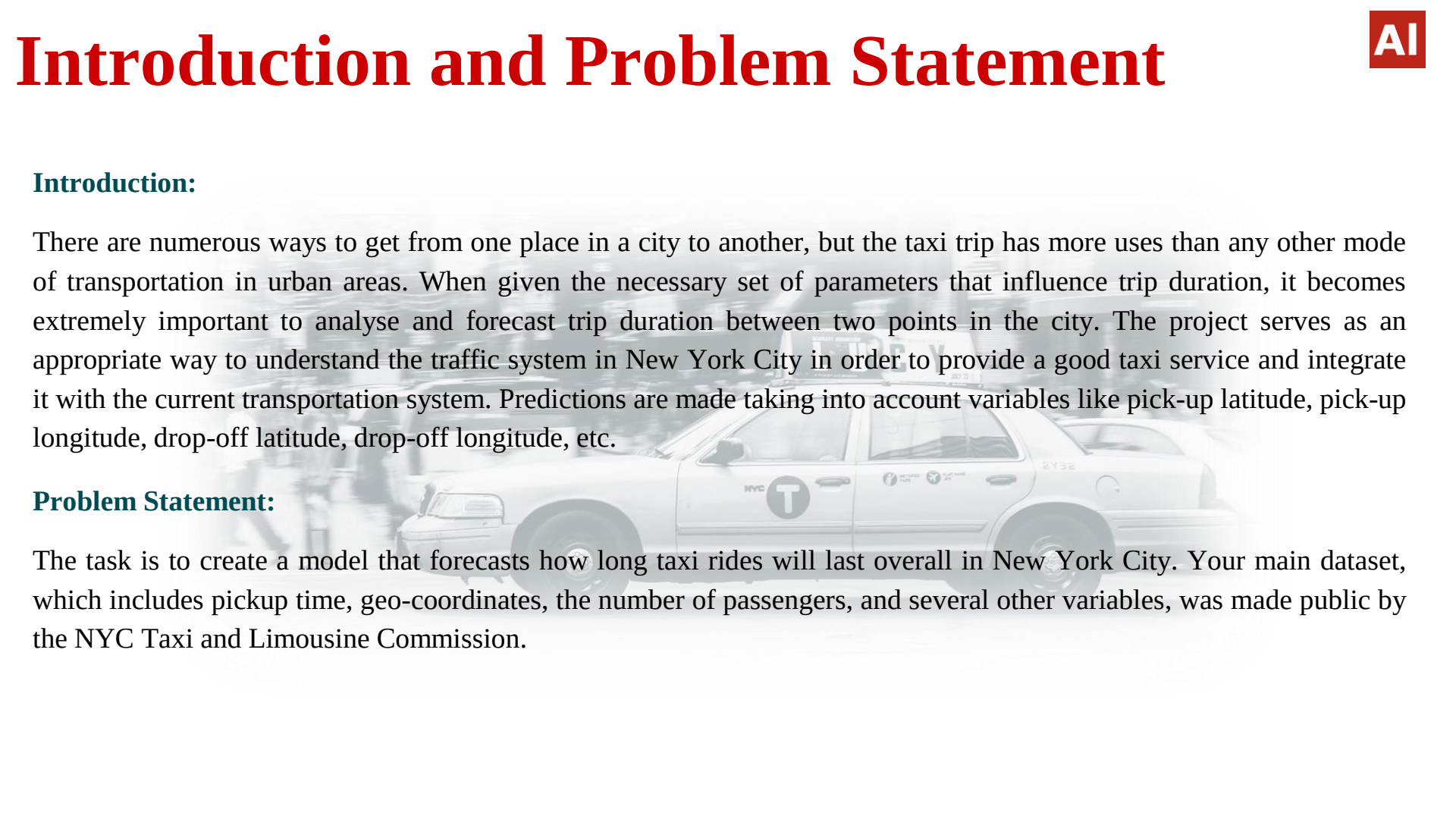
Capstone Project
Supervised ML - Regression
NYC Taxi Trip Time Prediction

Completed By :
Ankush Kumar

Points to Discuss:

- Introduction and Problem Statement
- Data Exploration
- Analysis Methodology
- Data Processing and Exploratory Data Analysis(EDA)
- Modeling
- Implementing PCA (Principle Component Analysis)
- Final Modeling and Results
- Conclusions and Summary

Introduction and Problem Statement



Introduction:

There are numerous ways to get from one place in a city to another, but the taxi trip has more uses than any other mode of transportation in urban areas. When given the necessary set of parameters that influence trip duration, it becomes extremely important to analyse and forecast trip duration between two points in the city. The project serves as an appropriate way to understand the traffic system in New York City in order to provide a good taxi service and integrate it with the current transportation system. Predictions are made taking into account variables like pick-up latitude, pick-up longitude, drop-off latitude, drop-off longitude, etc.

Problem Statement:

The task is to create a model that forecasts how long taxi rides will last overall in New York City. Your main dataset, which includes pickup time, geo-coordinates, the number of passengers, and several other variables, was made public by the NYC Taxi and Limousine Commission.

Data Exploration:

Dataset fil format: CSV(Comma Separated) file is used

Name of the data-source file : NYC Taxi Data.csv

Number of columns : 11

Number of rows : 1458644

Number of Numerical columns are: 7

Number of Categorical columns are: 2

Missing Values in the dataset

```
print(data.isnull().sum())
```

```
id                    0
vendor_id             0
pickup_datetime       0
dropoff_datetime      0
passenger_count       0
pickup_longitude      0
pickup_latitude       0
dropoff_longitude     0
dropoff_latitude      0
store_and_fwd_flag    0
trip_duration         0
dtype: int64
```

No missing value in any feature of our data.

Dataset Summary

Feature Columns / Variables:

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

Feature Columns / Variables:

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

Target Column/Variable:

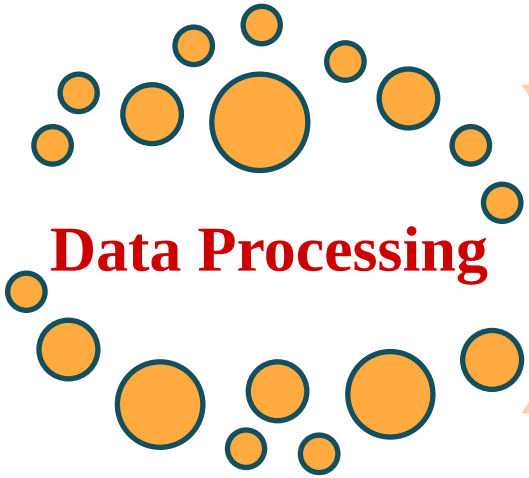
trip_duration - duration of the trip in seconds.

Unique Values and Five Point Statistical Summary

	Column_name	Number_of_Unique_Values
0	vendor_id	2
1	store_and_fwd_flag	2
2	passenger_count	10
3	trip_duration	7417
4	pickup_longitude	23047
5	dropoff_longitude	33821
6	pickup_latitude	45245
7	dropoff_latitude	62519
8	pickup_datetime	1380222
9	dropoff_datetime	1380377
10	id	1458644

	count	mean	std	min	25%	50%	75%	max
vendor_id	1458644.0	1.534950	0.498777	1.000000	1.000000	2.000000	2.000000	2.000000e+00
passenger_count	1458644.0	1.664530	1.314242	0.000000	1.000000	1.000000	2.000000	9.000000e+00
pickup_longitude	1458644.0	-73.973486	0.070902	-121.933342	-73.991867	-73.981743	-73.967331	-6.133553e+01
pickup_latitude	1458644.0	40.750921	0.032881	34.359695	40.737347	40.754101	40.768360	5.188108e+01
dropoff_longitude	1458644.0	-73.973416	0.070643	-121.933304	-73.991325	-73.979752	-73.963013	-6.133553e+01
dropoff_latitude	1458644.0	40.751800	0.035891	32.181141	40.735885	40.754524	40.769810	4.392103e+01
trip_duration	1458644.0	959.492273	5237.431724	1.000000	397.000000	662.000000	1075.000000	3.526282e+06

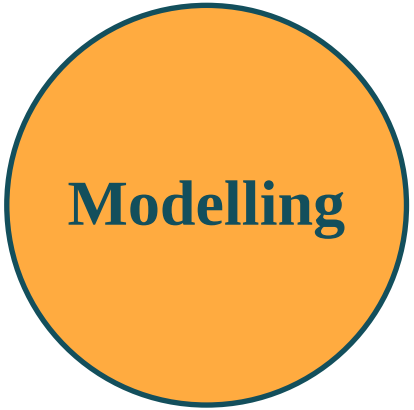
Analysis Methodology



- Exploratory Data Analysis (EDA)
- Outlier Analysis
- Correlation Analysis

Feature Transformation

- Label Encoding
- Standardizing Features

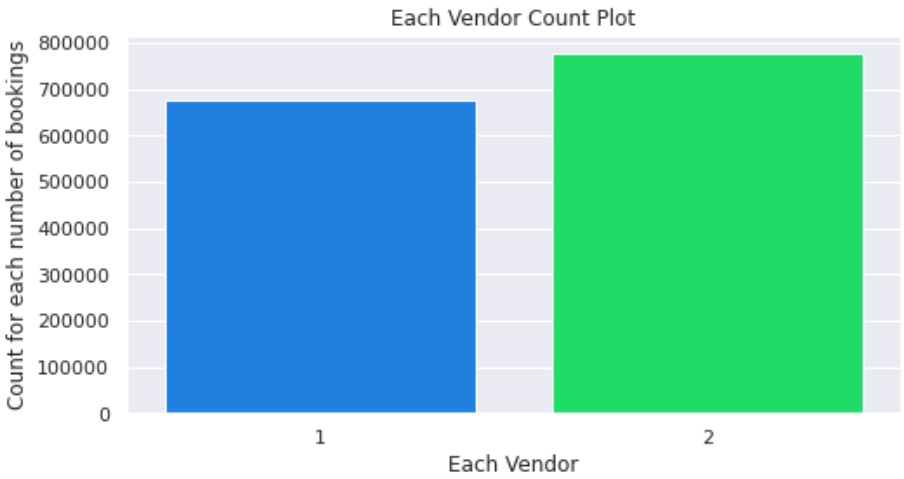


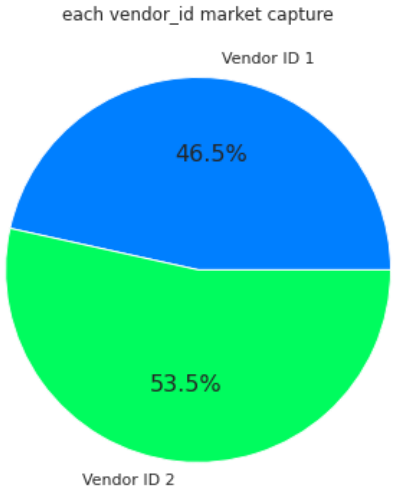
- Implementation of PCA
- Final Modelling

Exploratory Data Analysis (EDA)

Univariate Analysis

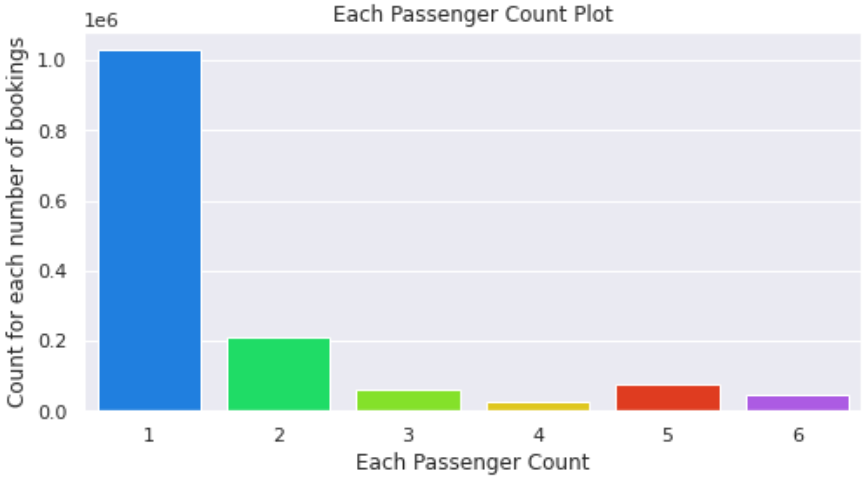
Checked how much bookings are done for each vendor type





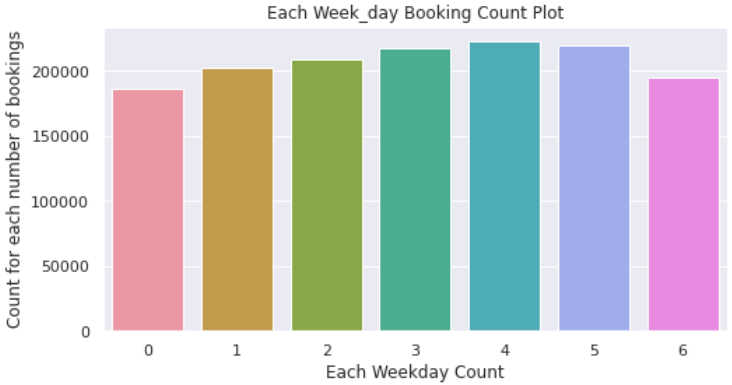
- **We can see that for vendor 2 there are more number of bookings which is of 54 %**

Checked how much bookings are done for each passenger_count



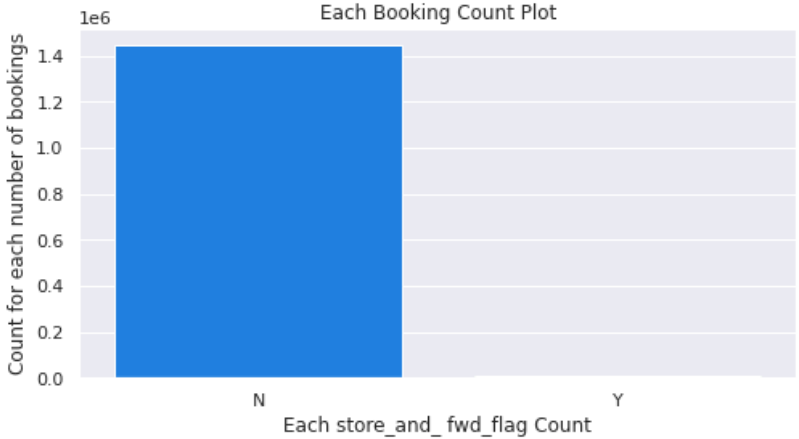
Taxi ride is booked by only a single person there are more number of bookings is high as compare to multiple people booking the taxi ride.

•Checked the rate of bookings according to week days



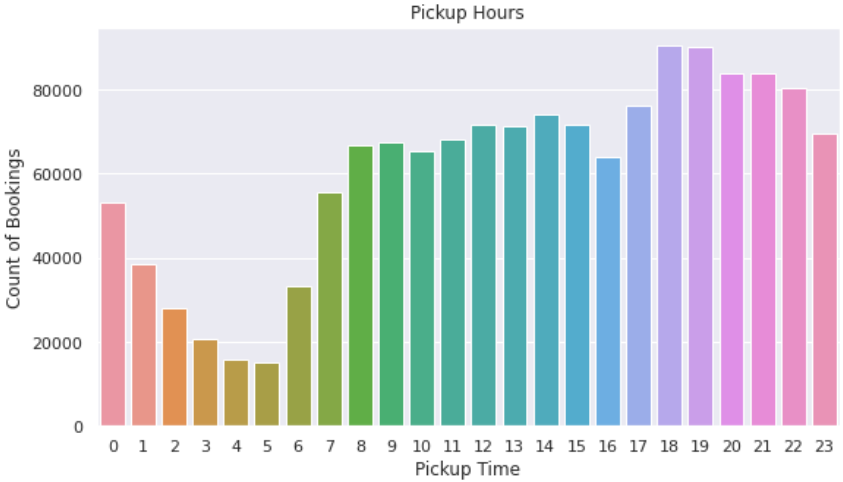
Weekends 4 - Friday | 5 - Saturday there are high booking rate for taxi as compare to other days.

Checked how much bookings are done for each store_and_ fwd_flag Count



Very less count when the store_and_ fwd_flag is Marked Yes and most of the time the taxi rider haven't connected to the servers of Vendor.

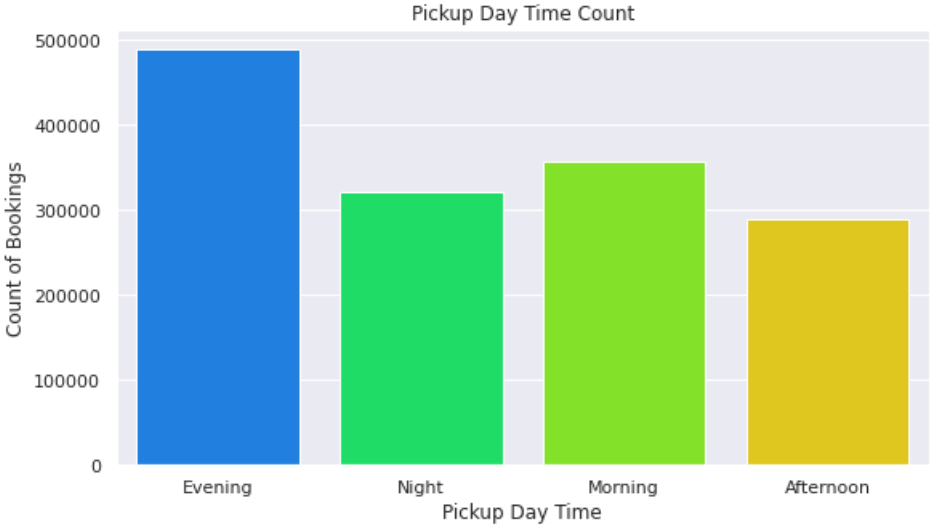
Checked number of booking hourly



We can observe that morning after 10 O'clock people use to book taxi because they want to go out to their work places.

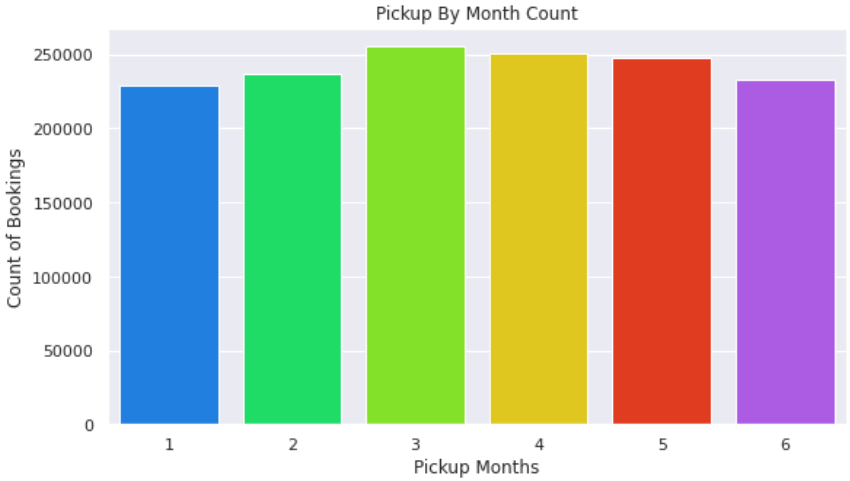
And at in the evening after 6 O'clock the taxi demand tends to in peak.

Checked the demand of taxi booking in each kind of day time



pickup and dropoff day time count plot booking count is maximum in the EVENING day time.

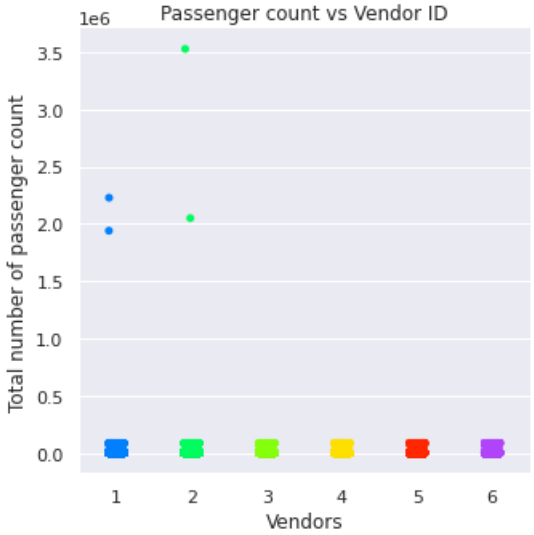
checked the bookings done per month



As we can see that in month of march and April there are more number of taxi booking occurred.

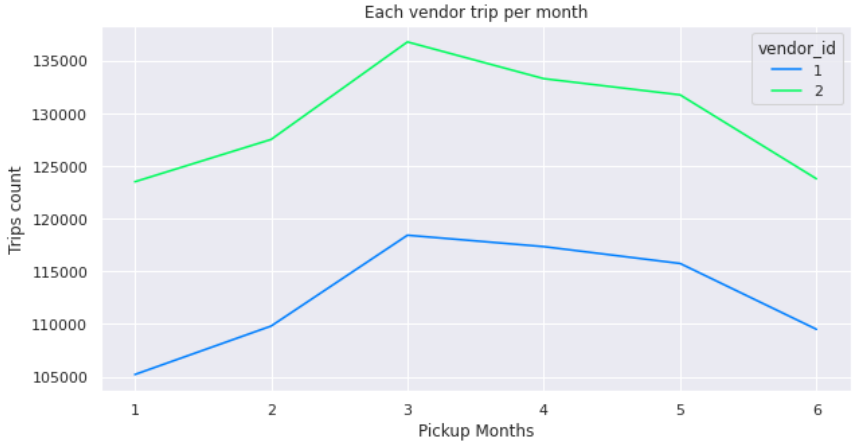
Bivariate and Multivariate Analysis

Checking monthly popularity of vendor



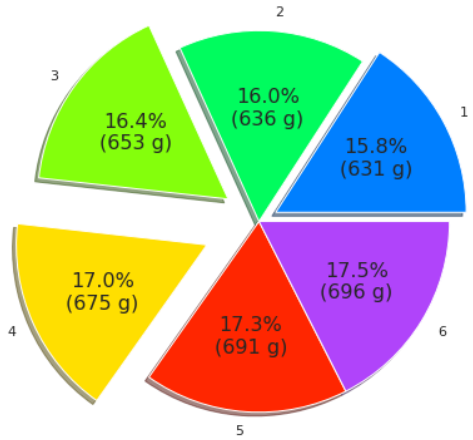
vendor 2 is more popular than vendor 1

compared vnedor_id with monthly pickups

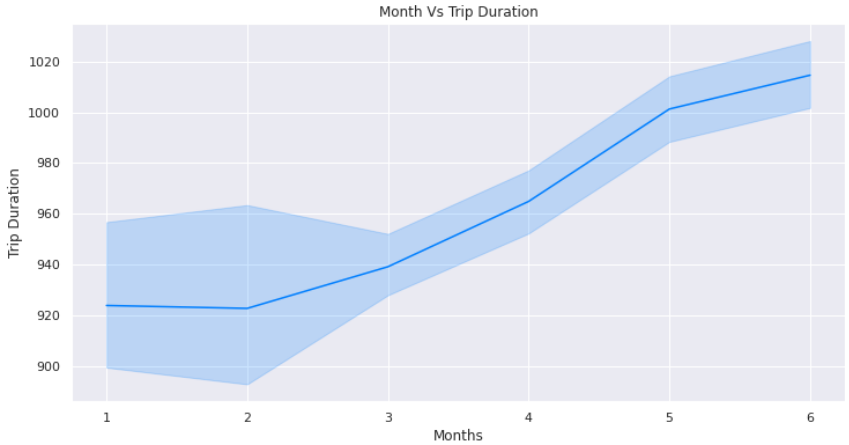


Both the vendors have the highest strips during the month of march and lowest at the Jan, Feb, and after June.

Each monthly average trip duration



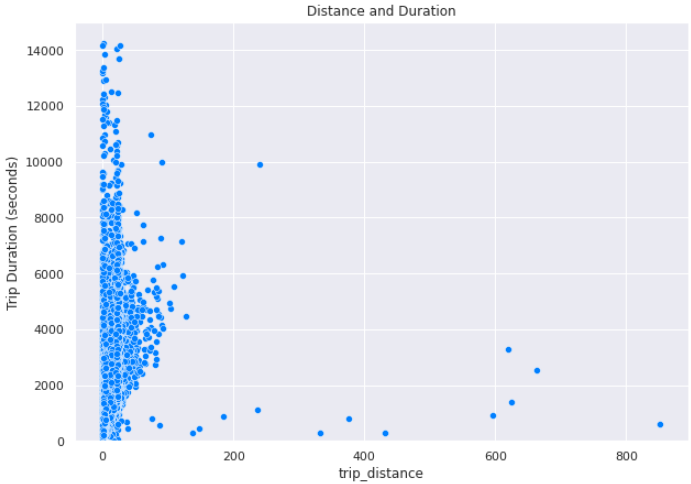
Checked monthly trip duration



We can observe that least trip duration occured in january and highest started occuring after the month of march.

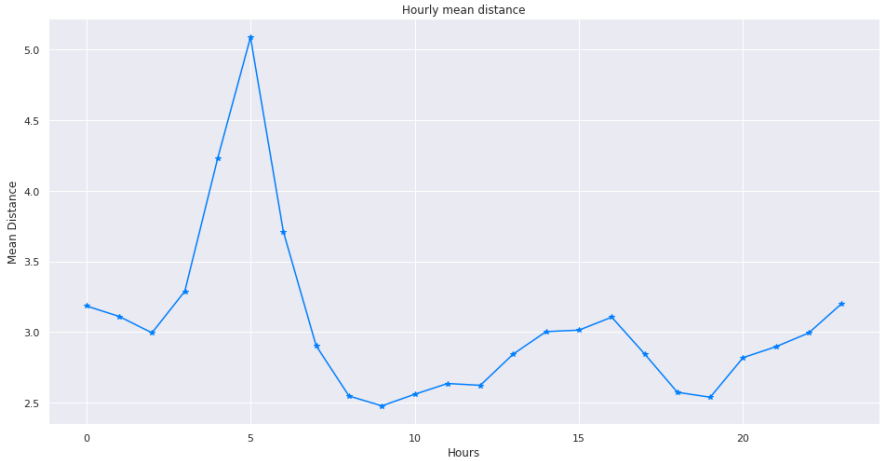
As we can see that trip duration start increases from 3rd month before that it is quite constant.

Compared trip_duration and distance.



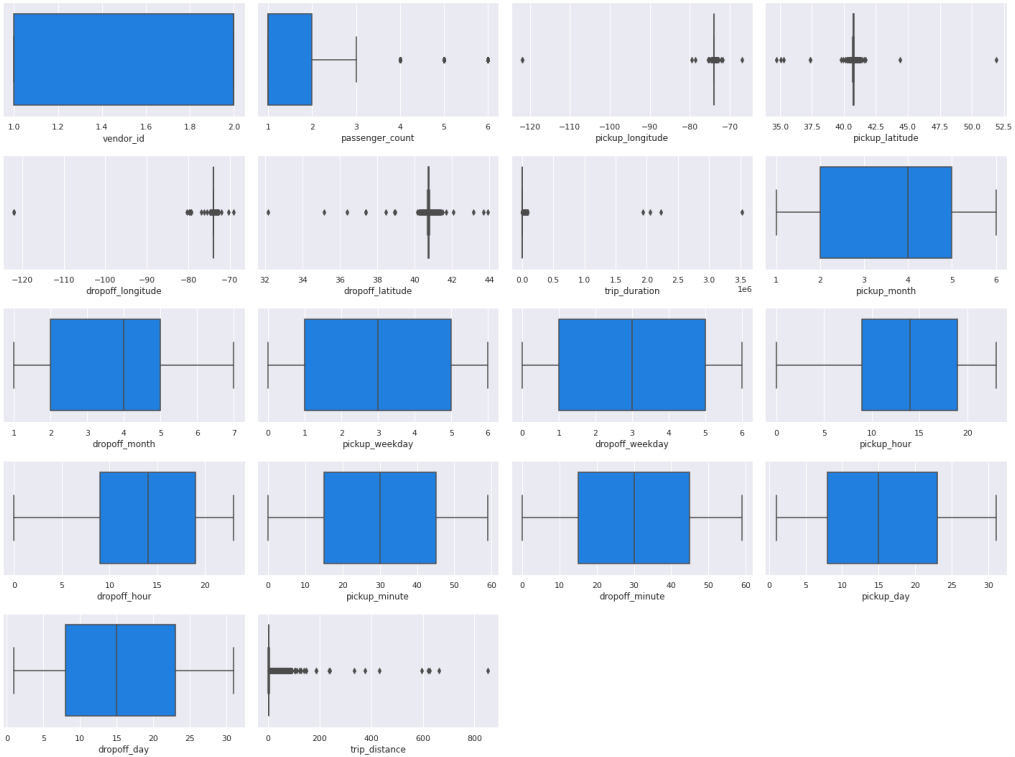
We can see that 0-100 km distance have the dense trip durations.

Compared mean distance for each hour



The longest average distance is travelled from late at night until early in the morning.

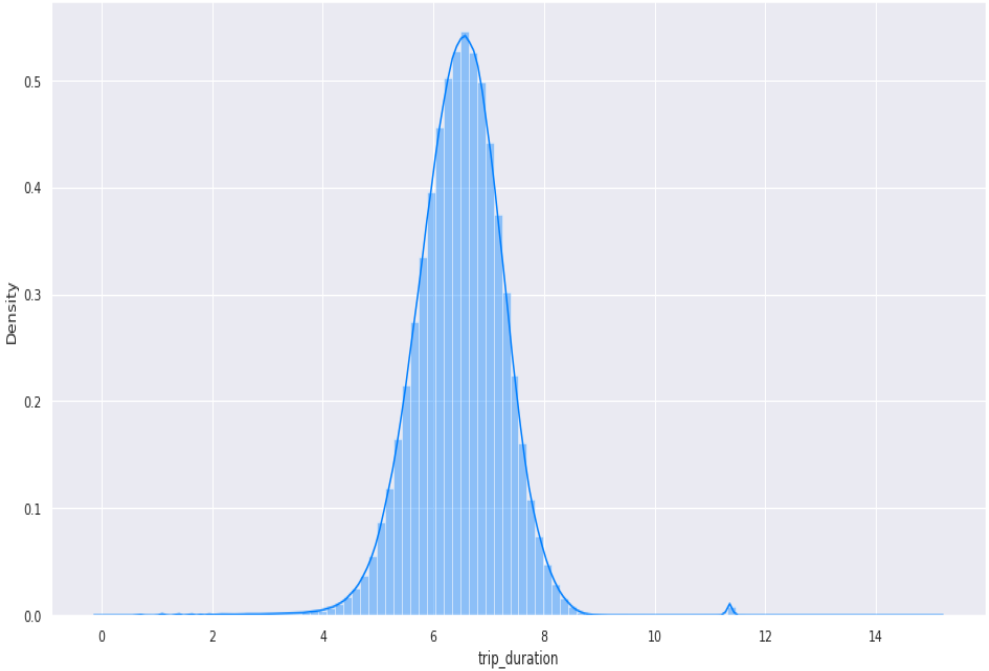
Outlier Analysis



As we already saw that trip duration has some trips that are not normal which are very higher like 900 + hrs in time duration but also we can see that there are some trip of no time.

If we try to remove that values till third standard deviation than also there will be outliers as trip duration contains duration of various and unique time let us try to transform our data so that we can get some improvement in our trip duration column.

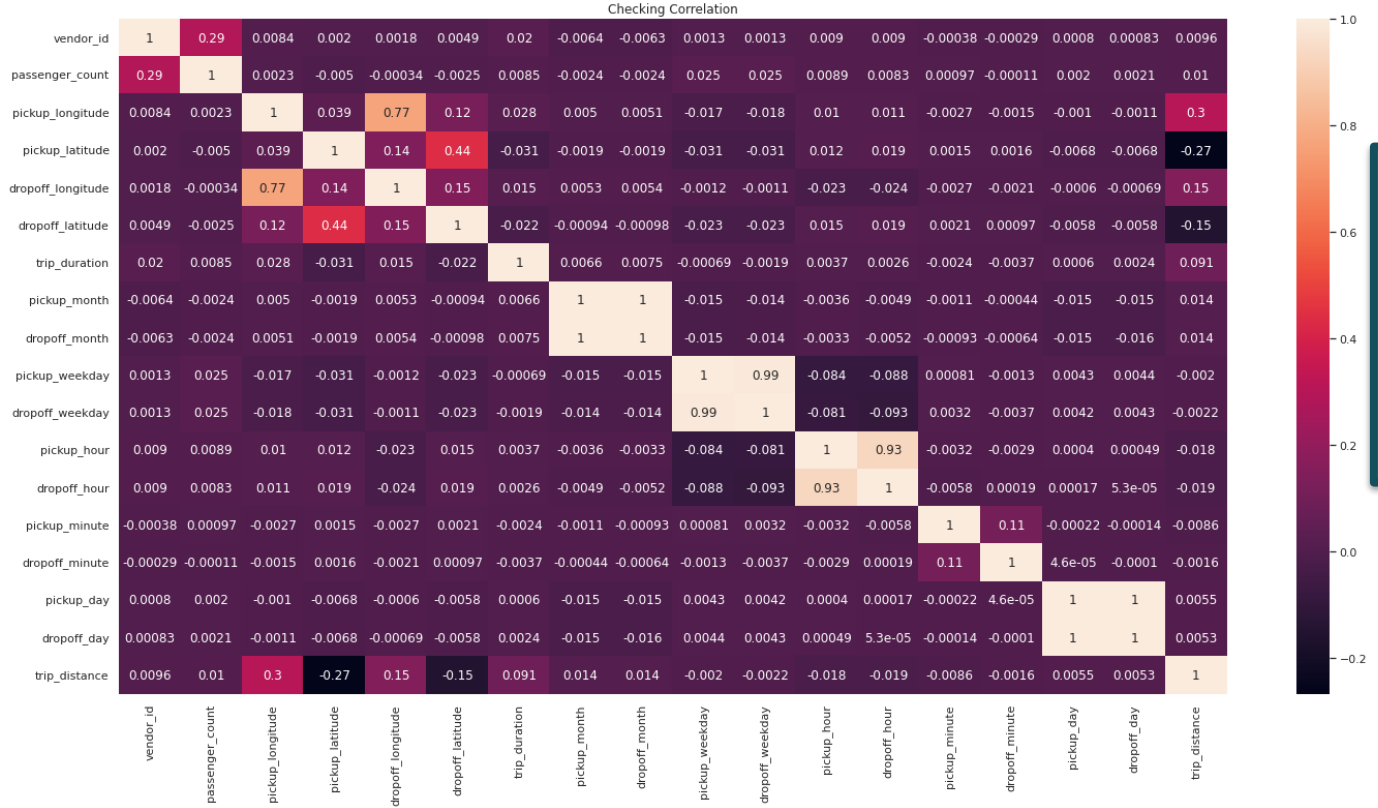
Outlier Analysis



OBSERVATION

Here we can observe that the distribution is now in normal distribution but also the is some small peak but we will use log transformed column for the training.

Correlation Analysis – Using Heatmap



•As we can see that 'dropoff_day', 'dropoff_hour', 'dropoff_month', 'dropoff_weekday' are highly corelated we can drop these features

Feature Transformation

We have many features that have some categorical values in them like **'store_and_fwd_flag', 'pickup_day_time', 'dropoff_day_time'**

We have applied LabelEncoder from Scikit Learn and transformed these features to numeric format.

LabelEncoder:

Sklearn is a very efficient tool for encoding categorical feature levels into numeric values. LabelEncoder encodes labels with values ranging from 0 to n classes-1, where n denotes the number of distinct labels.

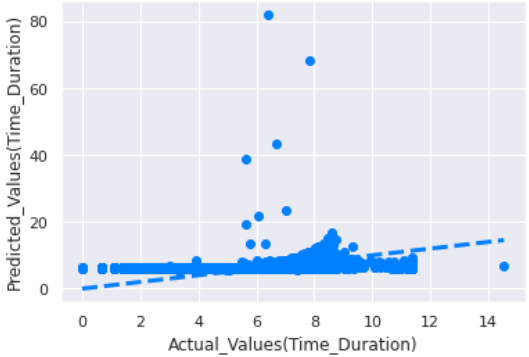
We have used StandardScaler from Scikit Learn and transformed these features to standard normal form.

StandardScaler:

By deducting the mean from a feature and scaling it to unit variance, StandardScaler standardises it. Divide all the values by the standard deviation to get the unit variance. The strict definition of scale that I previously presented is not adhered to by StandardScaler.

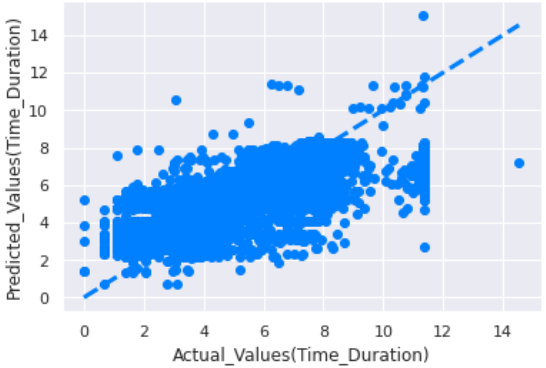
Modelling without implementing PCA

Linear Regression



	Training_Score	Testing_Score	R2_Score	ADJ_R2	MSE	RMSE
Linear Regression	0.298175	0.274146	0.274146	0.274117	0.449004	0.670078
Decision Tree Regression	0.610109	0.601903	0.601903	0.601887	0.246258	0.496244
Random Forest Regression	0.627799	0.621592	0.621592	0.621577	0.234078	0.483816
XGBOOST	0.924322	0.907907	0.907907	0.907904	0.056967	0.238678

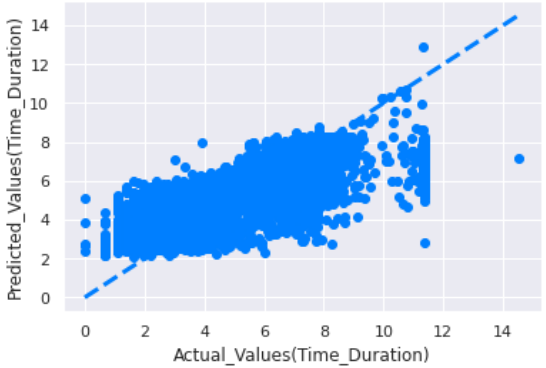
Decision Tree



In terms of accuracy measure, which is not up to par, as well as in terms of R2, which is very low and RMSE, which is quite high, the linear regression model is not performing well at all. In terms of accuracy, R2, and RMSE evaluation for all matrices, Decision Tree Regressor and Random Forest Regressor perform marginally better than Linear Regression.

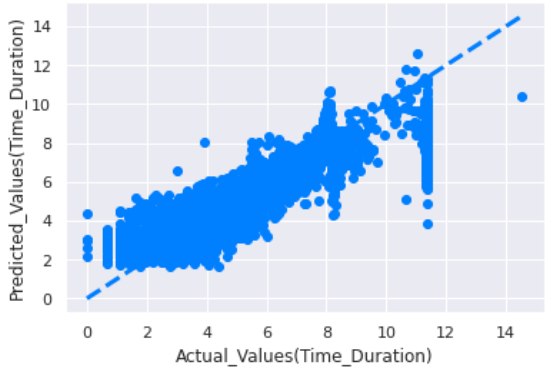
Modeling without implementing PCA

Random Forest



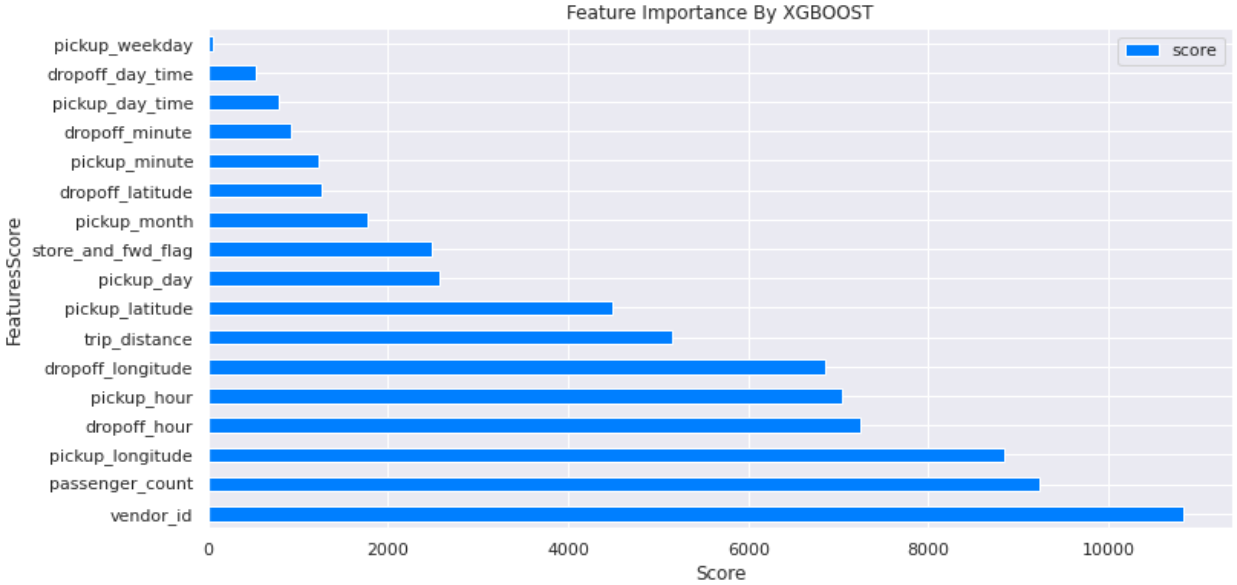
	Training_Score	Testing_Score	R2_Score	ADJ_R2	MSE	RMSE
Linear Regression	0.298175	0.274146	0.274146	0.274117	0.449004	0.670078
Decision Tree Regression	0.610109	0.601903	0.601903	0.601887	0.246258	0.496244
Random Forest Regression	0.627799	0.621592	0.621592	0.621577	0.234078	0.483816
XGBOOST	0.924322	0.907907	0.907907	0.907904	0.056967	0.238678

XGBOOST



But in terms of accuracy, R2 Score, and RMSE evaluation, Random Forest Regressor is marginally superior to Decision Tree Regressor. As we can see, XGBOOST is doing a fantastic job and offering us excellent training and testing accuracy. Additionally, the R2 score is much better than other models, and the RMSE value is also very low.

Feature Selection using Backward Elimination and XGBOOST Feature Importance:

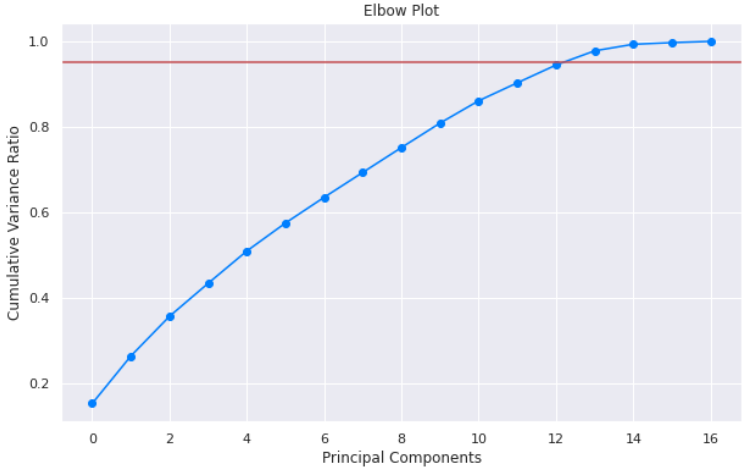


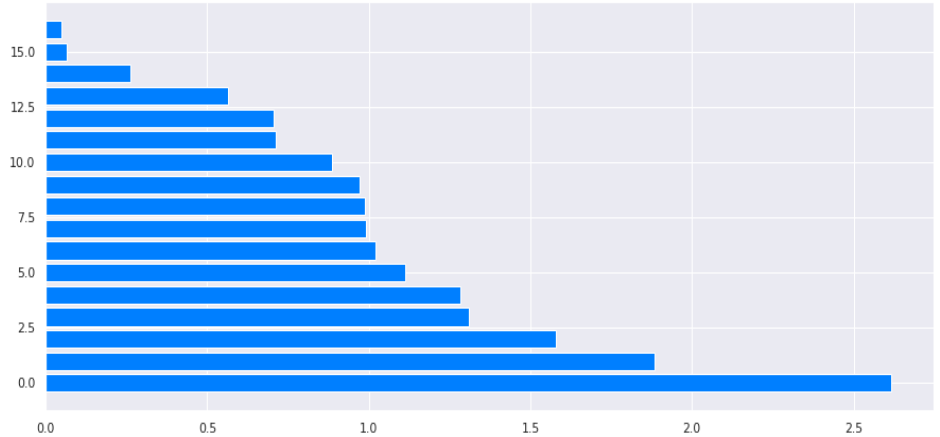
Although Backward Feature Selection did not identify any features to remove, we can remove pickup_weakday from our data set so that our model can be trained.

After selecting the features using the backward elimination method and feature importance, the impact on the results is minimal.

Implementation of PCA

Charts Showing the Important components after PCA transformation

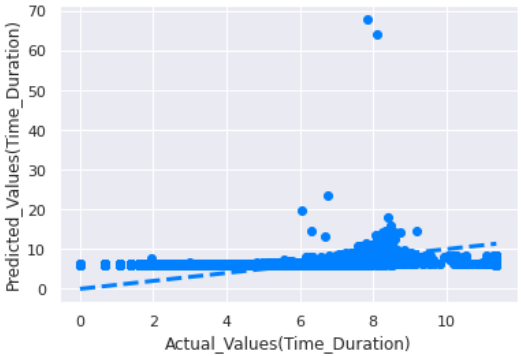




12 PCA components were found to account for more than 90% of the variance.
So we have reduced the dimensionality of our feature data from 17 to 12

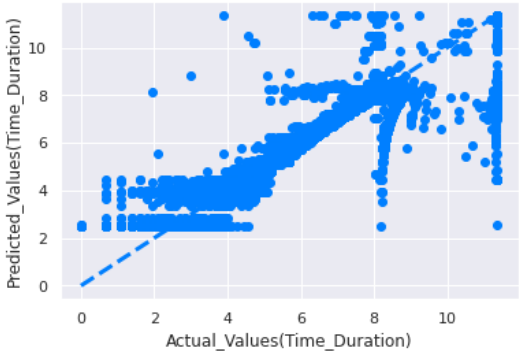
Modeling with implementing PCA

Linear Regression



	Training_Score	Testing_Score	R2_Score	ADJ_R2	MSE	RMSE
Linear Regression	0.283090	0.283905	0.283905	0.283886	0.444148	0.666444
Decision Tree Regression	0.965253	0.957100	0.957100	0.957098	0.026608	0.163121
Random Forest Regression	0.967994	0.962333	0.962333	0.962332	0.023363	0.152849
XGBOOST	0.979301	0.971078	0.971078	0.971077	0.017938	0.133934

Decision Tree

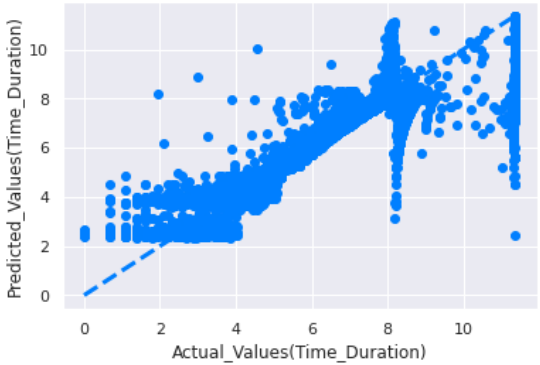


The linear regression model is completely underperforming.

After reducing the Dimension of our features, Decision Tree and Random Forest are now performing very well here, and other evaluation matrices are also producing respectable results.

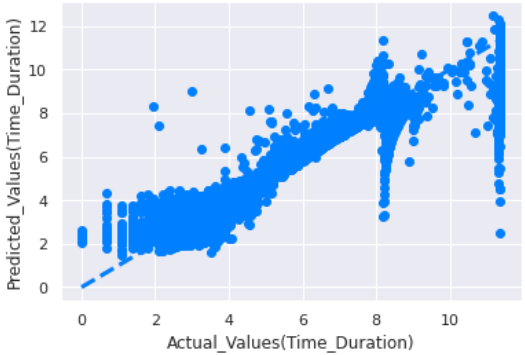
Modelling with implementing PCA

Random Forest



	Training_Score	Testing_Score	R2_Score	ADJ_R2	MSE	RMSE
Linear Regression	0.283090	0.283905	0.283905	0.283886	0.444148	0.666444
Decision Tree Regression	0.965253	0.957100	0.957100	0.957098	0.026608	0.163121
Random Forest Regression	0.967994	0.962333	0.962333	0.962332	0.023363	0.152849
XGBOOST	0.979301	0.971078	0.971078	0.971077	0.017938	0.133934

XGBOOST



But in terms of accuracy, R2 Score, and RMSE evaluation, Random Forest Regressor is marginally superior to Decision Tree Regressor.

As we can see, XGBOOST is doing a fantastic job and offering us excellent training and testing accuracy. Additionally, the R2 score is much better than other models, and the RMSE value is also very low.

Conclusions

- **We can see that we used a variety of models and methodologies, and that, with the exception of Decision Tree and Random Forest Regression, XGBOOST performed flawlessly in every case.**

- **However, after applying PCA to our features and successfully reducing the feature dimension, we discovered that Decision Tree and Random Forest were performing just as well, if not better, than before transformation.**

- **Furthermore, XGBOOST performs better after PCA transformation, and it shows marginally better results in terms of RMSE | MSE, which are lower than those of other models, as well as R2 | Adj R2 scores, which are higher than those of other models.**

- **As a result, we can use the XGBOOST Regressor to achieve high prediction rates and lower error rates. It can also be improved by using finer-tuned hyperparameters.**

THANK YOU