Capstone Project -2

NYC Taxi Trip Time Prediction

Individual Project by:

Prabin Deb Nath

Contents:

- Introduction.
- Objective.
- Dataset preview.
- Exploratory Data Analysis.
- Feature Engineering and Data Preparation.
- Model Selection
- Hyperparameter Tuning.
- Conclusions

Introduction:

- Taxi services play an important role in daily commute for people in NYC, according to wikipedia 1.6% of the overall population rely on taxis for their daily travelling.
- With the increasing use of taxis companies try to provide their services as fast as possible. These services do come at a cost as they collect data and analyse it to find the factors that affect the trip durations, which then help in predicting the same.

Objective:

- Our task is to build a model that predicts the total ride duration of taxi trips
- For which we have been provided a primary dataset which is released by the NYC Taxi and Limousine Commission, which includes pickup time, geo-coordinates, number of passengers, and several other variables.
- in New York City

Dataset Preview:

- The dataset is based on the 2016 NYC Yellow Cab trip record data made available in Big Query on Google Cloud Platform. The data was originally published by the NYC Taxi and Limousine Commission (TLC). The data was sampled and cleaned for the purposes of this project

Let's have a look at these features….

Dependent Feature:
trip_duration: duration of the trip in seconds.

Independent Features:
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

Exploratory Data Analysis:

What is Exploratory Data Analysis?

Simply defined, EDA is an approach to analyse the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations

Why is EDA important?

As said by David McCandless, “Visualizing information can give us a very quick solution to problems. We can get clarity or the answer to a simple problem very quickly.” So let's do some visualization….

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155	40.767937	-73.964630	40.765602	N	455
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.980415	40.738564	-73.999481	40.731152	N	663
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.979027	40.763939	-74.005333	40.710087	N	2124
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.010040	40.719971	-74.012268	40.706718	N	429
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.973053	40.793209	-73.972923	40.782520	N	435

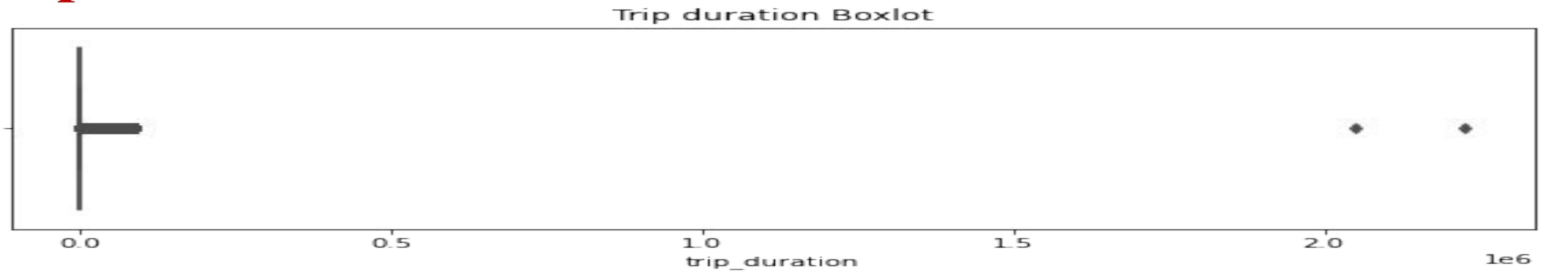
- The first 5 values of our data gives us a basic idea of what we need to work on.
- The check for null values revealed that our data doesn't contain any null values.
- The dataset contains 1458644 rows and 11 columns and none of these values are duplicates.

```
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
```

Statistics of data:

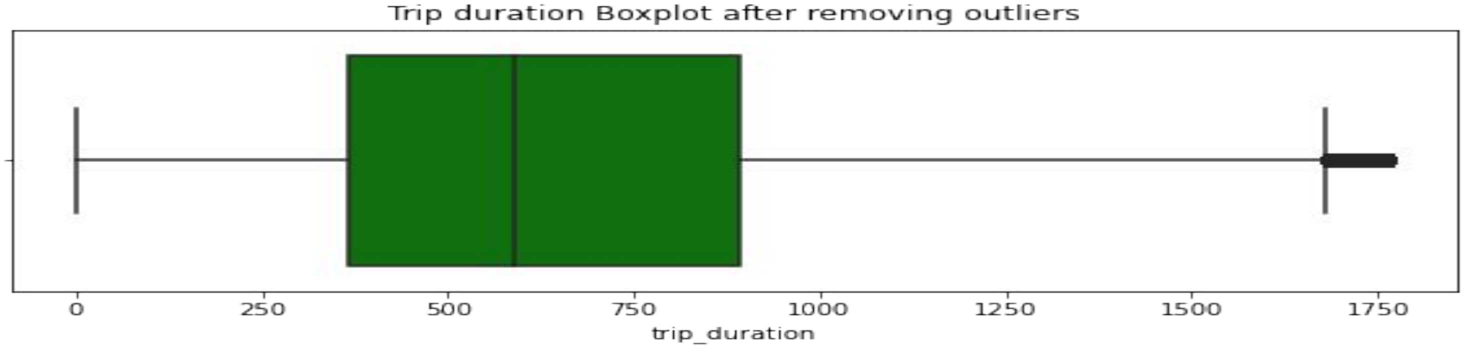
	vendor_id	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	trip_duration
count	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06	1.458644e+06
mean	1.534950e+00	1.664530e+00	-7.397349e+01	4.075092e+01	-7.397342e+01	4.075180e+01	9.594923e+02
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

Trip duration:



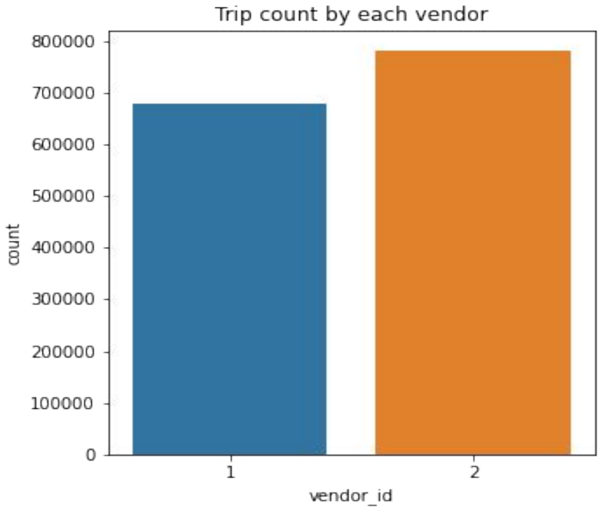
From the boxplot the we can see that there are many outliers in the data, for removing the outliers we use IQR range.

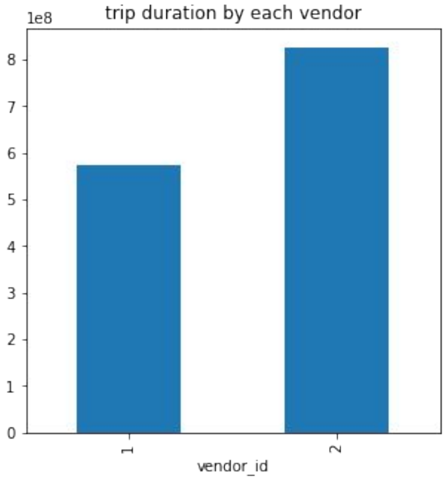
After removing the outliers this is how the boxplot looks like



Vendor_id:

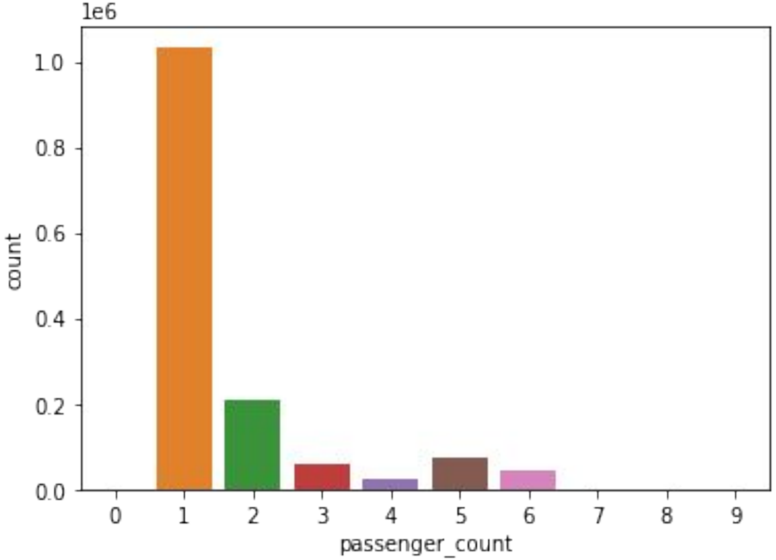
- Trip count of vendor 2 is more than that of trip count of vendor 2.
- Also, maximum duration is covered by vendor 2





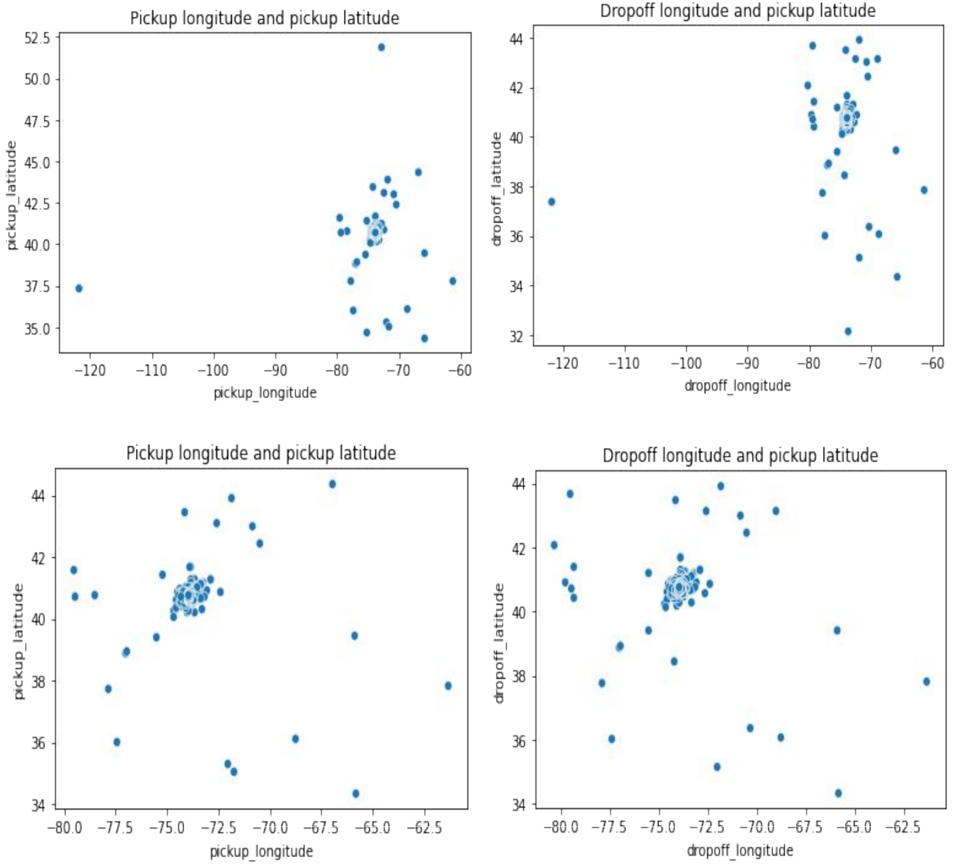
Passenger_count:

- some values are zero which mean either the trip was cancelled or there was an error in the data entry.
- 7, 8, 9 are extreme cases considering the capacity of a car, so we will get rid of them.



Pickup and drop off longitude and latitude:

- There are some outliers in the dropoff pickup longitude and latitude data.
- We have removed the outliers.
- Scatter plot of before and after removing outliers attached

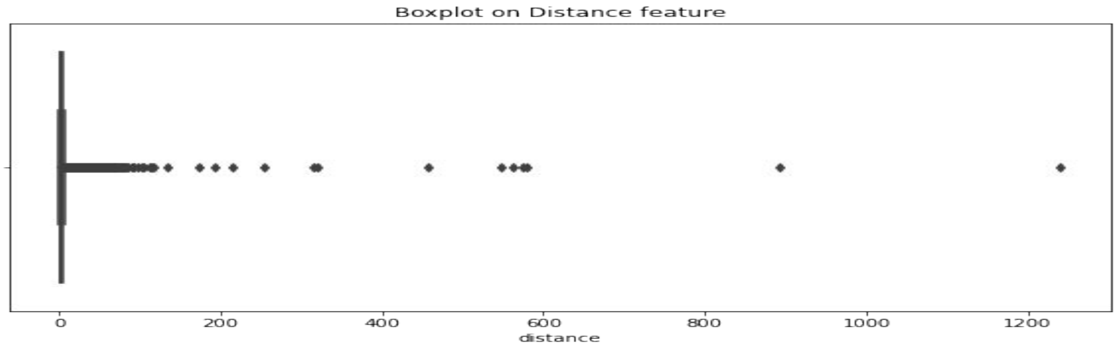


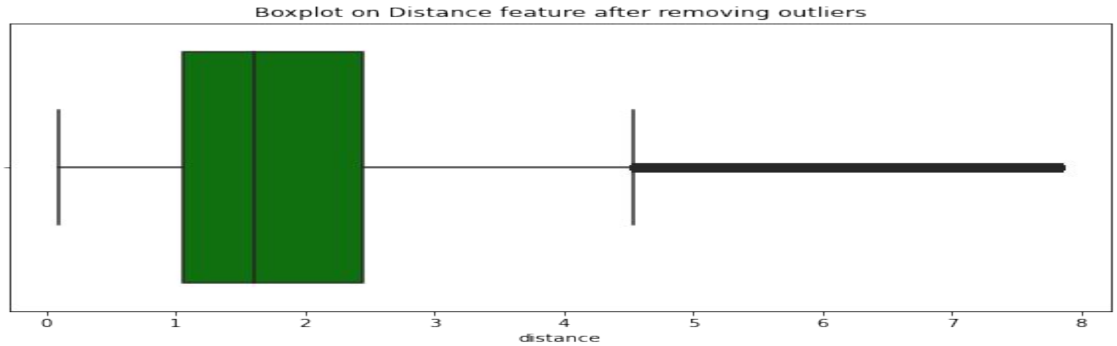
Distance:

Using pickup and drop off longitude and latitude we create the **Distance** feature.

From analysing the distance, we see that there are many outliers in the data.

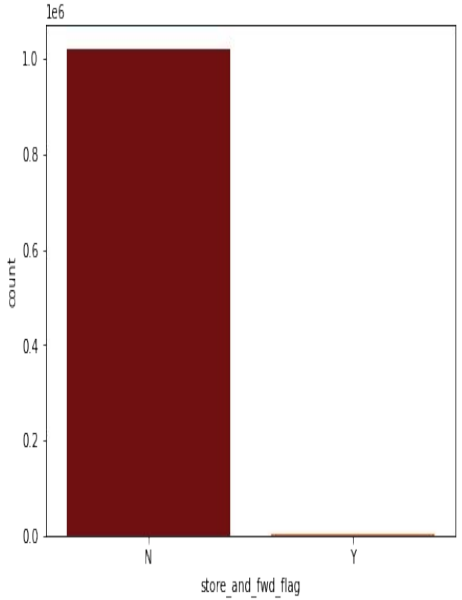
Using IQR method we remove the outliers.



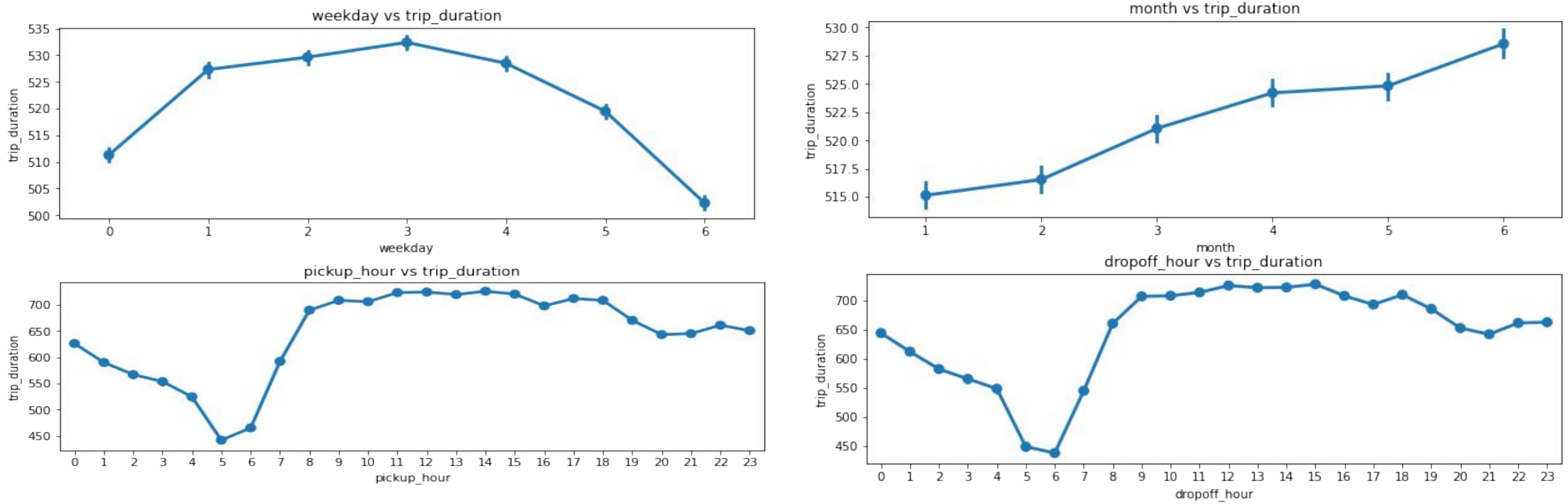


Store_and_fwd-flag:

- most of the data values are N and only few values are Y, which means most of the data was uploaded directly without storing it and forwarding.
- This is a categorical variable which will be converted to numerical using dummies.



Weekday/month/pickup_hour/drop_off_hour



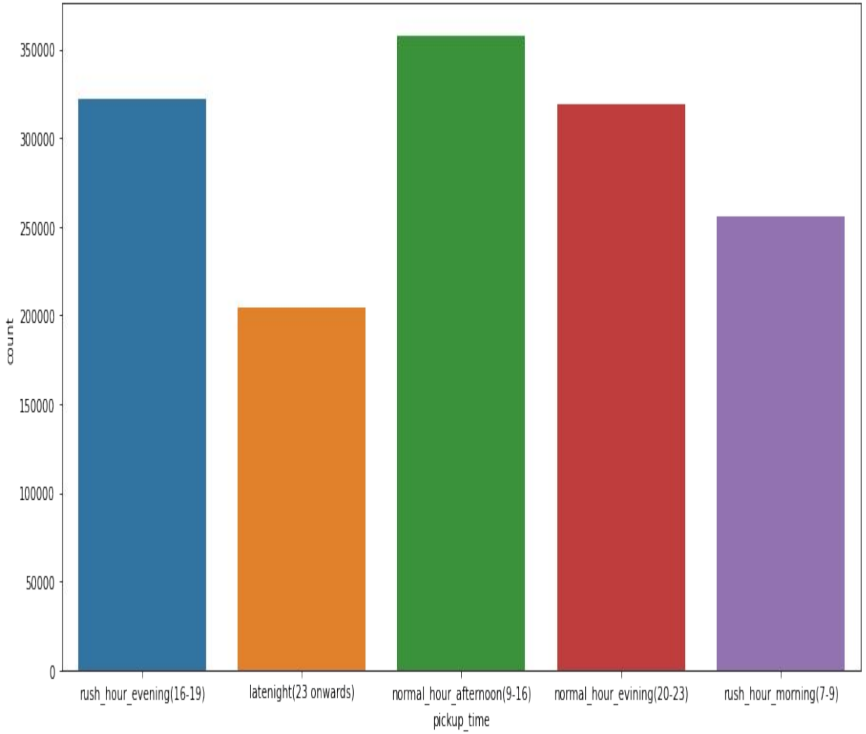
- Commute is least in early morning and late night.
- Trip duration decreases as the weekend approaches, it makes sense as most of the people either stay at home or go for vacations.
- Trip duration increases after February.

Feature Engineering and data preparation

Transform pickup hours.

Since there are 24 different values hour columns it would be better to categorize it.

between 7 - 9 = rush_hour_morning(7-10),
between 11 - 15=normal_hour_afternoon(11-15)
between 16 - 19 = rush_hour_evening(16-19),
between 20 - 23 = normal_hour evening(20-23),
between 0 - 6 = late_night(0-6),



Get dummies for all the categorical variables:

We got dummy variables for the categorical features:

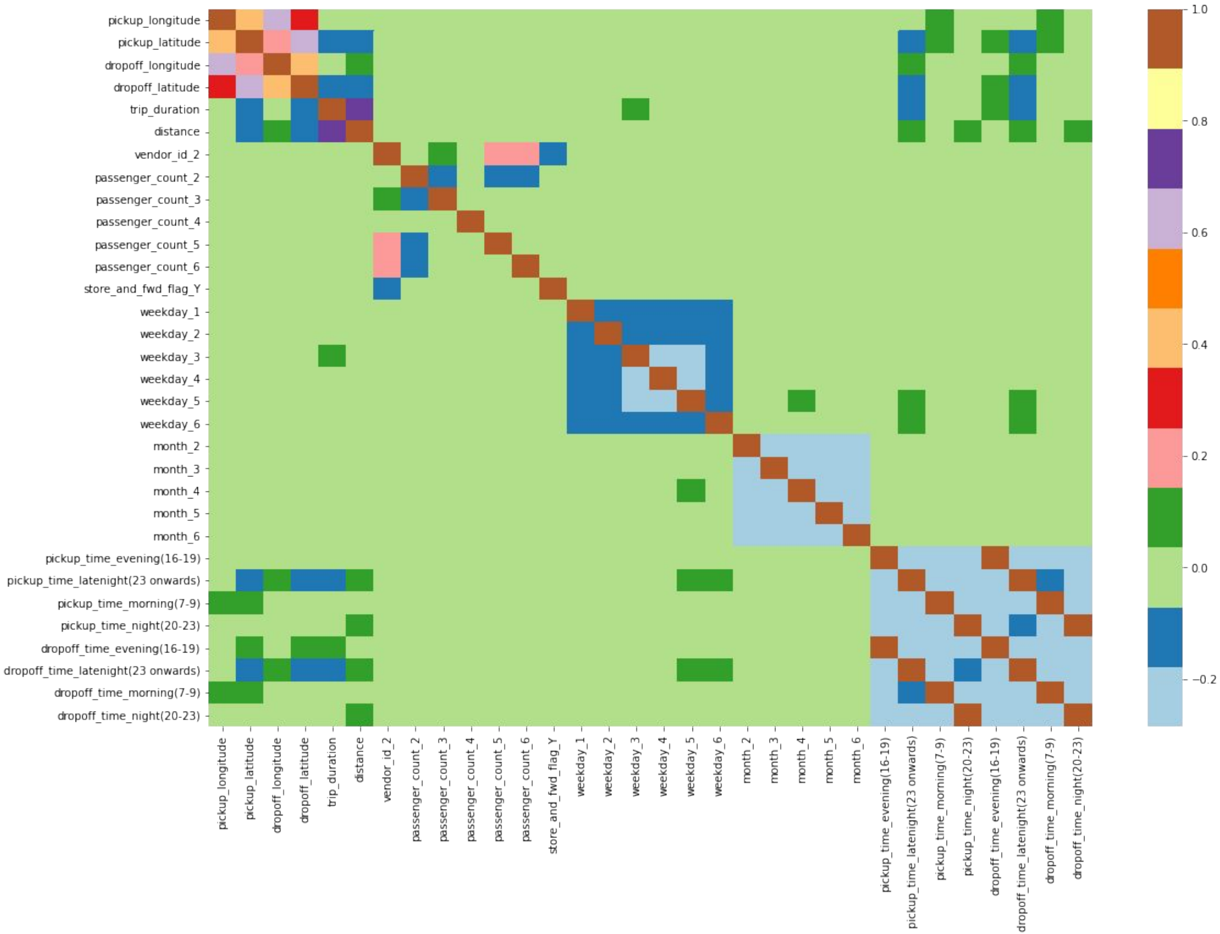
['vendor_id', 'passenger_count', ‘store_and_fwd_flag’, 'weekday', 'month', 'pickup_time’, ‘drop_off-time’]

After creating the dummy variables and dropping the un necessary columns this is transpose view of our data.

pickup_longitude	-73.982155	-73.980415	-74.010040	-73.973053	-73.982857
pickup_latitude	40.767937	40.738564	40.719971	40.793209	40.742195
dropoff_longitude	-73.964630	-73.999481	-74.012268	-73.972923	-73.992081
dropoff_latitude	40.765602	40.731152	40.706718	40.782520	40.749184
trip_duration	455.000000	663.000000	429.000000	435.000000	443.000000
distance	1.498523	1.805510	1.485500	1.188590	1.098944
vendor_id_2	1.000000	0.000000	1.000000	1.000000	1.000000
passenger_count_2	0.000000	0.000000	0.000000	0.000000	0.000000
passenger_count_3	0.000000	0.000000	0.000000	0.000000	0.000000
passenger_count_4	0.000000	0.000000	0.000000	0.000000	0.000000
passenger_count_5	0.000000	0.000000	0.000000	0.000000	0.000000
passenger_count_6	0.000000	0.000000	0.000000	0.000000	1.000000
store_and_fwd_flag_Y	0.000000	0.000000	0.000000	0.000000	0.000000
weekday_1	0.000000	0.000000	0.000000	0.000000	0.000000
weekday_2	0.000000	0.000000	1.000000	0.000000	0.000000
weekday_3	0.000000	0.000000	0.000000	0.000000	0.000000
weekday_4	0.000000	0.000000	0.000000	0.000000	0.000000
weekday_5	0.000000	0.000000	0.000000	1.000000	1.000000
weekday_6	0.000000	1.000000	0.000000	0.000000	0.000000
month_2	0.000000	0.000000	0.000000	0.000000	0.000000
month_3	1.000000	0.000000	0.000000	1.000000	0.000000
month_4	0.000000	0.000000	1.000000	0.000000	0.000000
month_5	0.000000	0.000000	0.000000	0.000000	0.000000
month_6	0.000000	1.000000	0.000000	0.000000	0.000000
pickup_time_evening(16-19)	1.000000	0.000000	1.000000	0.000000	0.000000
pickup_time_latenight(23 onwards)	0.000000	1.000000	0.000000	0.000000	0.000000
pickup_time_morning(7-9)	0.000000	0.000000	0.000000	0.000000	0.000000
pickup_time_night(20-23)	0.000000	0.000000	0.000000	0.000000	1.000000
dropoff_time_evening(16-19)	1.000000	0.000000	1.000000	0.000000	0.000000
dropoff_time_latenight(23 onwards)	0.000000	1.000000	0.000000	0.000000	0.000000
dropoff_time_morning(7-9)	0.000000	0.000000	0.000000	0.000000	0.000000
dropoff_time_night(20-23)	0.000000	0.000000	0.000000	0.000000	1.000000

Multicollinearity in features:

As we can see many of the features are correlated to each other, and among these the highly correlated groups are: The **pickup/ dropoff longitude/ latitude data, pickup/ dropoff hours**. Most of the days, months and time categories are negatively correlated to each other, the negative correlation between these features make sense as when one increases the other will decrease.



Data preparation:

We need to prepare the data before we put them through regression models.
We separate the dependent and the independent variables, where y is the dependent variable trip_duration_minutes and X contains rest of the features in our dataset.
Now do the train test split to separate the training and the testing data that we will use to build and validate our regression models. (80% training and 20% test data.)
Finally we will transform our data using Standard Scaler, this is done to standardize the data before feeding them to the models.

Model Selection:

We have prepared our data now its time to select the best performing model:

<u>The models we will be using are:</u>

- **Linear Regression**
- **Decision Tree Regressor**
- **XG Boost Regressor**
- **Hist Gradient Boosting Regressor**
- **Ada Boost Regressor**

We will compare these models and select the best performing model for the prediction.

	name	train_time	Train_R2_Score	Test_R2_Score	Test_RMSE_Score
0	Linear Regression	1.463104	0.542098	0.540535	269.141373
1	Decision Tree Regressor	24.522635	1.000000	0.395240	308.777470
2	XG Boost Regressor	124.418667	0.611064	0.608651	248.391290
3	Hist Gradient Boosting Regressor	20.812032	0.671445	0.668316	228.673741
4	AdaBoostRegressor	85.663039	0.374850	0.372462	314.538611

As we can see from the above table **Hist Gradient Boosting Regressor** performs the best.

Hence we will proceed with it and try some hyperparameter tuning and cross validations to improve the score and reduce the error.

Hyperparameter Tuning:

The parameters we will be tuning to get better performances are: max_depth, learning_rate, min_samples_leaf and max_iter.

5 Cross validations for each set will be conducted in order to find the best parameters.

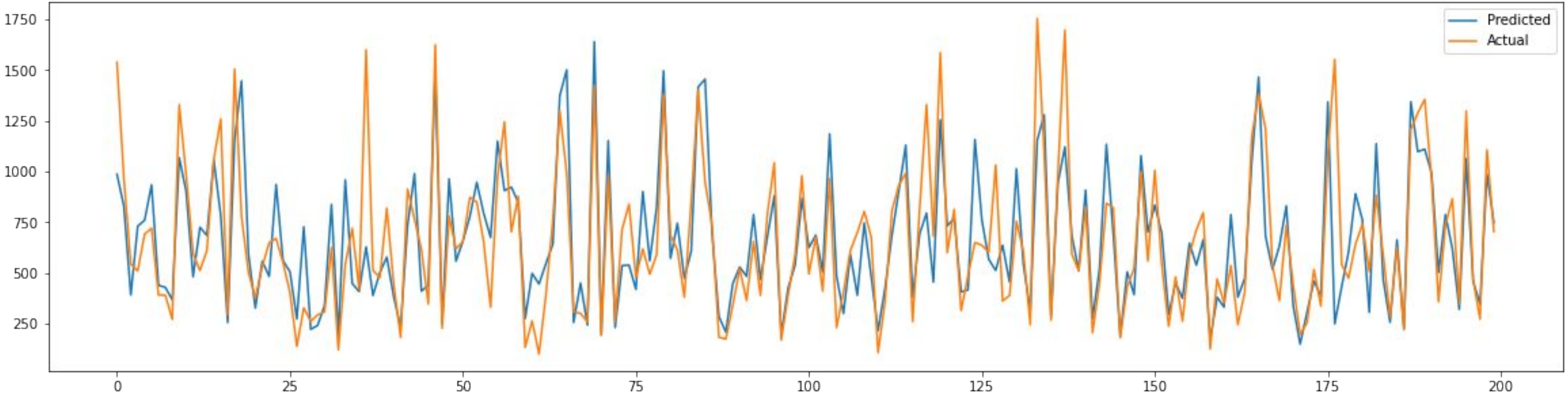
```
Best parameters are
 {'min_samples_leaf': 50, 'max_iter': 500, 'max_depth': 8, 'learning_rate': 0.4}

=========================================================

Train_R2_Score  0.7372568000095774
Test_R2_Score  0.717591763647963
Train_RMSE_Score   203.88610219621287
Test_RMSE_Score   211.00499291078603

=========================================================
```

Actual vs Predicted Values:



Since the data is very large we will consider the first 200 values to compare the actual and predicted trip_durations.

As we can see the model has done a pretty good job in predicting the durations. Hence it would be safe to say that Hist Gradient Boosting Regressor can be used for future predictions.

Conclusions:

1. Distance calculated using the haversine function plays an important role in predicting the trip durations.

2. The best algorithm in this case is Hist Gradient Boosting Regressor.

3. The untuned model was able to explain 67% of the variance on the test set, while the tuned model explained 71% of variance on the test set which is a good improvement.

4. The Test RMSE on test set by the Hist Gradient Boosting Regressor was 211.00499291078603

5. Hence, Boosting algorithms are by far the best while dealing with large datasets with most of thee features have very little correlation with the dependent feature.

Challenges Faced:

- Huge amount of data had to be dealt with keeping in mind not to loose anything of value.
- Data contained many outliers which had to be removed.
- Data being huge was very time consuming.
- Optimizing the model was very difficult.

Thank you.