Capstone Project-2

NYC Taxi Trip Time Prediction

Lavanya M

Contents:

1. Taxi in NYC
2. About dataset
3. EDA
4. Feature engineering
5. Models used
6. Model comparison
7. Conclusion

NYC taxi trip duration

New York City taxi rides form the core of the traffic in the city of New York. The many rides taken every day by New Yorkers in the busy city can give us a great idea of traffic times, road blockages, and so on.

Predicting the duration of a taxi trip is very important since a user would always like to know precisely how much time it would require to travel from one place to another. And plan trips accordingly.

Problem statement

Task is to build a model that predicts the total ride duration of taxi trips in New York City. The dataset is one released by the NYC Taxi and Limousine Commission, which includes pickup time, geo-coordinates, number of passengers, and several other variables.

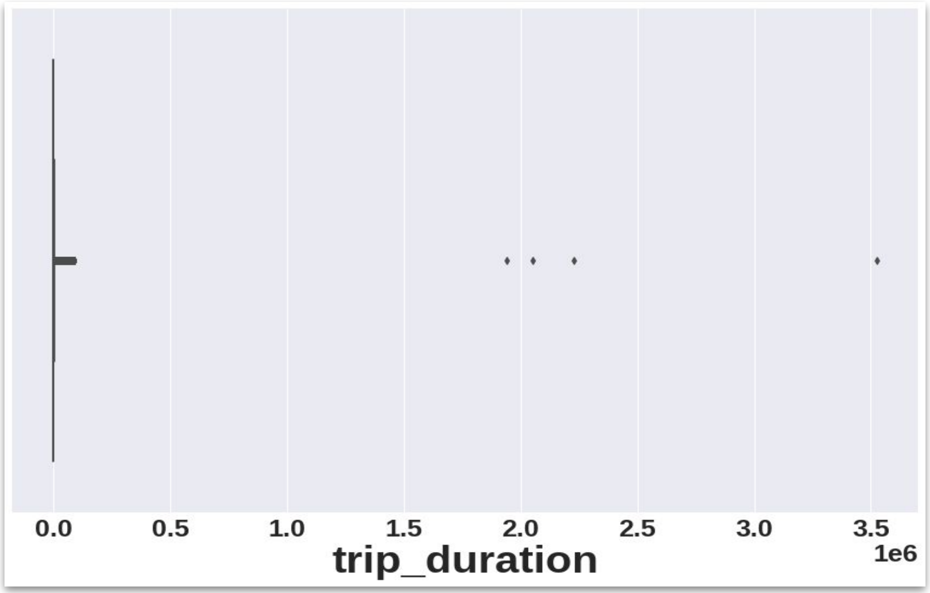
About data

- 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.
- Dataset has 1458644 rows and 11 columns.
- Data has no null values.

Columns in the dataset

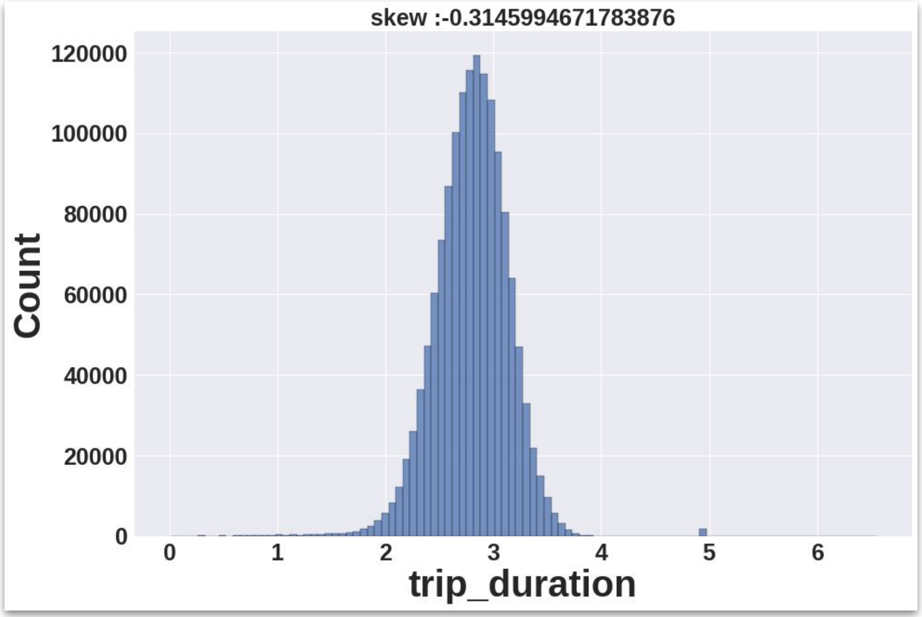
- **id** - a unique identifier for each trip.
- **vendor_id** - a code indicating the provider associated with the trip record.
- **pickup_datetime** - date and time when the meter was engaged.
- **dropoff_datetime** - date and time when the meter was disengaged.
- **passenger_count** - the number of 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_duration** - The total duration of the trip in seconds. This feature is the **target value**.

Exploratory Data Analysis

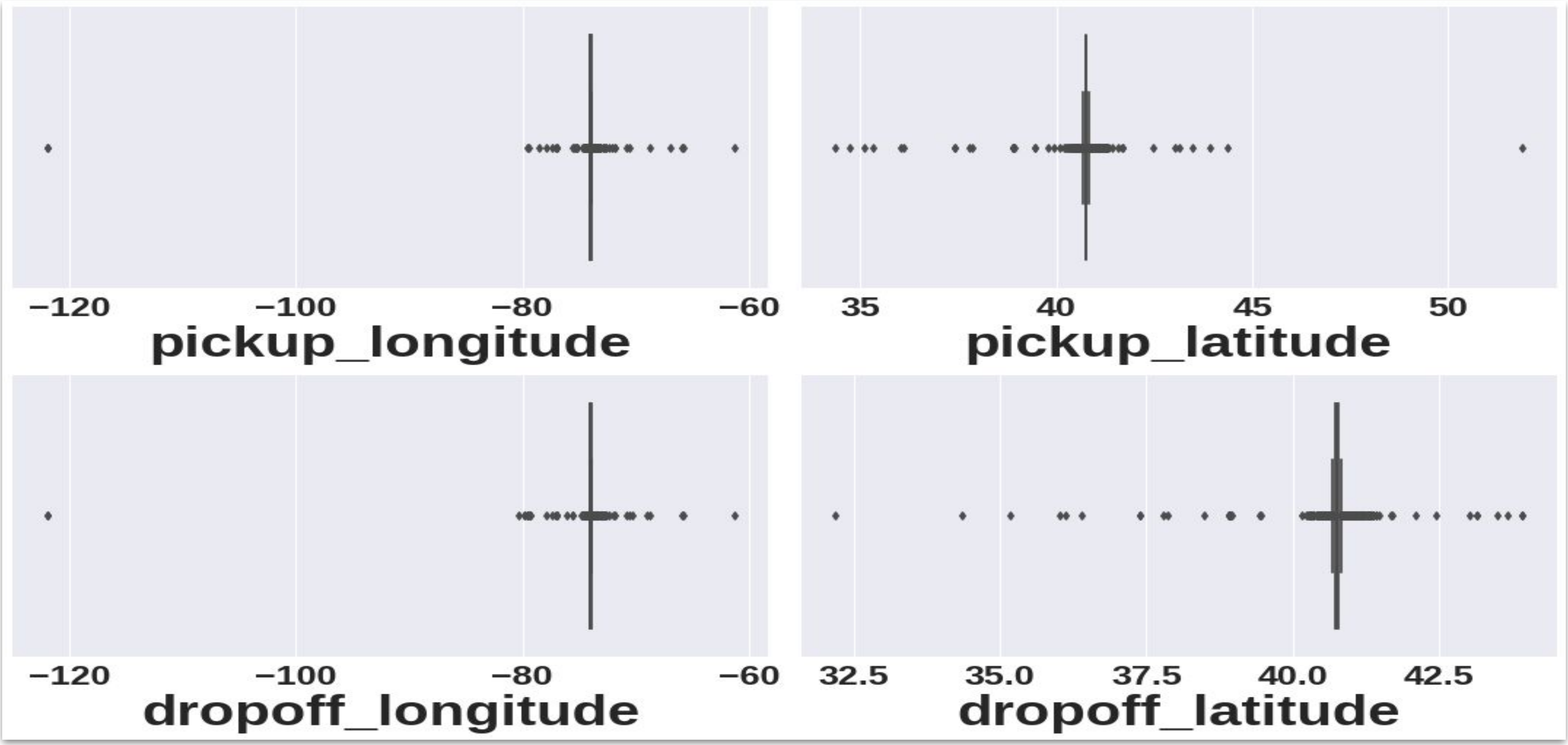


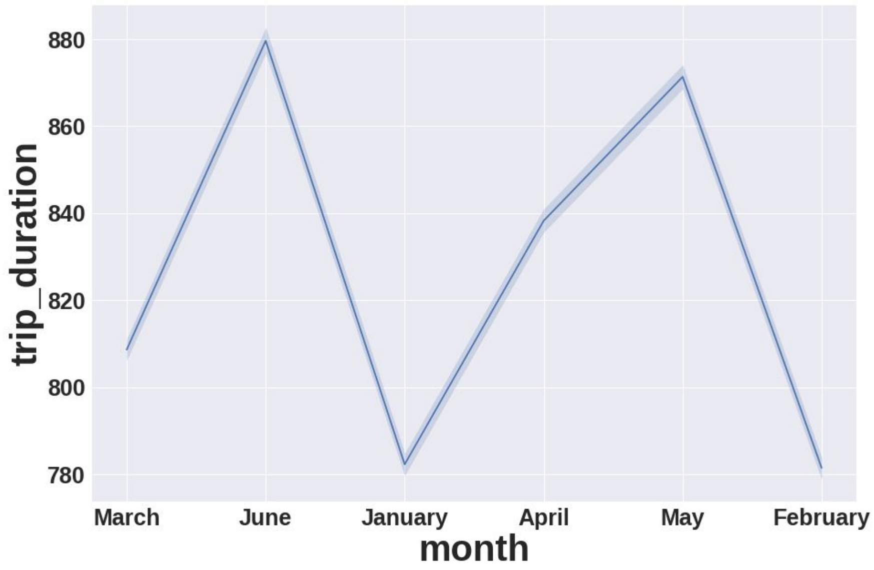
* Min values is 1 second and max is 352682 seconds (ie. 4 days)

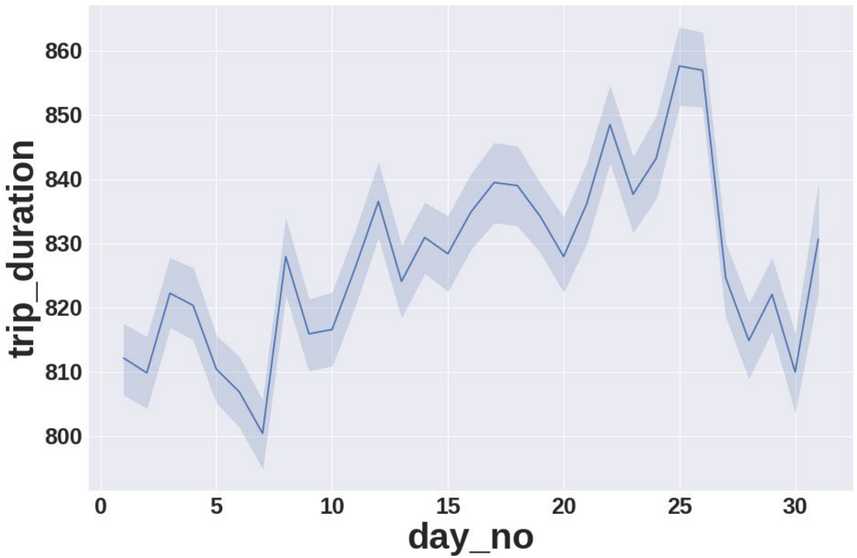
* Using 2 standard deviation after taking log10 of trip duration.

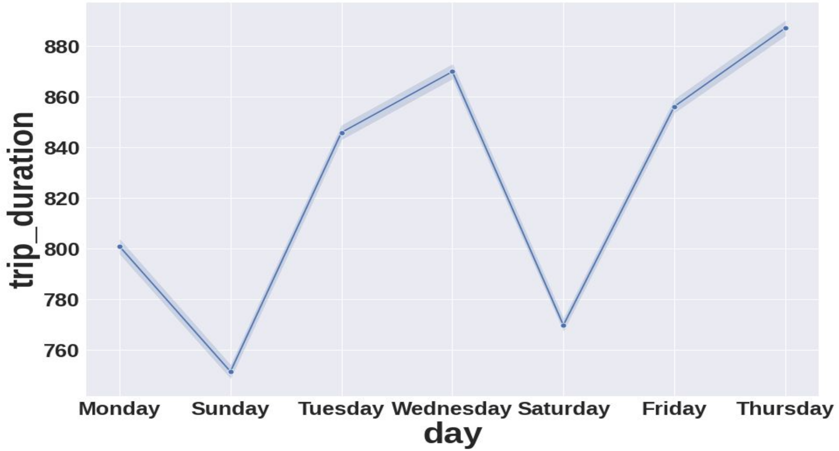


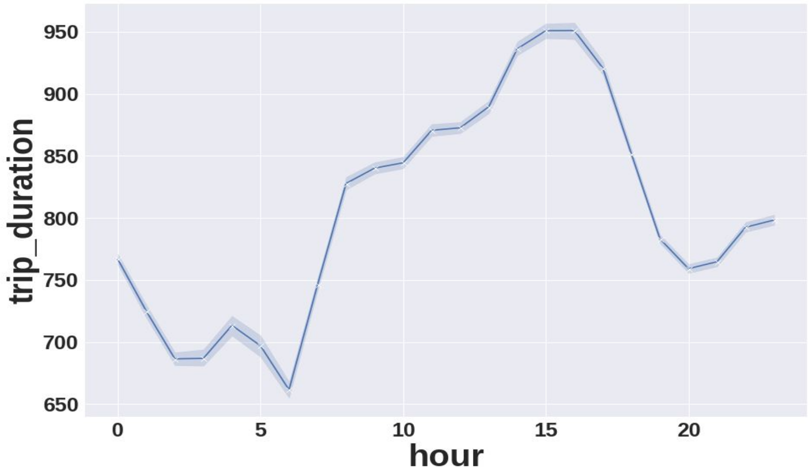
NYC city borders: city_long_border = (-74.03, -73.75) and city_lat_border = (40.63, 40.85)

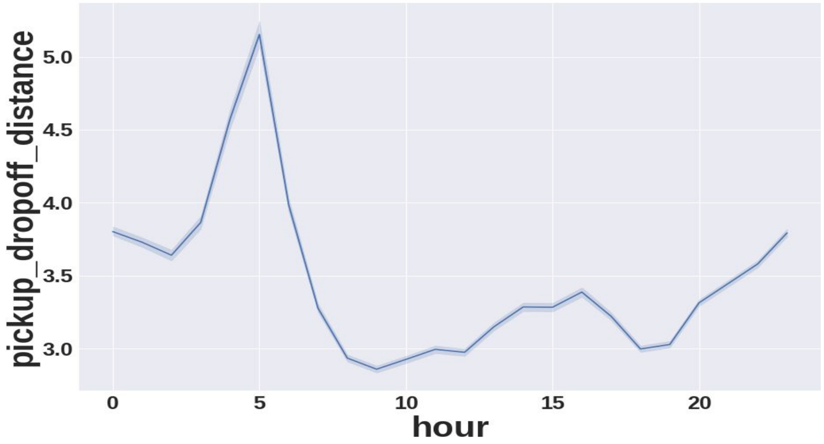


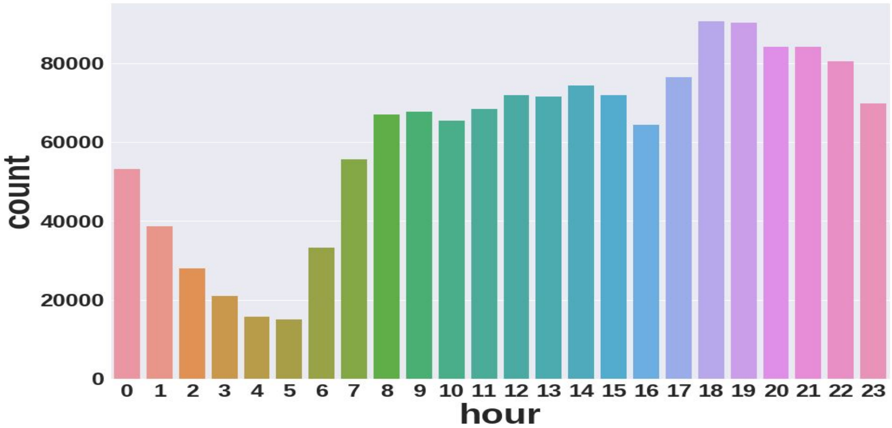

trip_duration
880
860
840
820
800
780
March
June
January
April
May
February
month

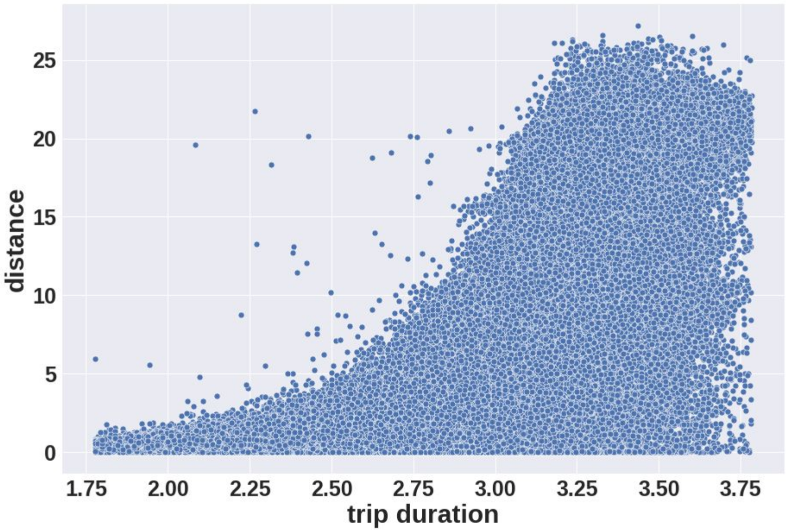

trip_duration
860
850
840
830
820
810
800
0
5
10
15
20
25
30
day_no

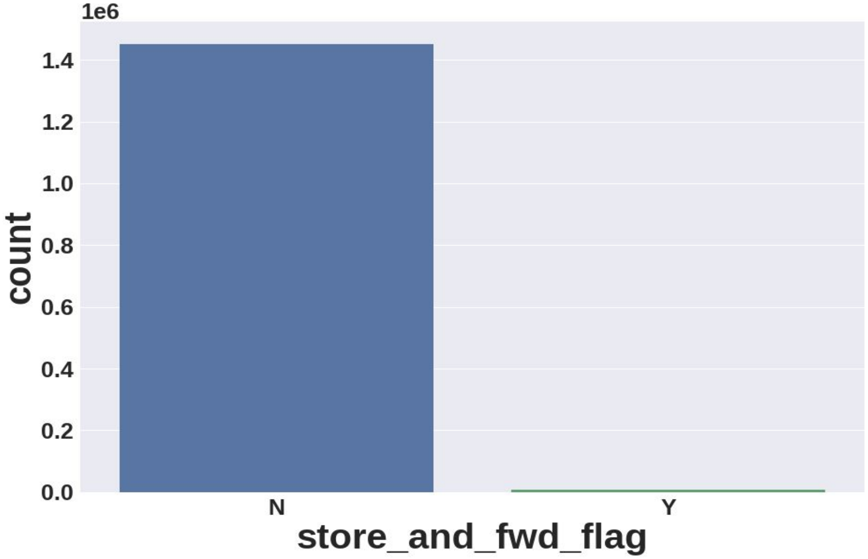

trip_duration
880
860
840
820
800
780
760
Monday
Sunday
Tuesday
Wednesday
Saturday
Friday
Thursday
day


trip_duration
950
900
850
800
750
700
650
0
5
10
15
20
hour


pickup_dropoff_distance
3.0
3.5
4.0
4.5
5.0
0
5
10
15
20
hour


count
0
20000
40000
60000
80000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
hour


distance
0
5
10
15
20
25
1.75
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
trip duration


1e6
count
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
N
Y
store_and_fwd_flag

Feature engineering

Created columns:

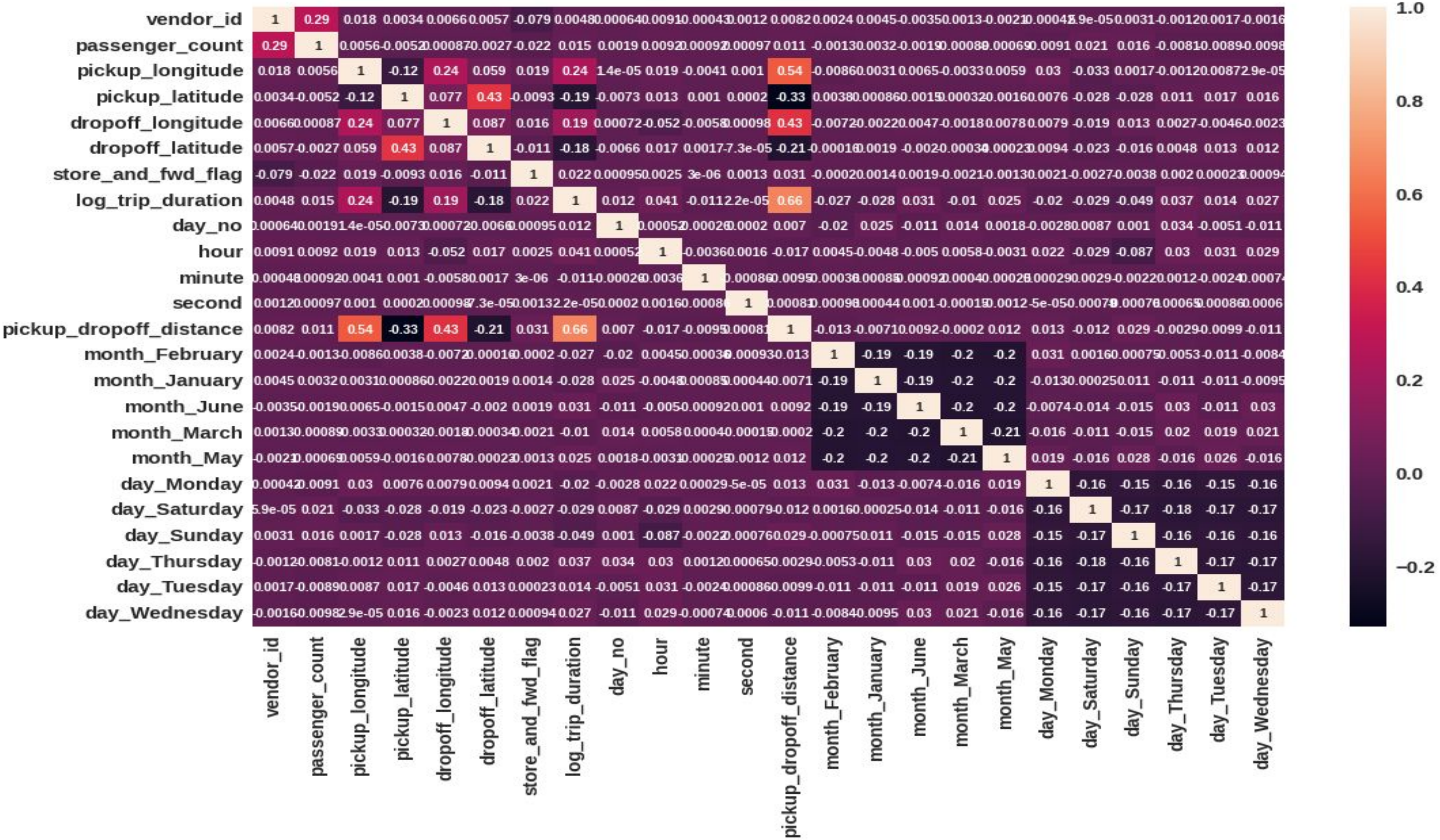
- **Day no, day , month, hour, minute and second** from pickip_date time column.
- **Pickup_dropoff_distance** by calculating geo distance between pickup_longitude, pickup_latitude, dropoff_longitude and latitude columns.

Transformed / scaled columns:

- **Month and day** columns one hot encoded.
- **store_and_fwd_flag** values Y and N mapped to 0 and 1
- **pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude** and **day_no were mix max scaled.**

Dropped columns:

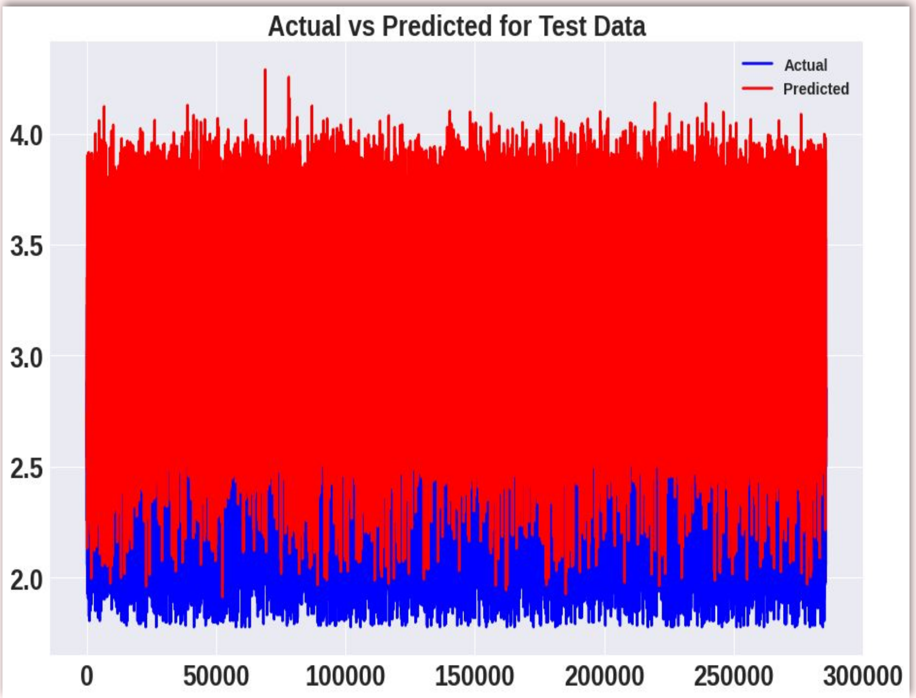
- **ID, Pickup_datetime and dropoff_datetime**



Models

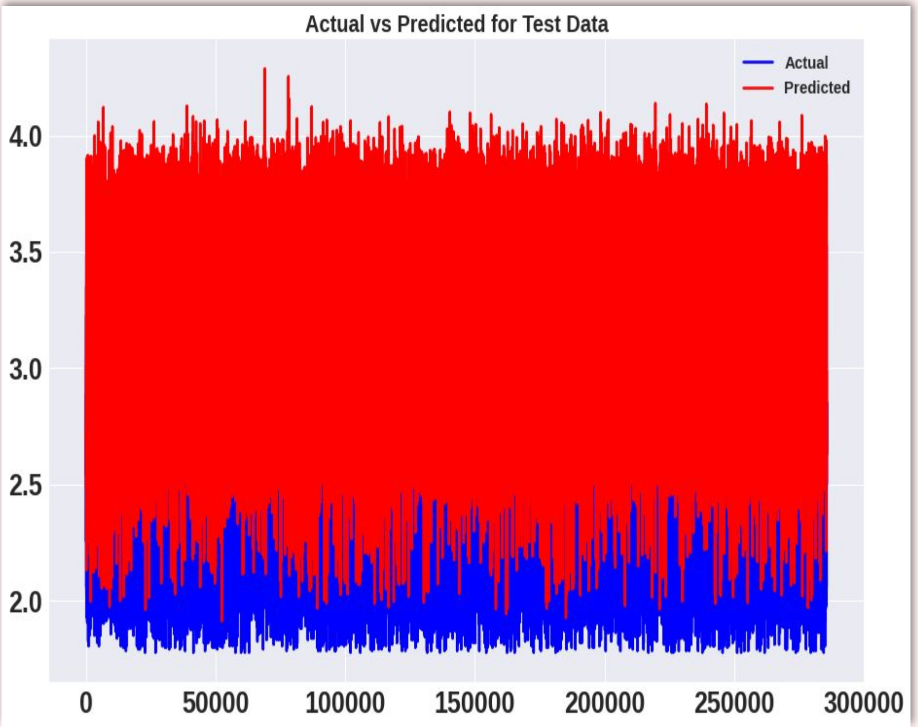
Linear Regression :

Linear regression model finds the set of θ coefficients that minimize the sum of squared errors.



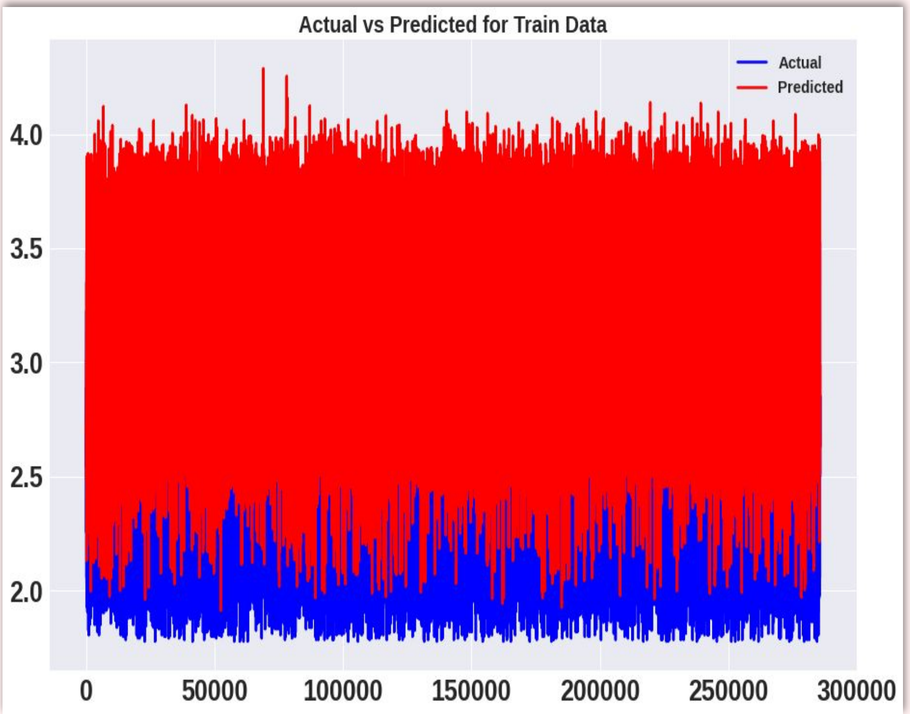
Lasso Regression :

The lasso method was used to shrink coefficients. Lasso was run using a range of values for the penalizing parameter, λ . Grid Search was used to find the lasso model with the lowest error and select the value of λ to use.



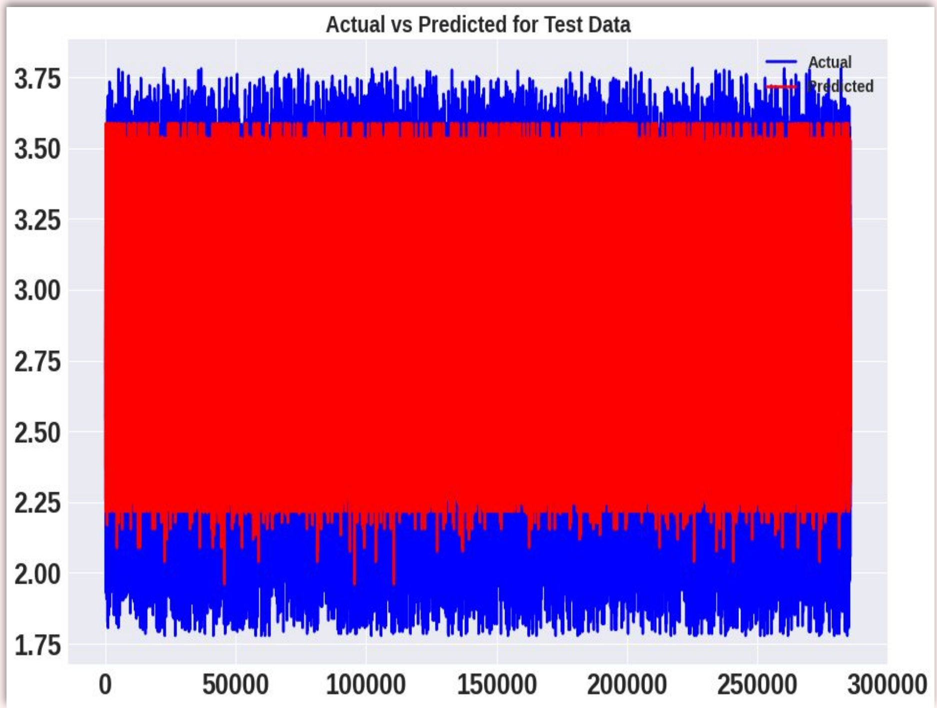
Ridge Regression :

To further confirm the best set of covariates to use, the regression method was used. It performs L2 regularization, i.e. adds penalty equivalent to square of the magnitude of coefficients.



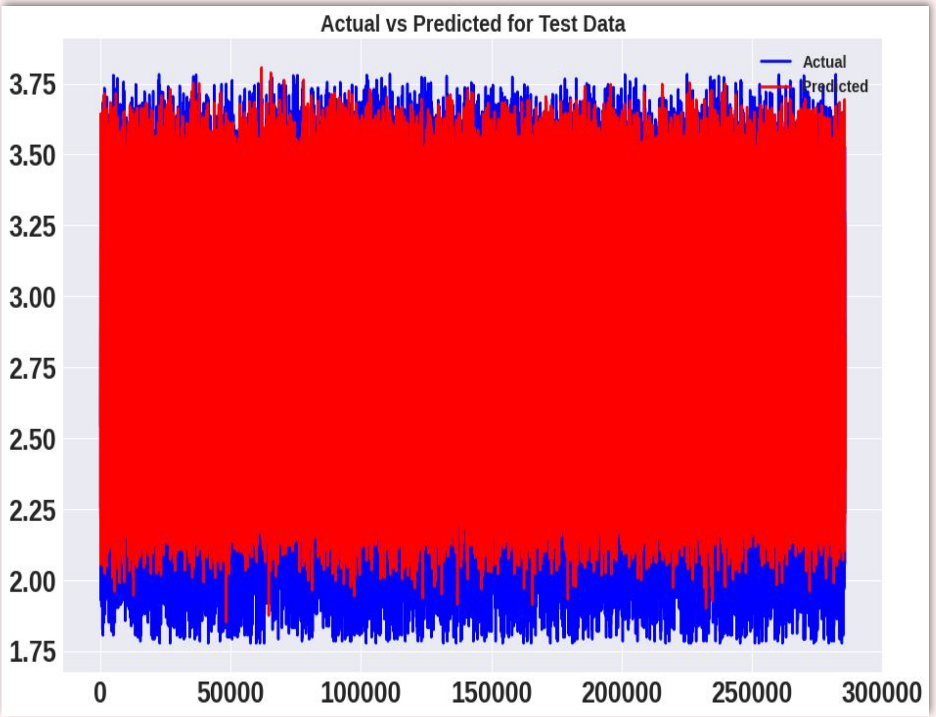
Decision Tree :

The decision trees was also built on the training data in order to improve prediction accuracy .We used GridSearch to tune the hyperparameters of Decision Tree to get the best possible test score.



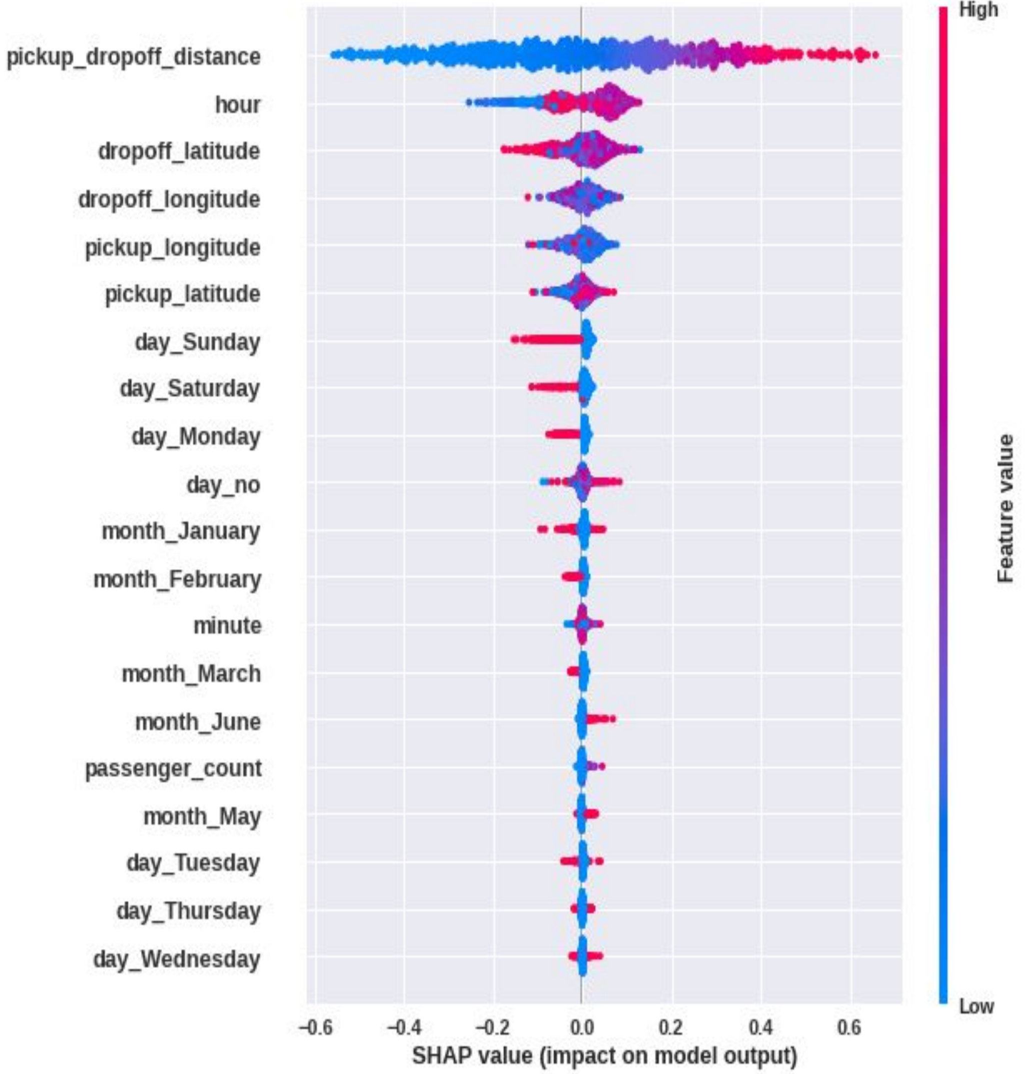
XGBoost :

is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. It performs well on large datasets. Another aspect of XGBoost is that it keeps a nice check between bias and variance which helps in better prediction.



Model comparison

Algorithm	MSE	RMSE	R2	ADJUSTED R2
Linear regression	0.05	0.22	0.48	0.48
Lasso	0.05	0.22	0.48	0.48
Ridge	0.05	0.22	0.48	0.48
Decision tree	0.02	0.16	0.71	0.71
XGboost	**0.01**	**0.13**	**0.82**	**0.80**



Feature importance

Conclusion

- Most passengers travel alone.
- Only few records were recorded in memory before sharing(Y).
- Trip duration has Min values is 1 second and max is 352682 seconds (ie. 4 days)
- Trip distance per hour is highest during early morning hours which can account for some things such as outstation trips taken during the weekends. Also because of longer trips towards the city airport which is located in the outskirts of the city.
- Trip duration is least in january and highest in june
- Trip duration on an average lasts for 10 minutes.
- Most distance is travelled at around 5 am and least at around 9 am.
- Most passengers prefer to travel on weekdays instead of week ends leading to high trip duration times.
- XGboost algorithm best fits our data.

Thank you