Capstone Project - 2

NYC Taxi Trip Duration Prediction

Team Members

Akanksha Agarwal
Ganeshkumar Patel

Problem Statement

Task is to predict total ride duration of taxi trips in New York City.

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.

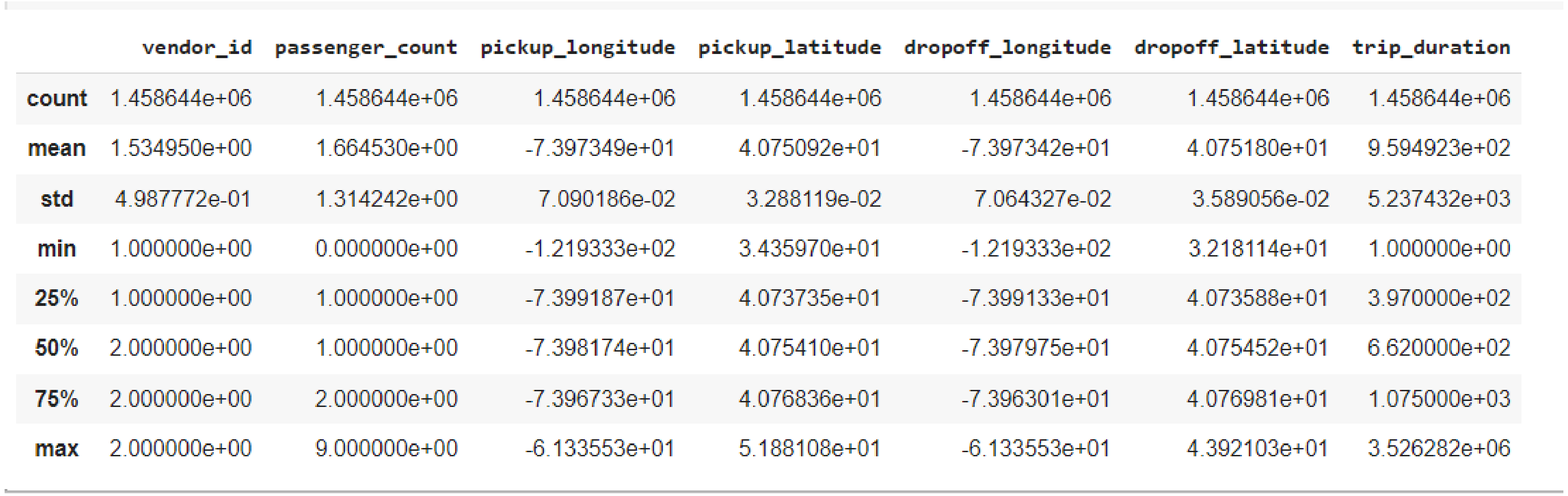
Target Feature :

trip_duration : duration of the trip in seconds.

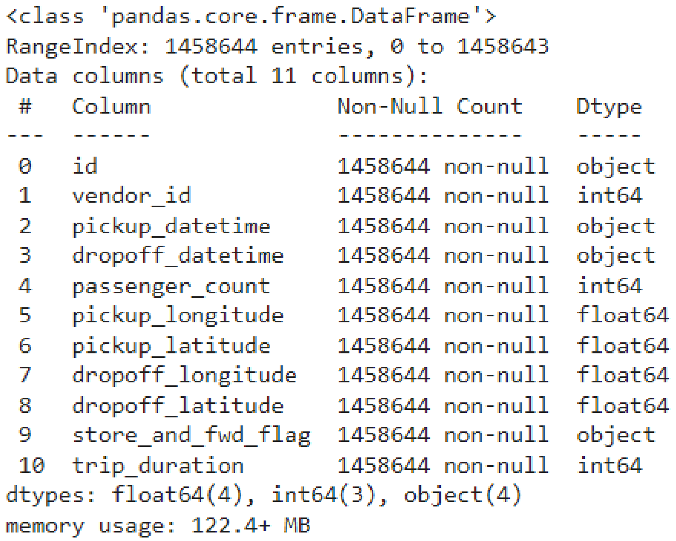
Approach towards Solution

1. Defining Problem Statement
2. Data Preparation
   - 2.1 Data Exploration
   - 2.2 Data Processing
   - 2.3 Feature Engineering
   - 2.4 EDA
3. Preparing Dataset For Modeling
   - 3.1 Feature Selection
   - 3.2 Categorical Feature Encoding
   - 3.3 Applying Model
4. Model Metrics Evaluation
5. Model explainabilty
6. Conclusion

Dataset Exploration



	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

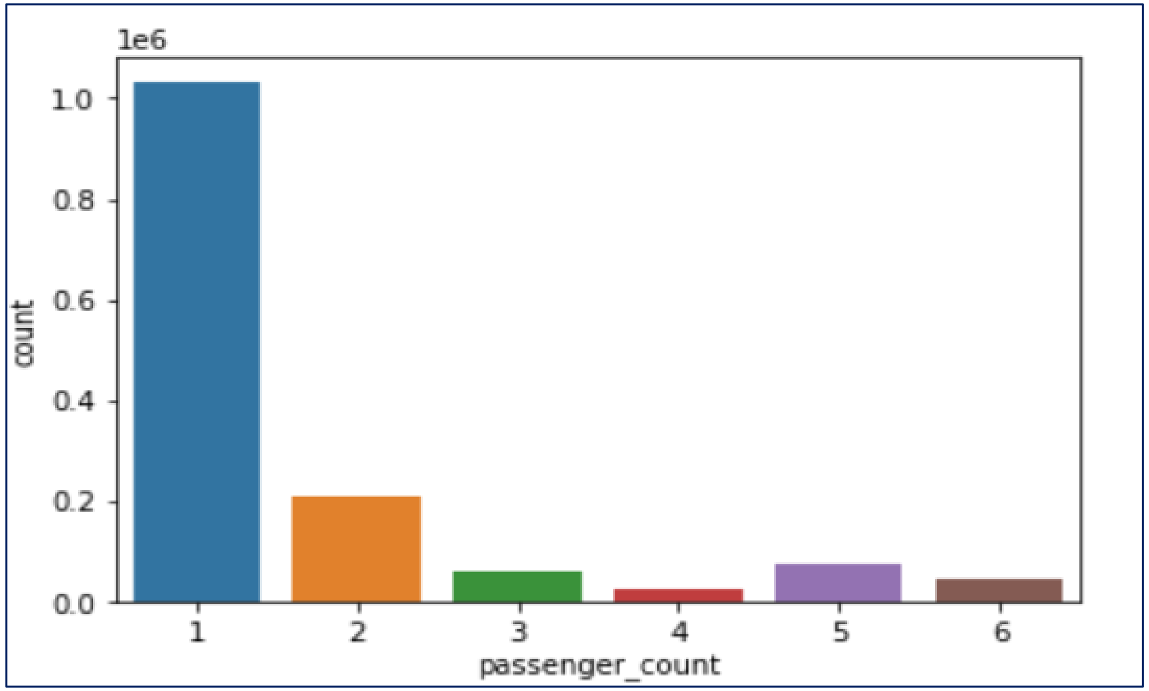


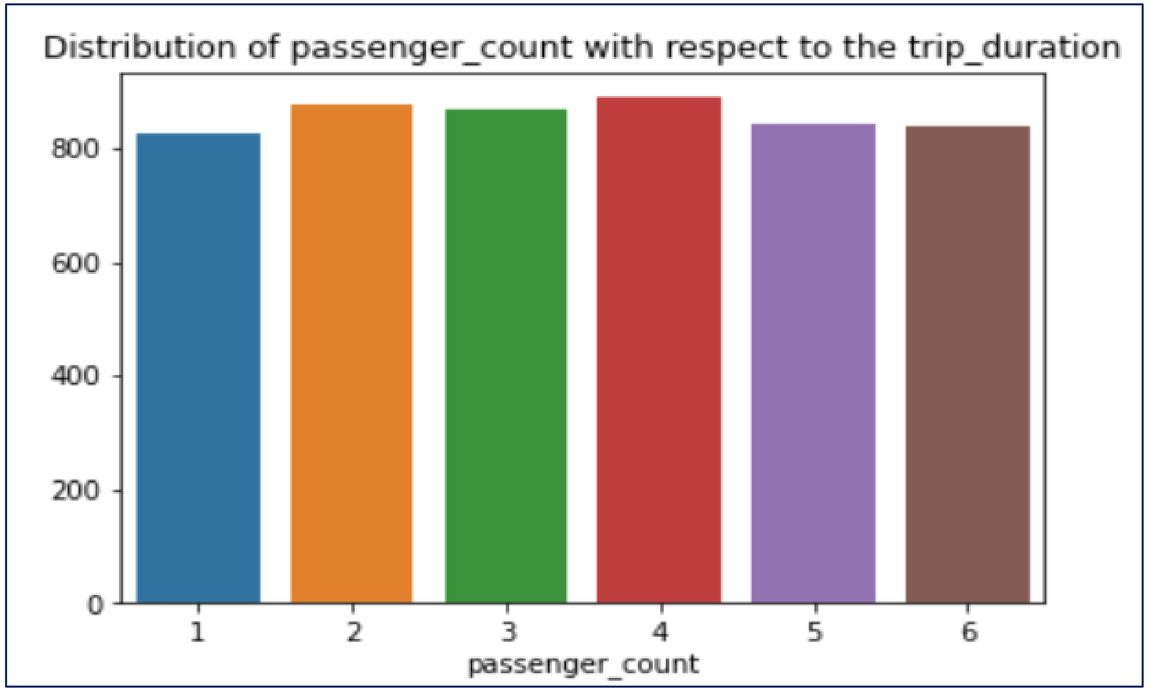
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1458644 entries, 0 to 1458643
Data columns (total 11 columns):
 #   Column              Non-Null Count    Dtype
---  ------              --------------    -----
 0   id                  1458644 non-null  object
 1   vendor_id           1458644 non-null  int64
 2   pickup_datetime     1458644 non-null  object
 3   dropoff_datetime    1458644 non-null  object
 4   passenger_count     1458644 non-null  int64
 5   pickup_longitude    1458644 non-null  float64
 6   pickup_latitude     1458644 non-null  float64
 7   dropoff_longitude   1458644 non-null  float64
 8   dropoff_latitude    1458644 non-null  float64
 9   store_and_fwd_flag  1458644 non-null  object
 10  trip_duration       1458644 non-null  int64
dtypes: float64(4), int64(3), object(4)
memory usage: 122.4+ MB
```

1. There are 1458644 observations and 11 available features
2. There are no null values and duplicate data.
3. Datetime column is of object datatype
4. From above table we can infer that trip duration has max value of 3526282 seconds i.e almost 979.5 hours and minimum 1 second. Thus, it seems to have outliers. Also the number of passenger counts in upto 9, which may not be the case

EDA

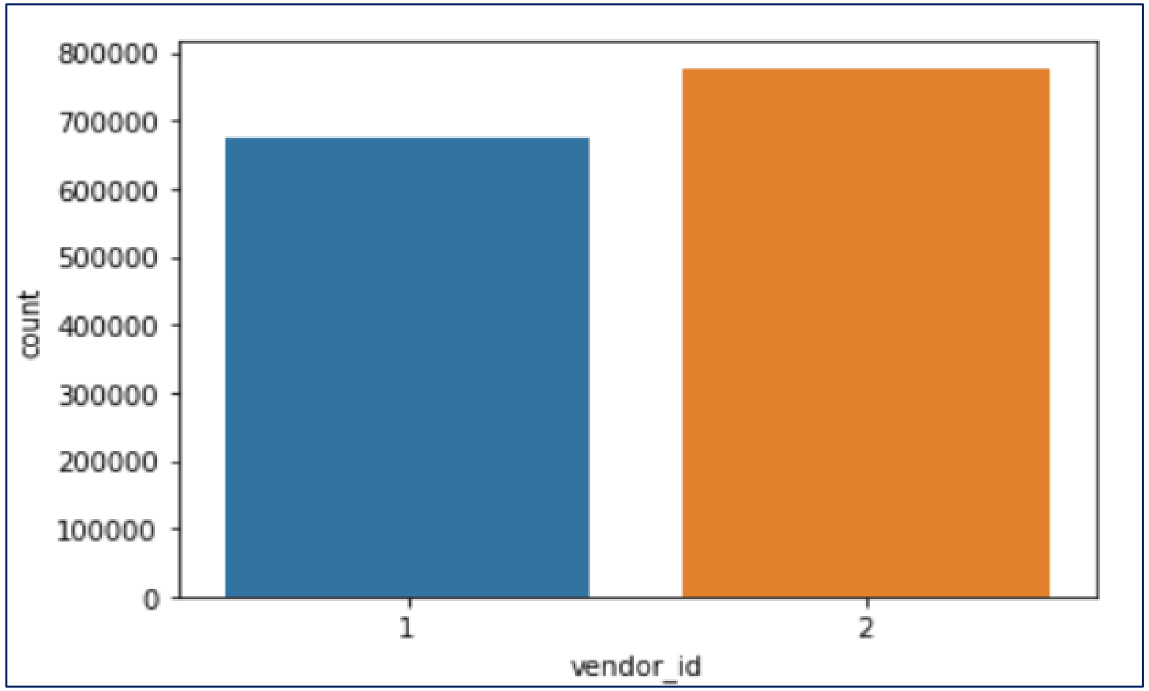
Passenger Count

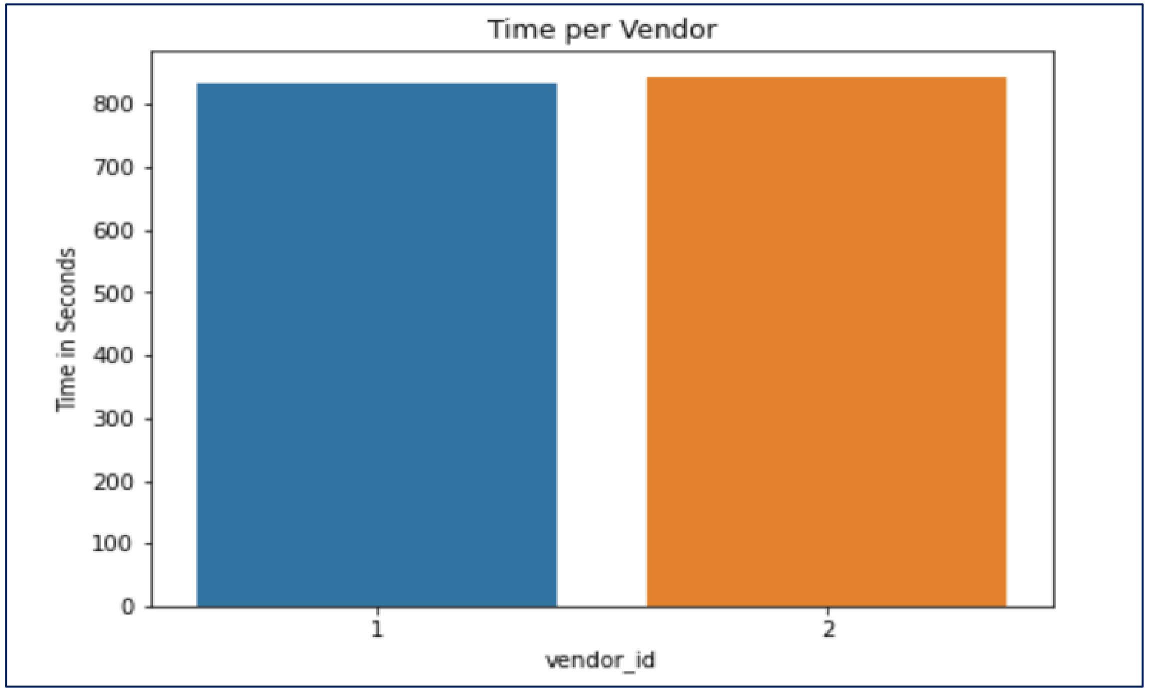




EDA

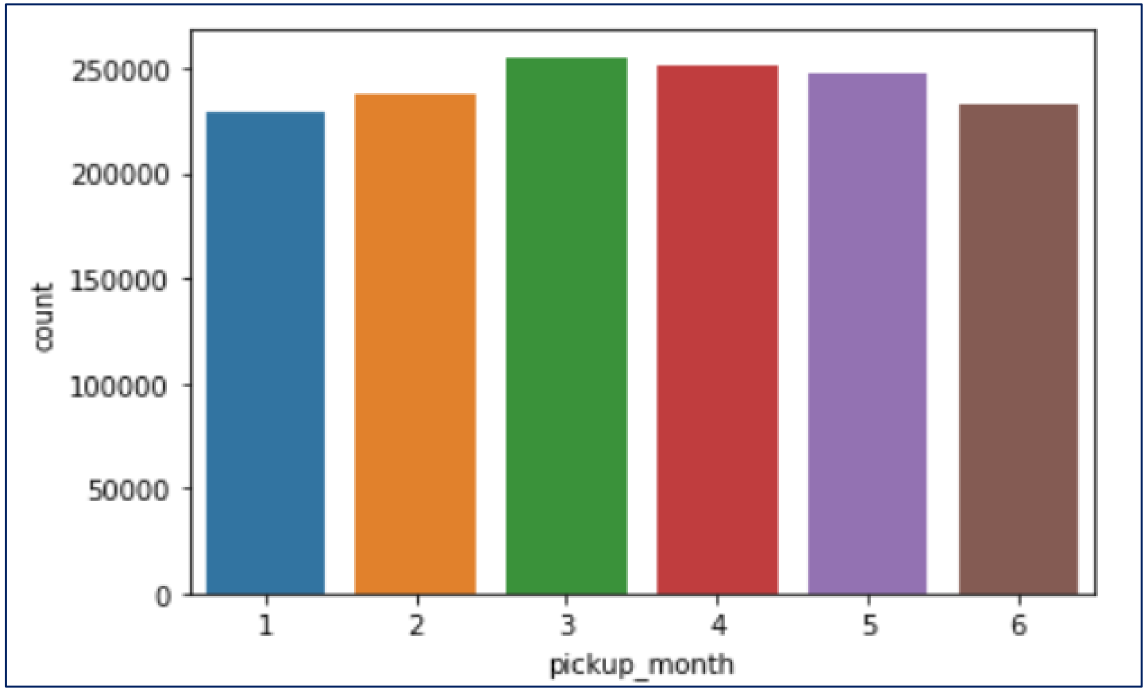
Vendor ID

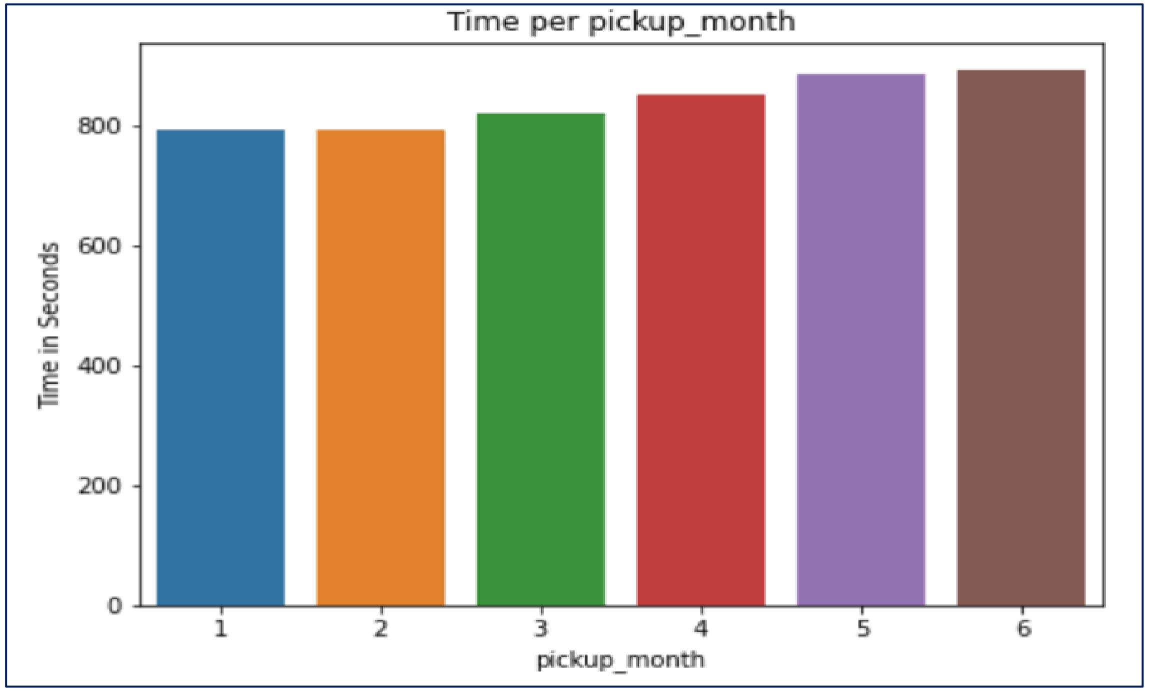




EDA

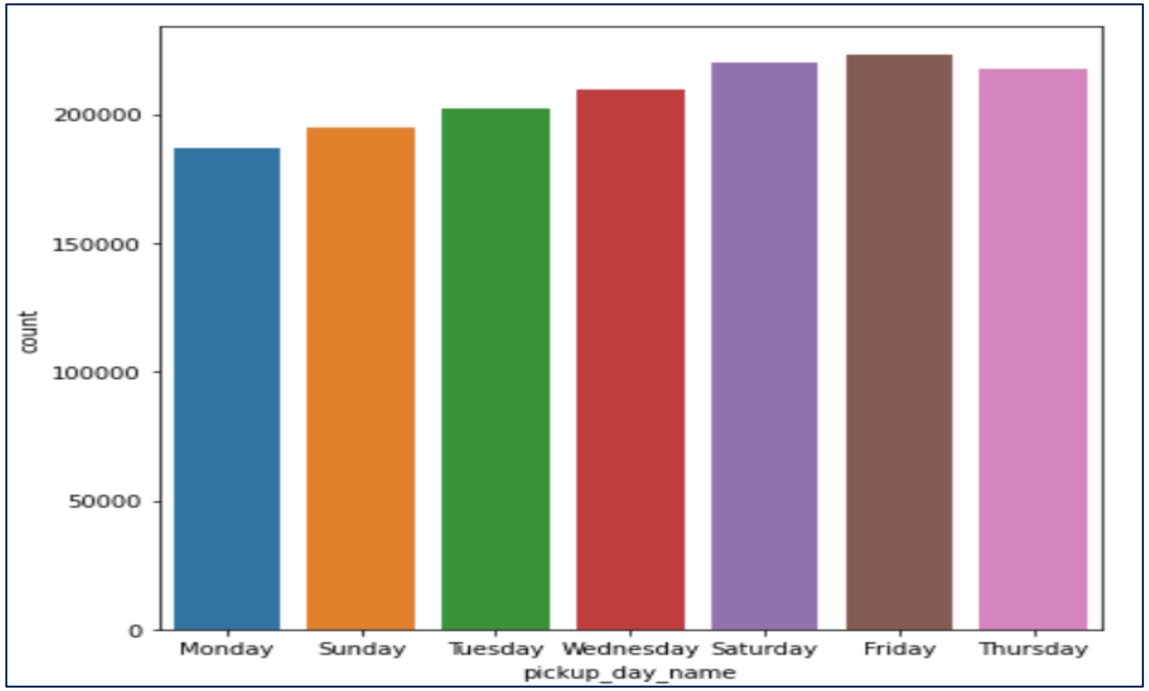
Pickup Month

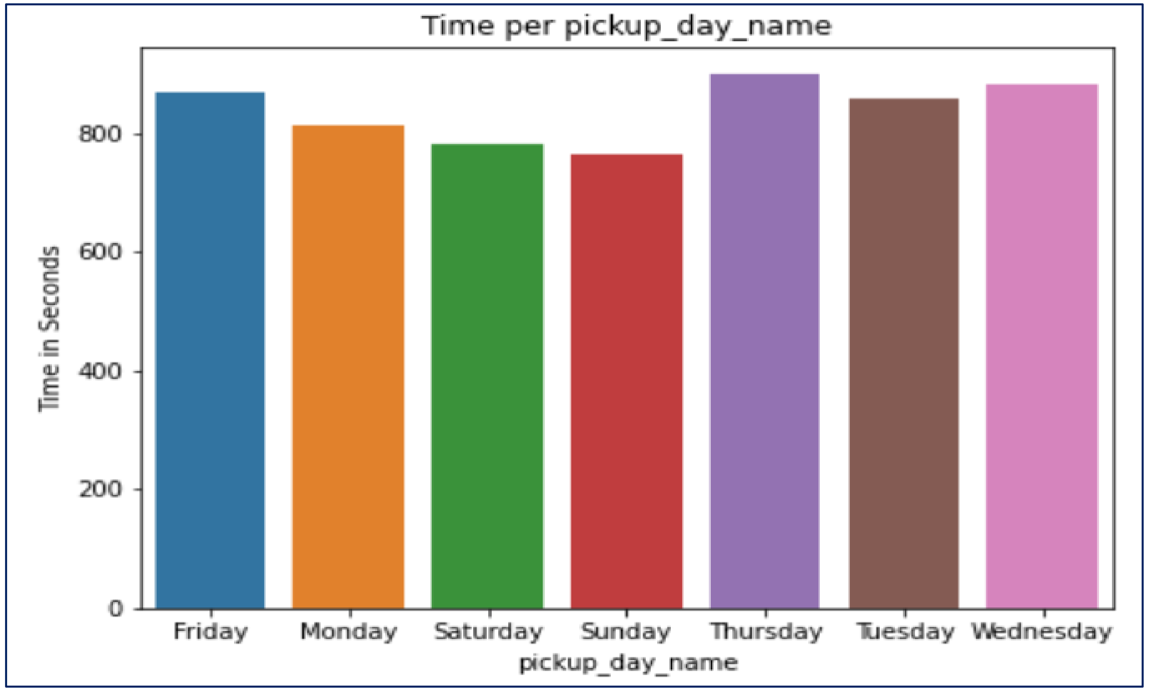




EDA

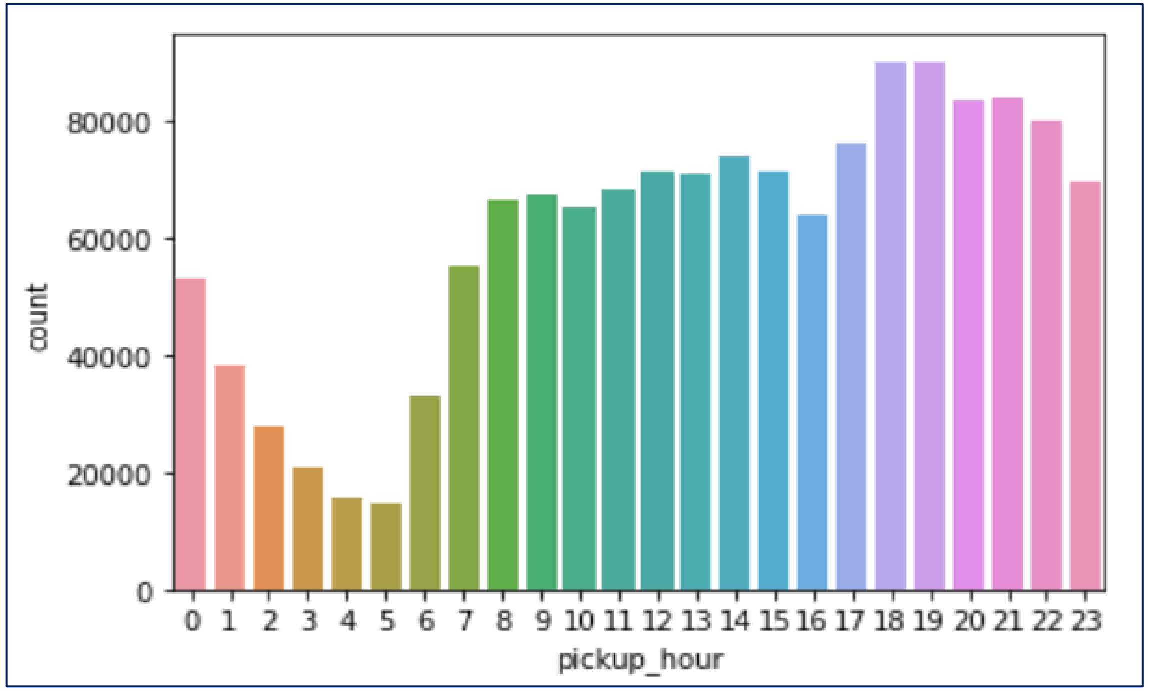
Pickup Day

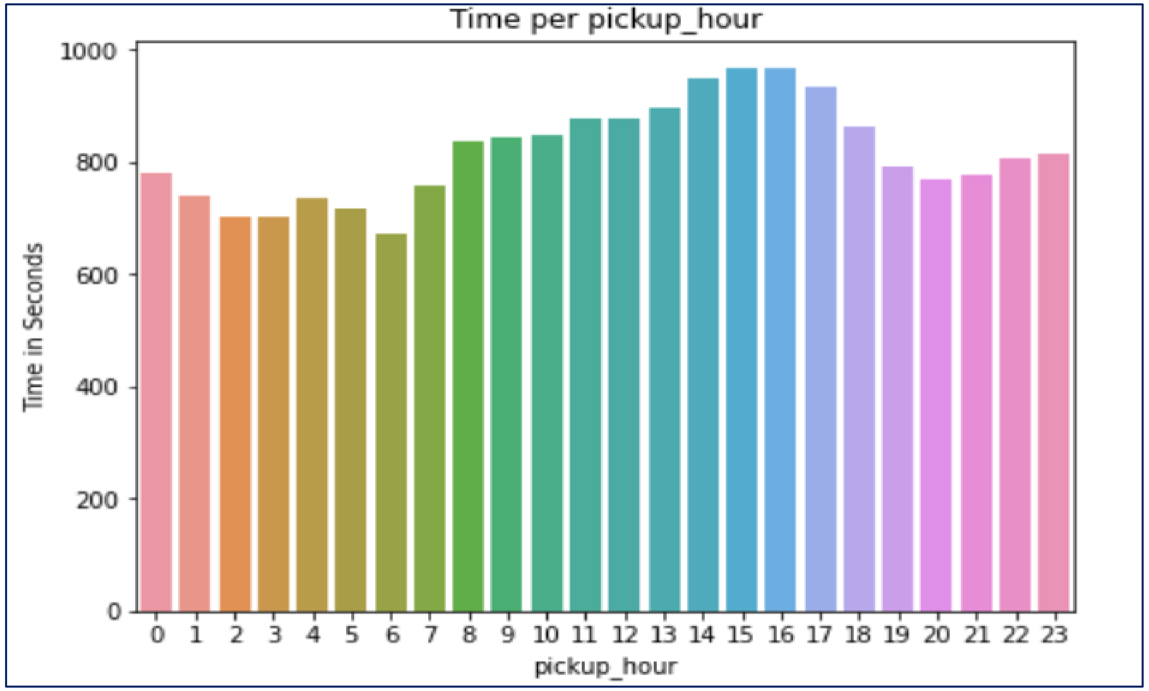




EDA

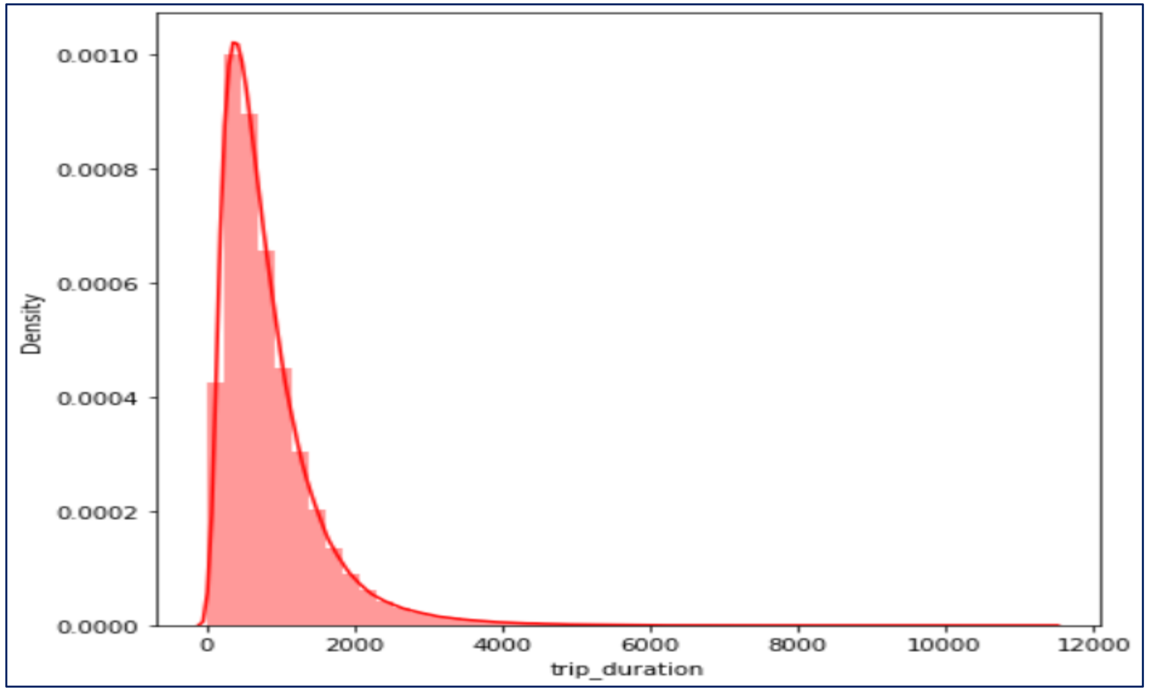
Pickup Hour

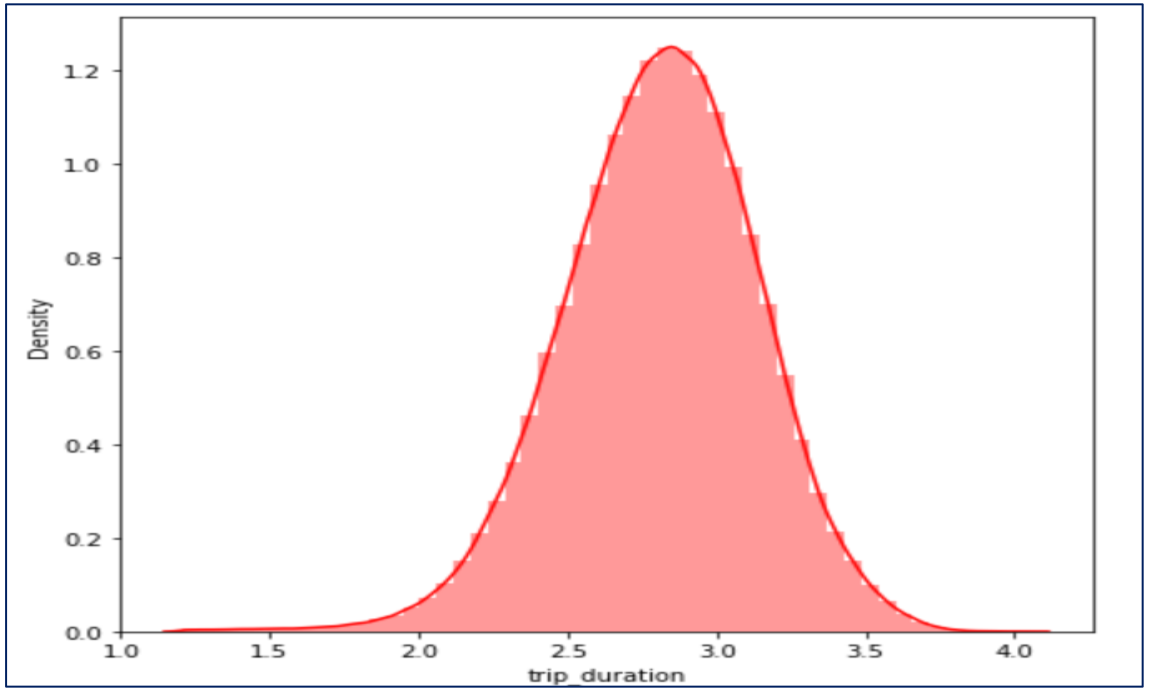




EDA

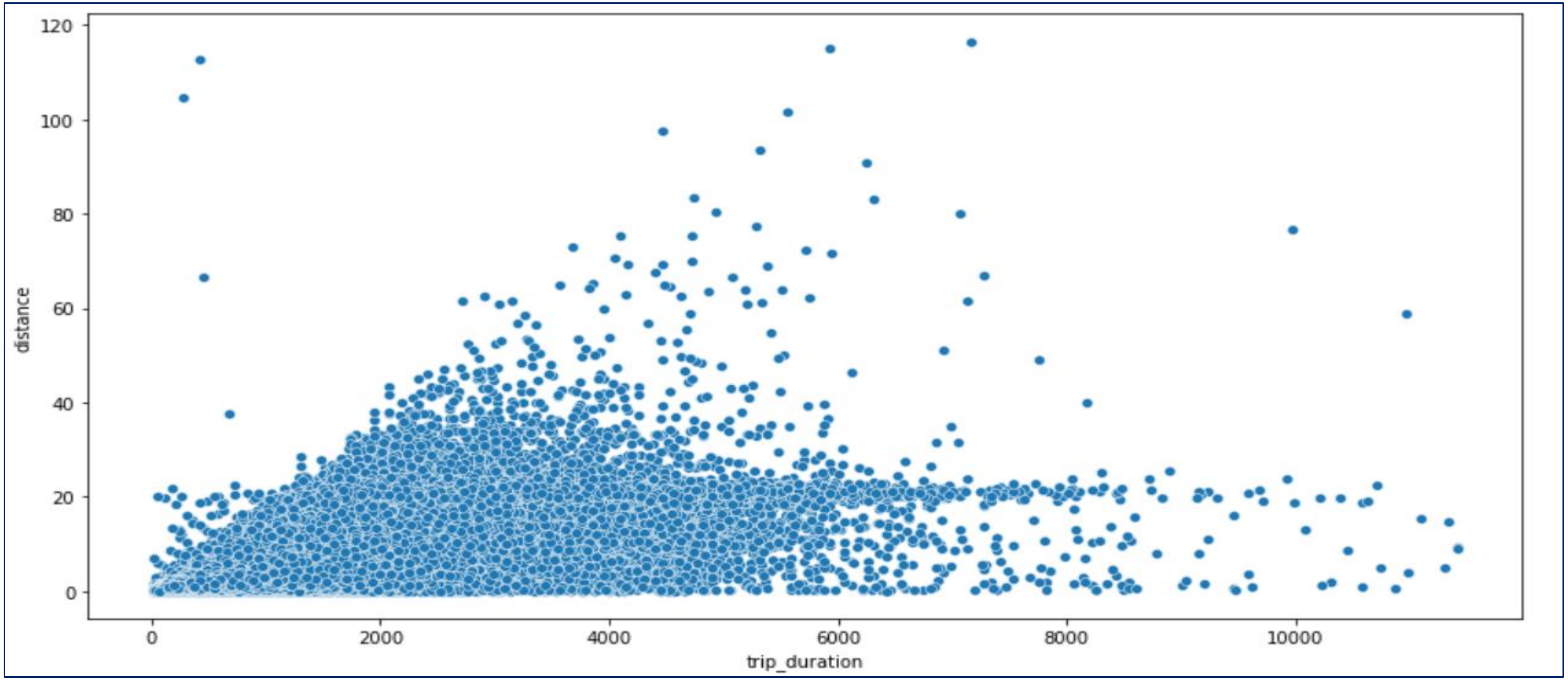
Trip Duration





EDA

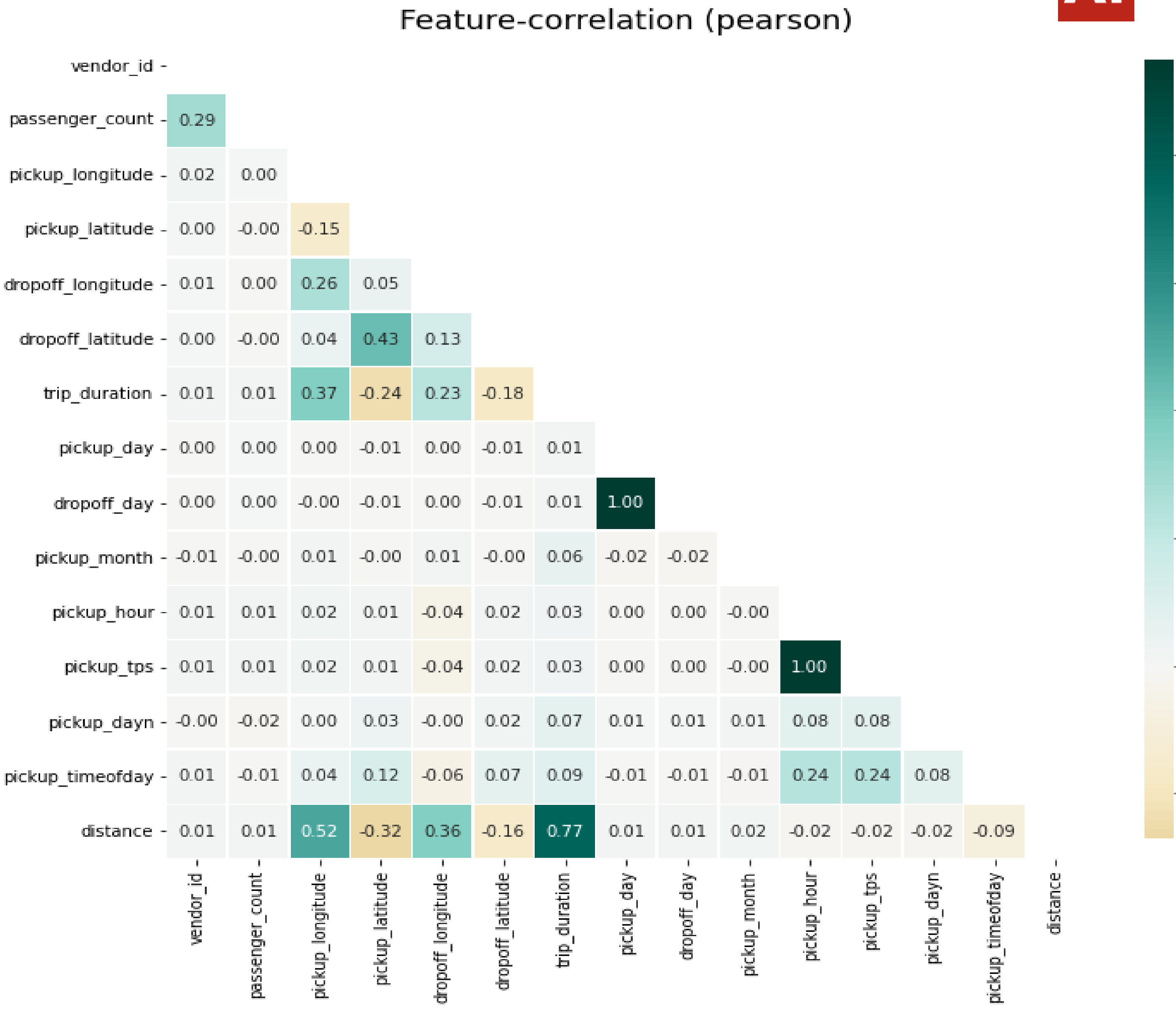
Distance vs Trip Duration



Feature Creation:

We have created the following features:

- pickup_day_name which contains the name of the day on which the ride was taken.
- pickup_hour with an hour of the day in the 24 - hour format.
- pickup_month with month number as January = 1 and December = 12.
- Distance from geographical coordinates in kms



Feature-correlation (pearson)

	vendor_id	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	trip_duration	pickup_day	dropoff_day	pickup_month	pickup_hour	pickup_tps	pickup_dayn	pickup_timeofday	distance
vendor_id															
passenger_count	0.29														
pickup_longitude	0.02	0.00													
pickup_latitude	0.00	-0.00	-0.15												
dropoff_longitude	0.01	0.00	0.26	0.05											
dropoff_latitude	0.00	-0.00	0.04	0.43	0.13										
trip_duration	0.01	0.01	0.37	-0.24	0.23	-0.18									
pickup_day	0.00	0.00	0.00	-0.01	0.00	-0.01	0.01								
dropoff_day	0.00	0.00	-0.00	-0.01	0.00	-0.01	0.01	1.00							
pickup_month	-0.01	-0.00	0.01	-0.00	0.01	-0.00	0.06	-0.02	-0.02						
pickup_hour	0.01	0.01	0.02	0.01	-0.04	0.02	0.03	0.00	0.00	-0.00					
pickup_tps	0.01	0.01	0.02	0.01	-0.04	0.02	0.03	0.00	0.00	-0.00	1.00				
pickup_dayn	-0.00	-0.02	0.00	0.03	-0.00	0.02	0.07	0.01	0.01	0.01	0.08	0.08			
pickup_timeofday	0.01	-0.01	0.04	0.12	-0.06	0.07	0.09	-0.01	-0.01	-0.01	0.24	0.24	0.08		
distance	0.01	0.01	0.52	-0.32	0.36	-0.16	0.77	0.01	0.01	0.02	-0.02	-0.02	-0.02	-0.09	

Model Creation:

Linear Regression : The linear regression model finds the set of θ coefficients that minimize the sum of squared errors.

Random Forest Regressor : Provides higher accuracy through cross validation. Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.

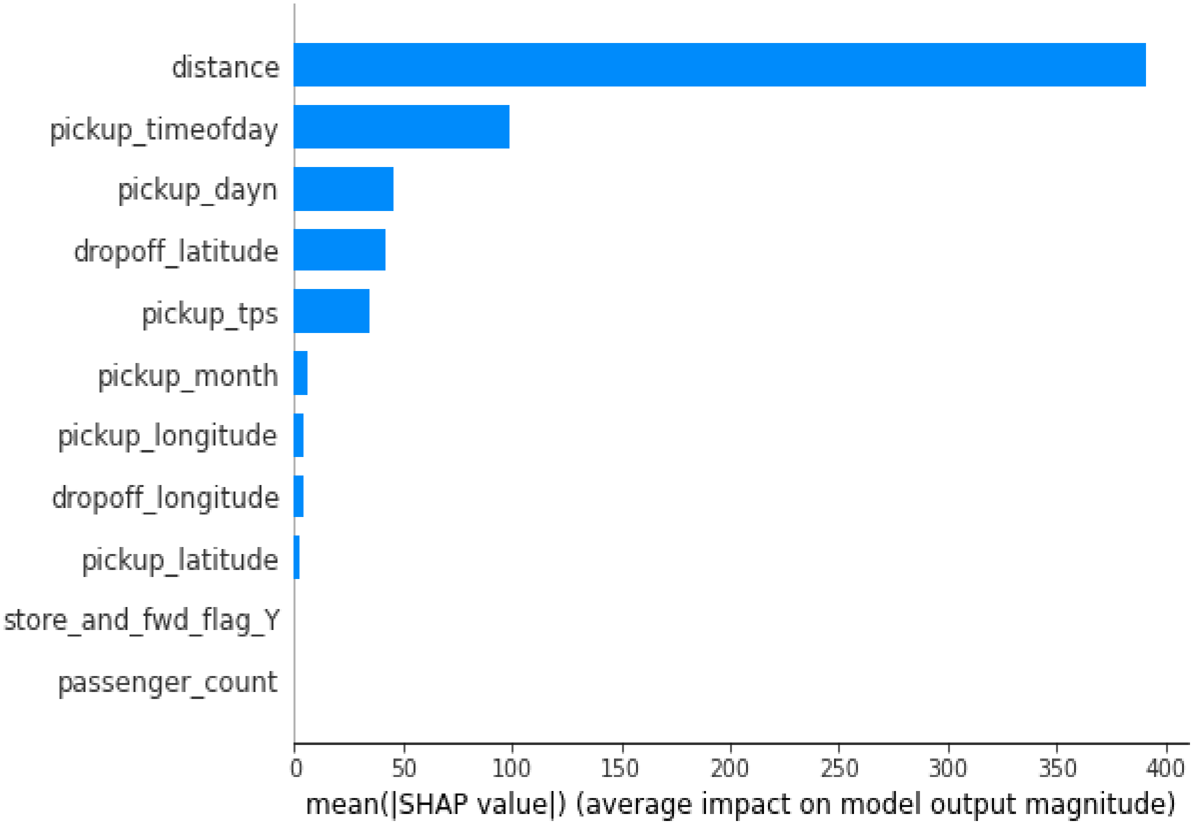
LGBM Regressor : It is a gradient boosting framework based on decision trees to increases the efficiency of the model and reduces memory usage.

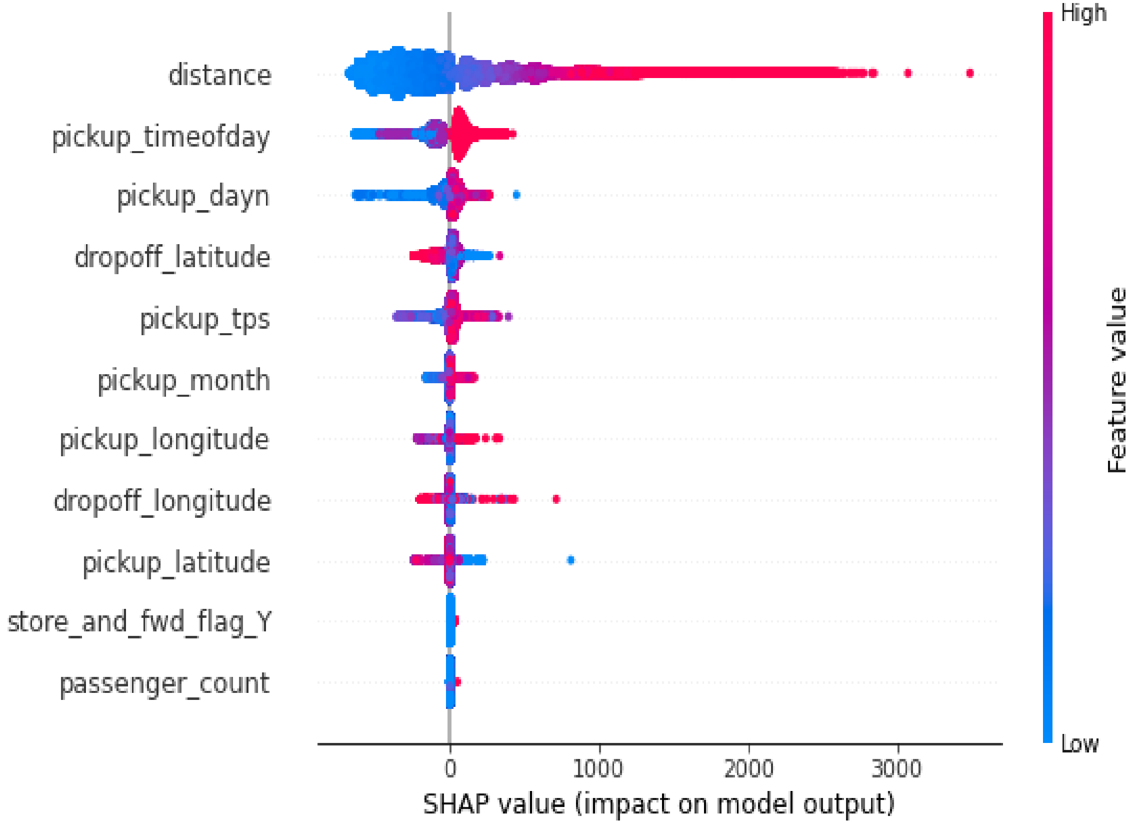
XGBoost : The dataset was very large, as a result for this type of problem XGBoost was applied in which all the attributes were taken and parallel processing of boosting trees executed. Another aspect of XGBoost is that it keeps a nice check between bias and variance which helps in better prediction.

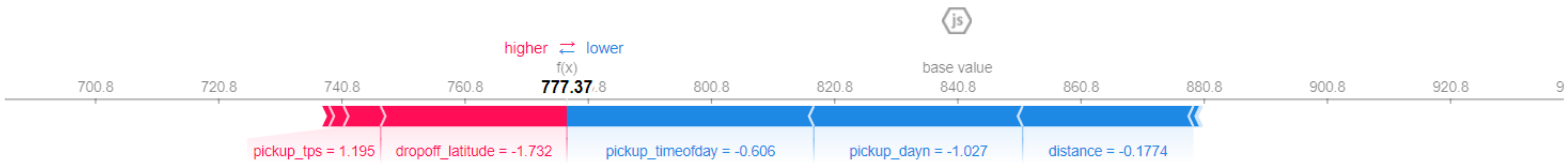
Model Evaluations:

Training Model	Train MAE	Test MAE	Train MSE	Test MSE	Train RMSE	Test RMSE	Train R2	Test R2	Adjusted R2
Linear Regression	269.2901	268.9965	159653.8131	159956.1934	399.5670	399.9452	0.6310	0.6314	0.6314
Random Forest Regressor	223.6980	223.3548	117301.2883	117756.9907	342.4927	343.1573	0.7289	0.7286	0.7286
LGBM Regressor	164.7305	175.7869	66466.9927	81785.4432	257.8119	285.9815	0.8463	0.8115	0.8115
XGB Regressor	215.5263	215.2902	112840.3943	113606.1822	335.9172	337.0551	0.7392	0.7382	0.7382

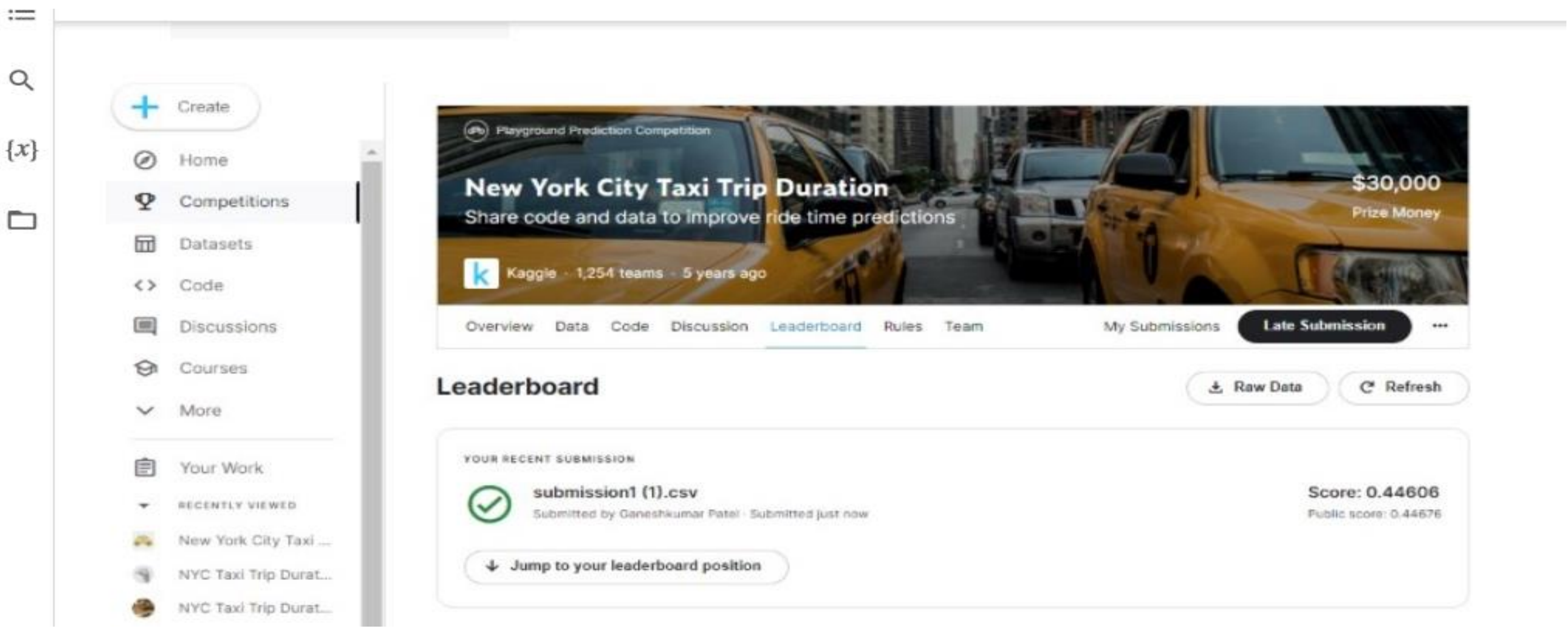
Explainability







Real world model performance check



New York City Taxi Trip Duration

Share code and data to improve ride time predictions

Leaderboard

YOUR RECENT SUBMISSION

submission1 (1).csv

Submitted by Ganeshkumar Patel · Submitted just now

Score: 0.44606

Public score: 0.44676

We get 0.44606 public score and 0.44676 private score on Kaggle platform. It means our model is working perfectly fine.

Conclusion:

- For Linear regression model, MSE and RMSE for training and testing are similar but has very poor R2 for training and testing data.
- Random Forest Regressor R2 increases, but not with significant amount.
- We can see that MSE and RMSE of XGB Regressor model are not varying much during training and testing time. Also the R2 is almost same for training and testing time.
- RMSE of LGBM Regressor model are very similar and their R2 is above 81% for training and test data.
- From above table, we can conclude LGBM Regressor is best model for our dataset.

Challenges we faced:

- Large dataset to handle which consumes time for complex algorithms.
- Need to Remove outliers to build generalize model.
- feature engineering and feature selection part as it affects the R2 score.
- Choosing the right model and metrics after several comparisions and hypertuning.