Capstone Project

Supervised ML – Regression
NYC Taxi Trip Time Prediction

By,
Prashant Shaw
Data Science Trainee at AlmaBetter

Agenda

- Introduction
- Problem Statement
- Attributes
- Data Exploration
- Univariate Analysis
  - Exploring Passenger Count Feature
  - Number of Trips done by each vendor
  - Number of trips in each day of the week
  - Trips during Each Hr
  - Trips during Each Time of the Day
  - Distribution of trip_duration
  - Pickup Locations
  - Dropoff Locations
- Bivariate Analysis
  - Trip Duration vs Vendor ID
  - Trip Duration per Hr
  - Trip Duration per Day and Time of day
  - Trip Duration vs Distance
- Correlation Heat Map
- Modelling
  - Evaluation Metrics
  - Machine Learning Models Used and Results
  - Gradient Boosting
  - XG Boosting
- Model Explainability
- Conclusion

Introduction

- 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).
- From the given dataset we have to predict the ride duration of the taxi trips in New York City.
- New York City is well known for traffic jams which happens due to roadworks, street closures or various other reason.
- Due to this high congestion on the roads it has become very much important to know the trip duration before starting any trip.
- This project will be developing machine learning model which will predict the trip timing based on the given dataset.
- This will give the person taking ride a prior estimation of the trip timings so that he/she can plan their work accordingly.

Problem Statement

- Now a days there is sudden increase in the popularity of app based taxi service providers like Uber, Curb, Lyft etc. in the NYC. So, it has become very much important for these companies to predict the ride timings more accurately before start of any trip.
- Knowing the trip duration beforehand will help the rider in planning their day accordingly.
- Also, for people taking taxis for going to office, knowing trip duration is very much important so that they can leave on proper time from their home.
- So, it has become extremely important to predict the trip duration accurately which depend on several parameters like trip distance, start time, pickup location, drop location etc.
- As trip duration depends on a lot of parameters, so the best way to predict the trip duration is with the help of some historical dataset (in this case NYC taxi dataset) by employing machine learning models.

Attributes

Independent Variables –

- id - A unique identifier for each trip.
- vendor_id - A code indicating the service provider associated with the trip record. (There are two service providers in the dataset Vendor-1 and Vendor-2).
- 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.

Dependent / Target Variable –

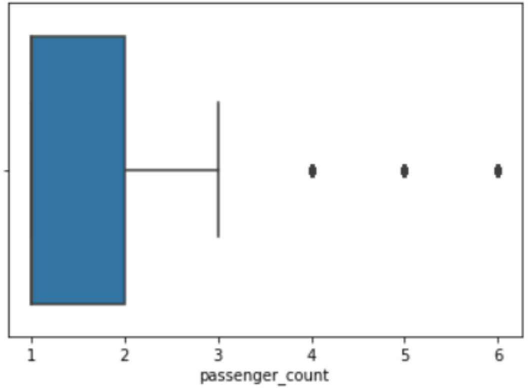
- trip_duration - Duration of the trip in seconds.

Data Exploration

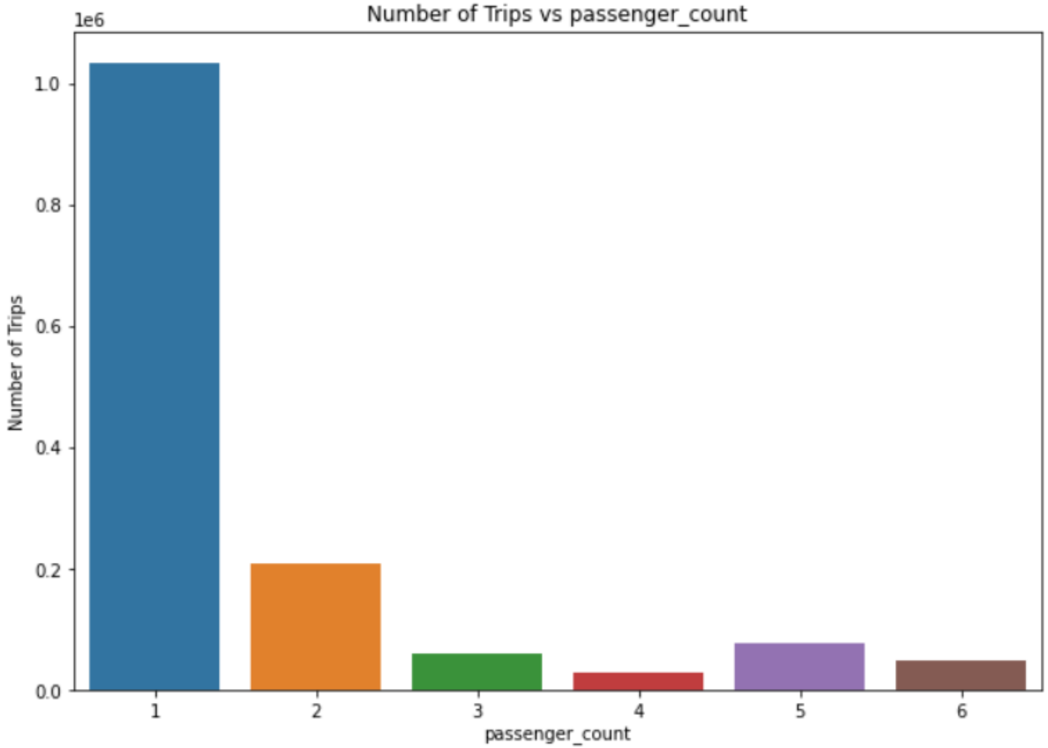
- Dataset contains 1458644 entries and 11 features.
- No null or duplicate entries present in the dataset.
- Datatype conversion needed for pickup and dropoff_datetime from object type to datetime dtype.
- vendor_id and store_and_fwd_flag are the two categorical variables.
- Feature engineering to be done on the datetime features to extract some more features.
- We can extract the distance covered from the pickup and drop-off latitude and longitude.
- Outlier treatment is needed for some of the features.

Univariate Analysis

Exploring Passenger Count Feature



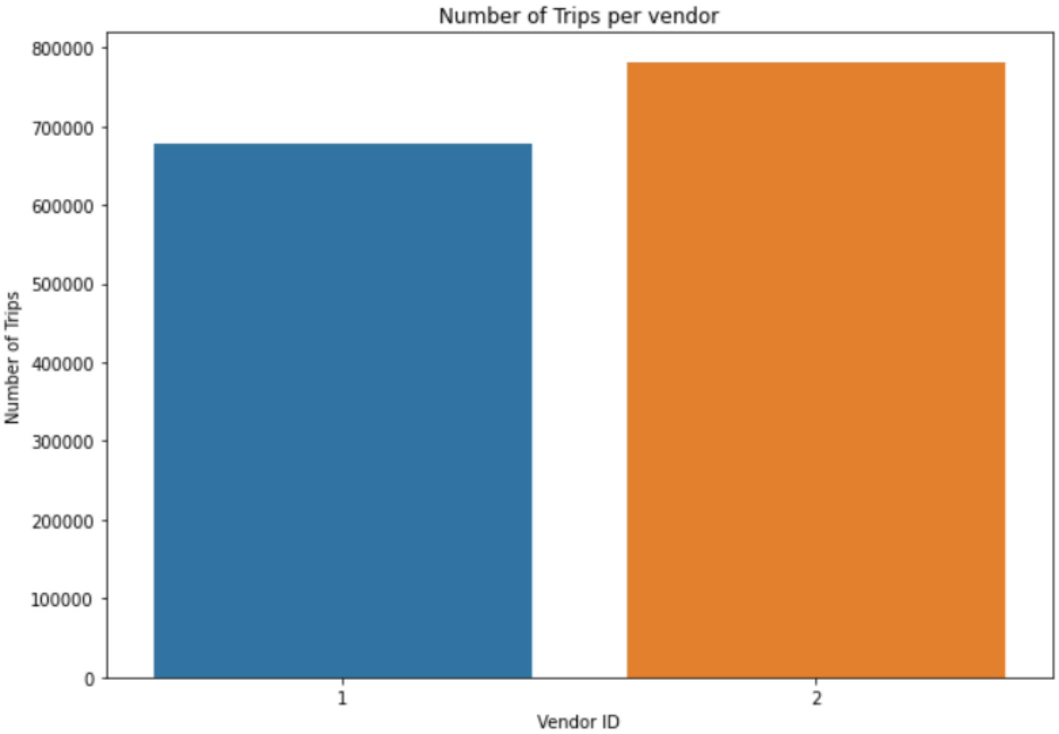
Box Plot for passenger_count



Most of the trips are done by solo travellers.

Univariate Analysis

Number of Trips done by each vendor

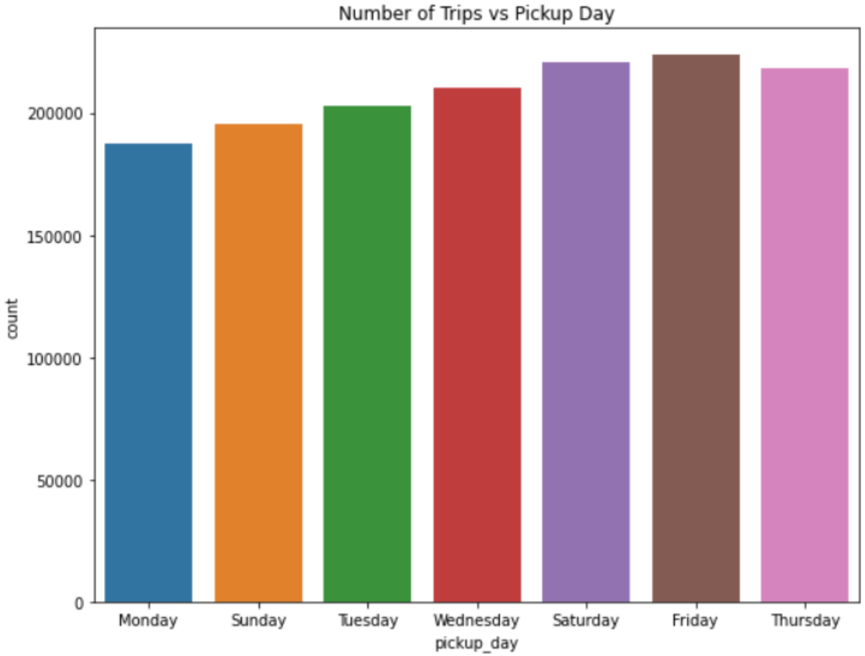


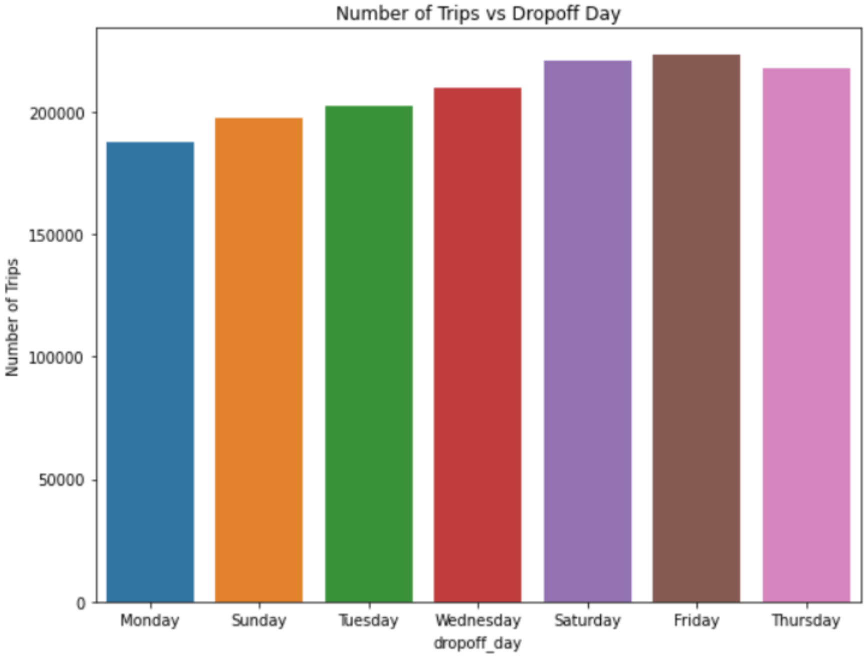
Trips done by Vendor 2 is little bit more than Vendor 1.

Univariate Analysis

Number of trips in each day of the week

Extracted pickup_day and dropoff_day from datetime feature.



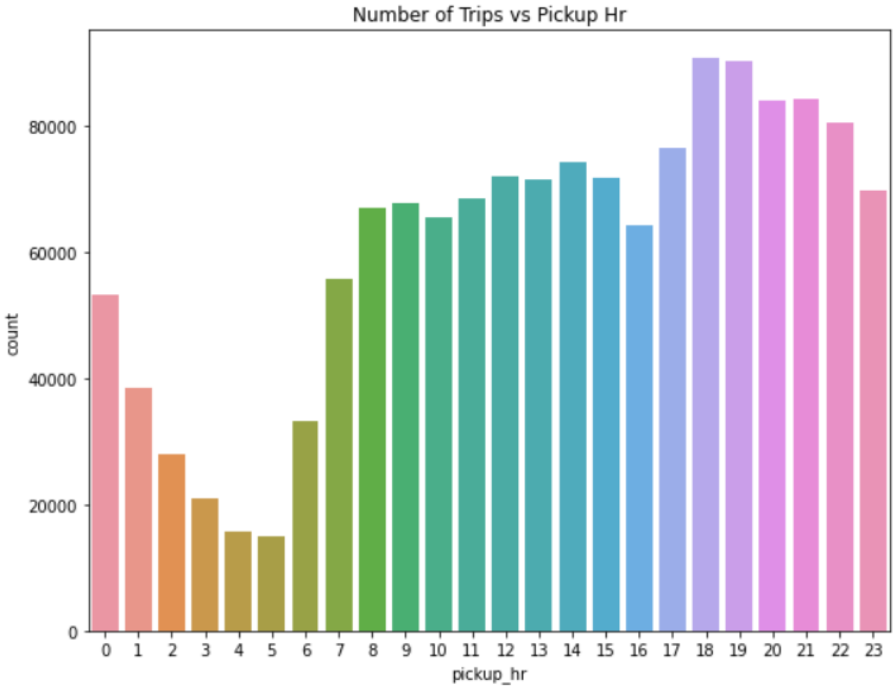


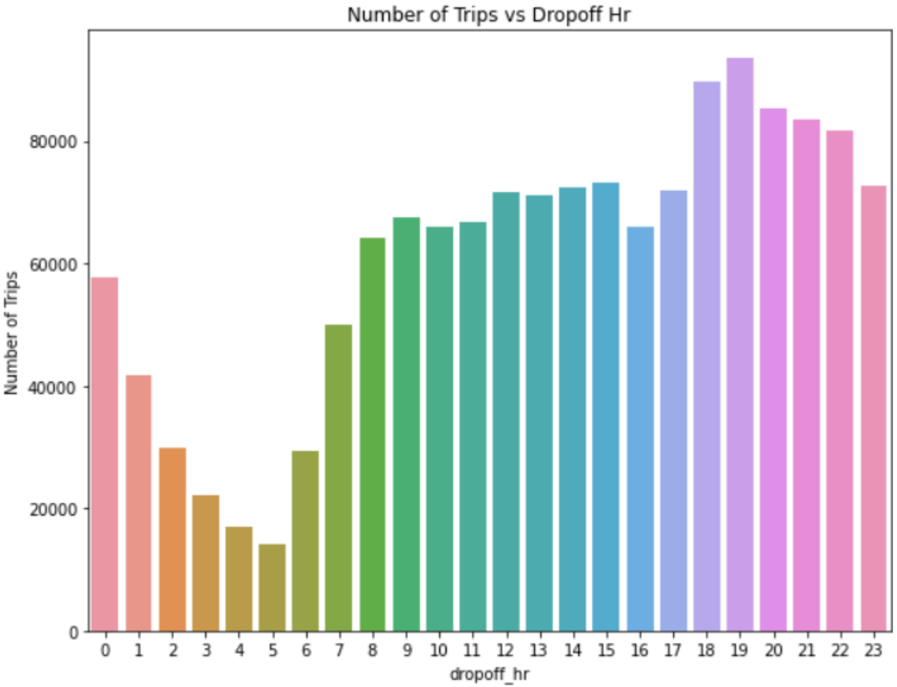
Most of the trips are done on Friday and Saturday, weekends people prefer to go more on trips.

Univariate Analysis

Trips during Each Hr

pickup_hr and dropoff_hr are also extracted from datetime feature.

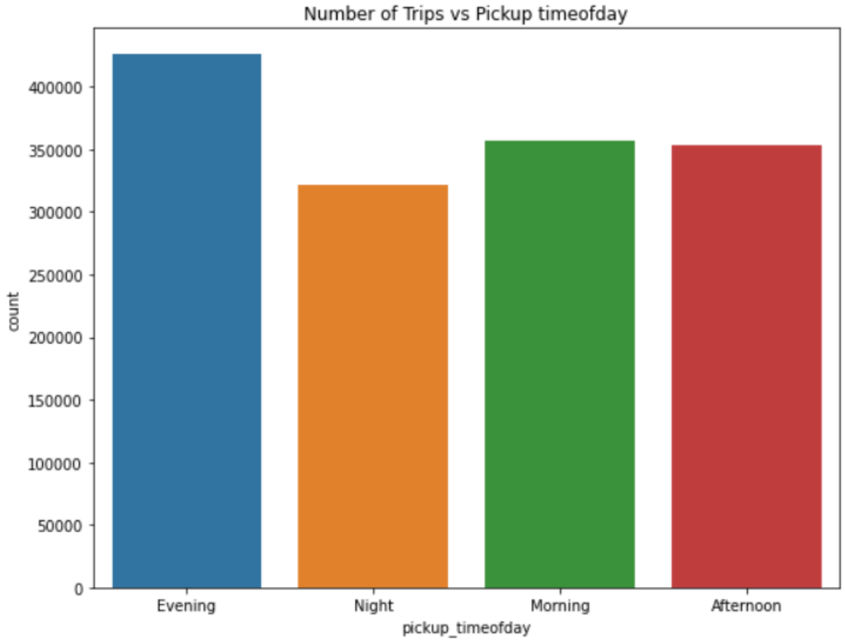


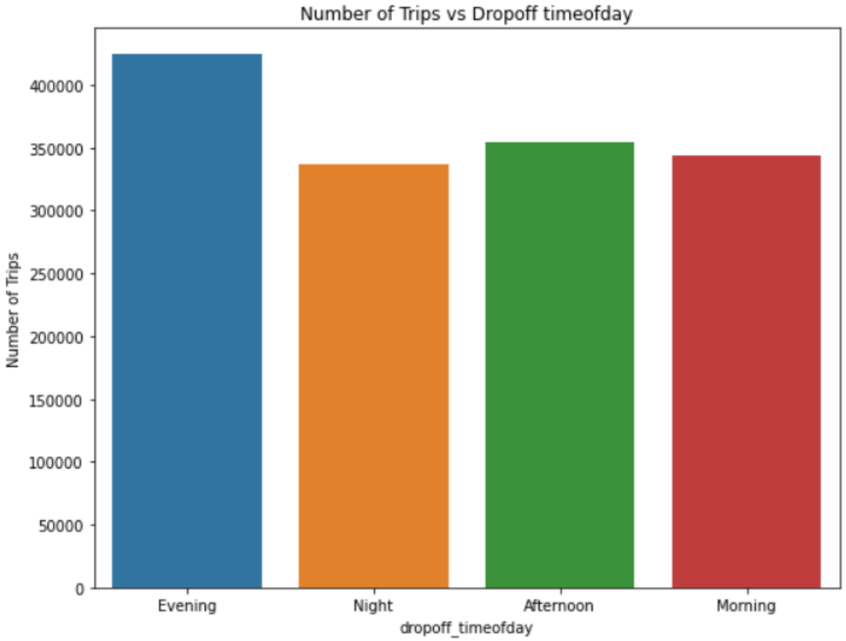


Most of the trips are done between 6:00 PM to 7:00 PM, reason might be this is the time when people will leave from offices and they are taking cab on their way back home or on Friday since we saw is the busiest day we can infer that most of the people are going for trips after office on Friday.

Univariate Analysis

Trips during Each Time of the Day

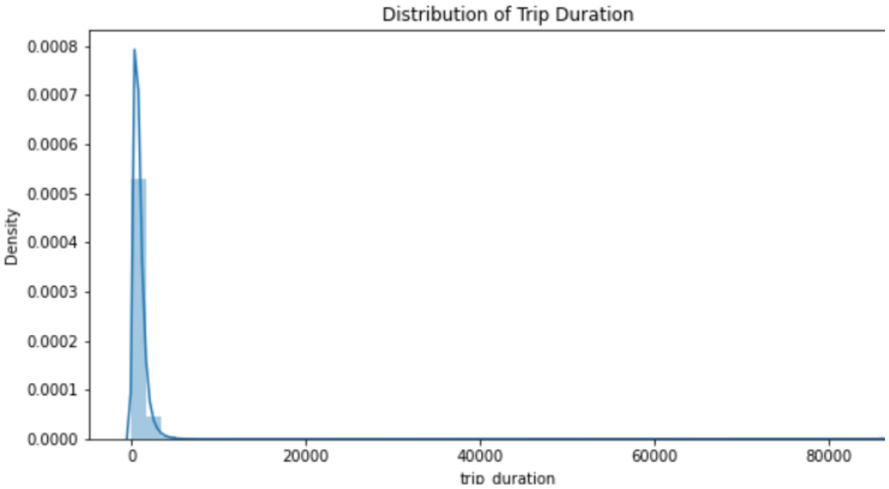




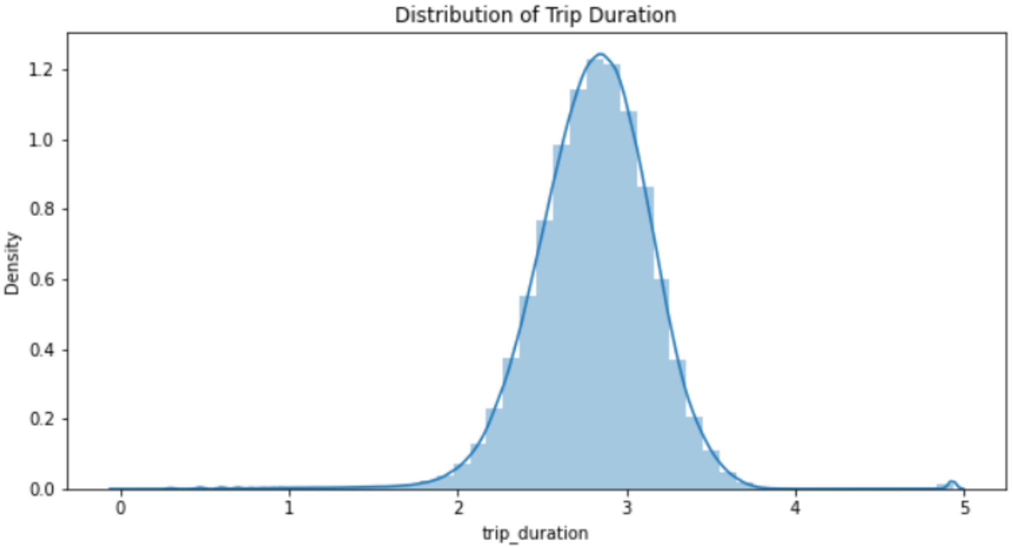
Most of the trips are taken in the evening we found the same thing in the previous graph that most of the trips are between 6:00 PM to 7:00 PM.

Univariate Analysis

Distribution of trip_duration

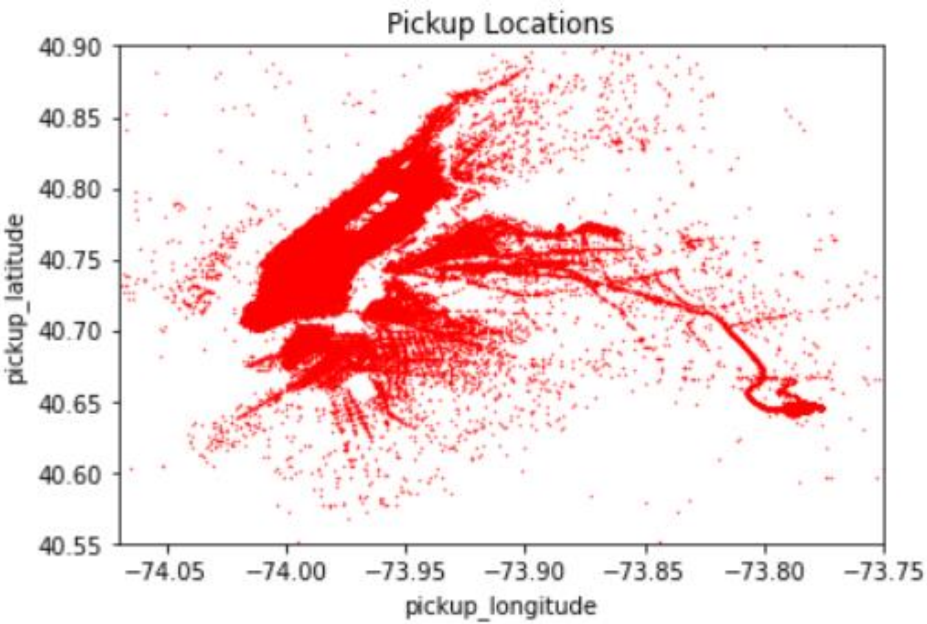


Skewness Removed After Doing Log Transformation

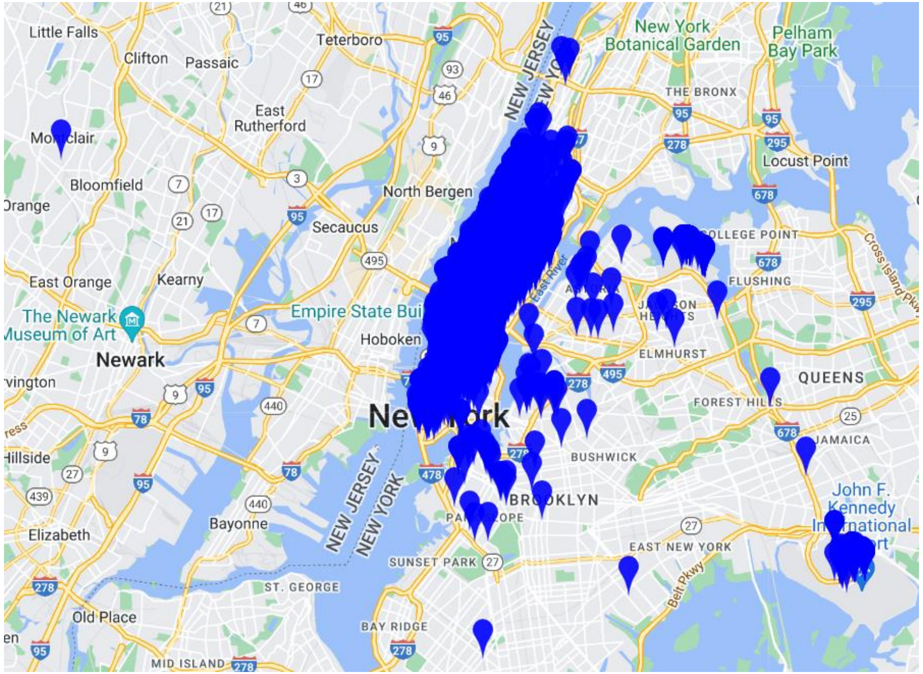


Univariate Analysis

Pickup Locations

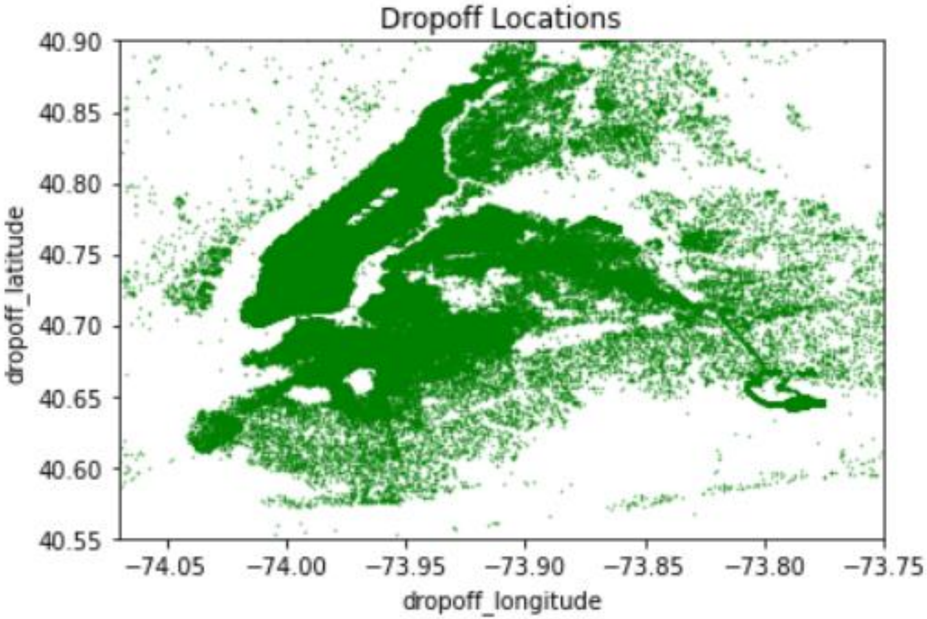


Some locations plotted on map

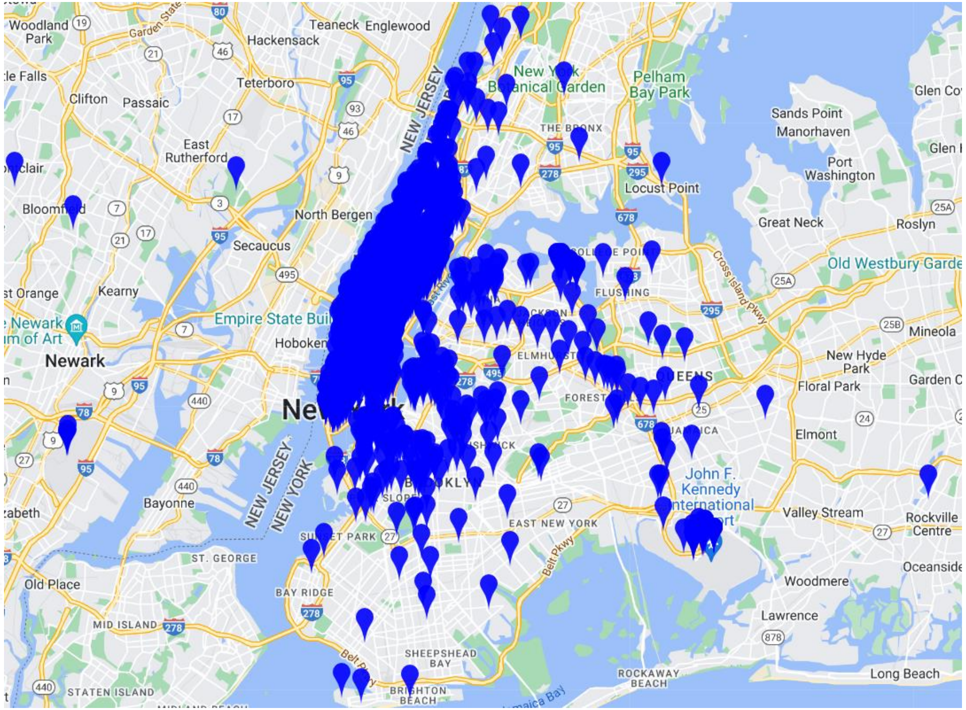


Univariate Analysis

Dropoff Locations

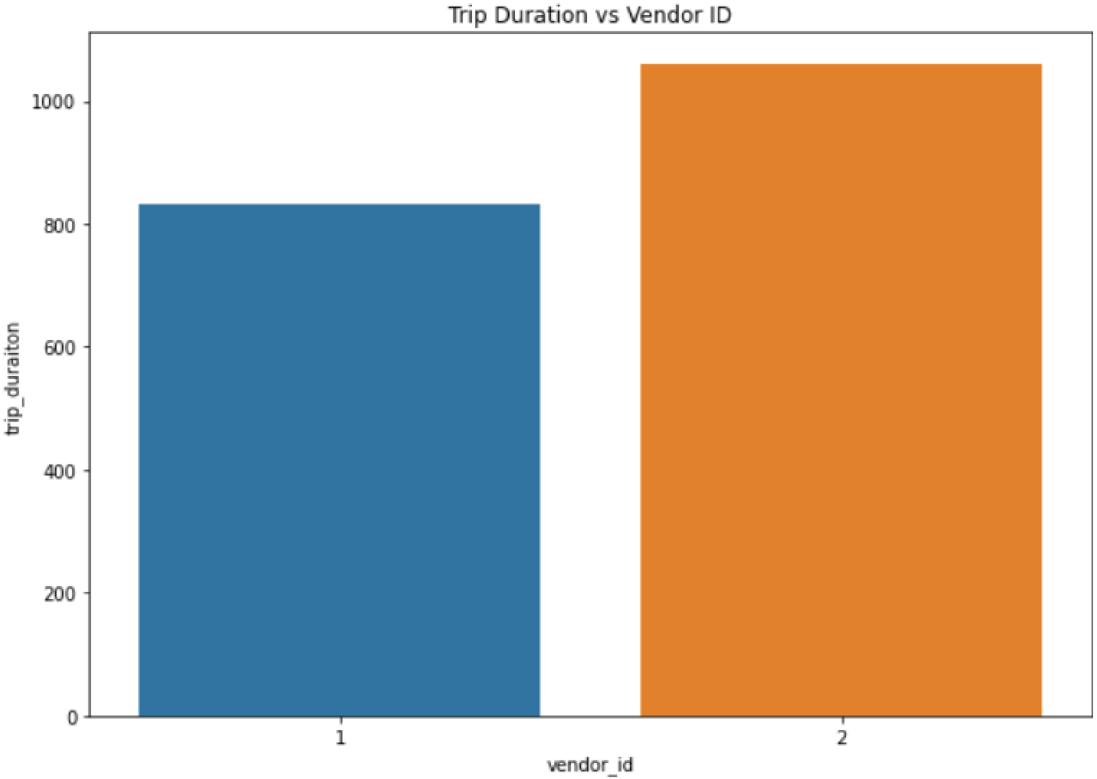


Some locations plotted on map



Bivariate Analysis

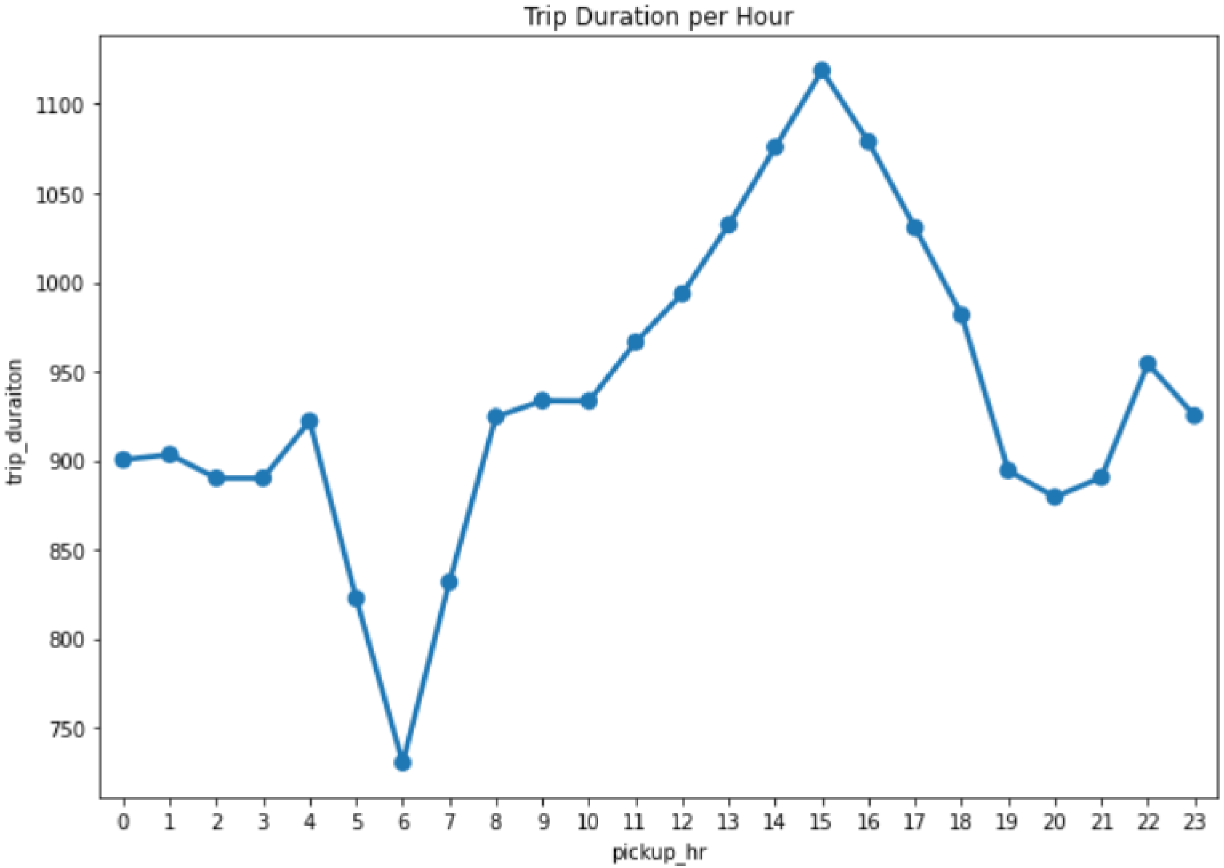
Trip Duration vs Vendor ID



On an average vendor 2 takes 200sec(or 3mins) more than vendor 1 per trip.

Bivariate Analysis

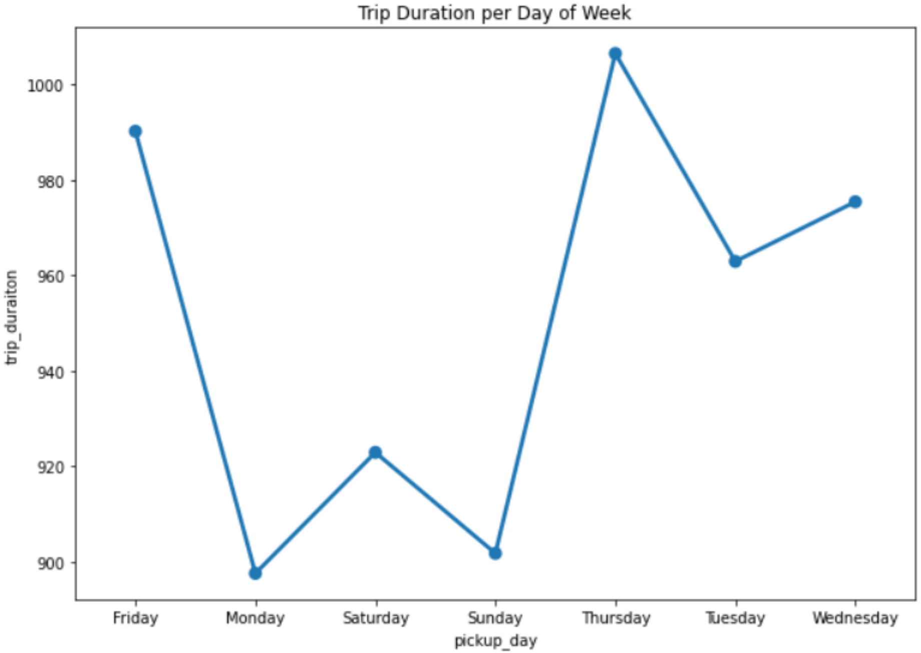
Trip Duration per Hr

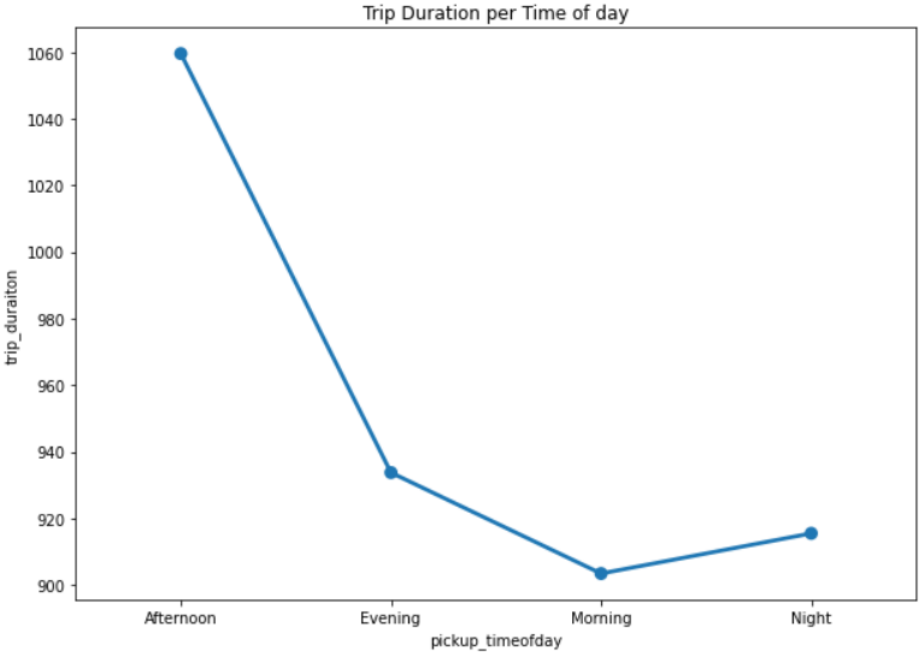


- Average trip_duration is highest at 3:00 PM.
- Morning 6:00 AM there is very less traffic.
- Traffic starts increasing after 6:00AM as more and more people starts going out for work, schools, vacation etc.
- Traffic starts decreasing after 3:00 PM as people starts coming to their home.
- Between 7:00 PM to 9:00 PM and 12:00 AM to 4:00AM there is average traffic.

Bivariate Analysis

Trip Duration per Day and Time of day





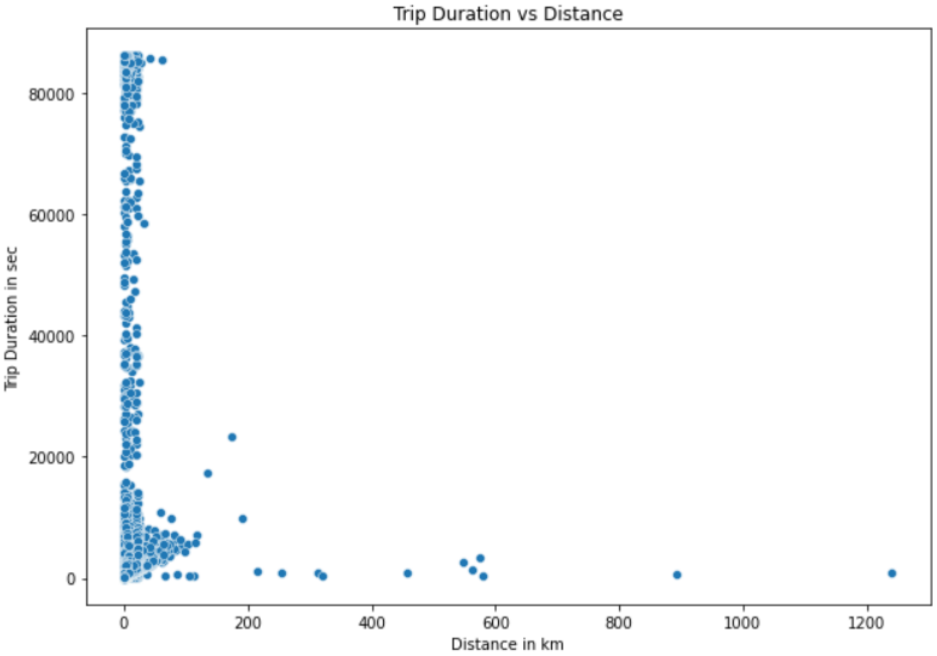
Average trip duration is less on Saturday, Sunday and Monday and it increases drastically on Tuesday, Wednesday and Thursday, having highest average trip duration on Thursday.

Average trip duration is most in the afternoon, lowest in the morning and average during evening and night.

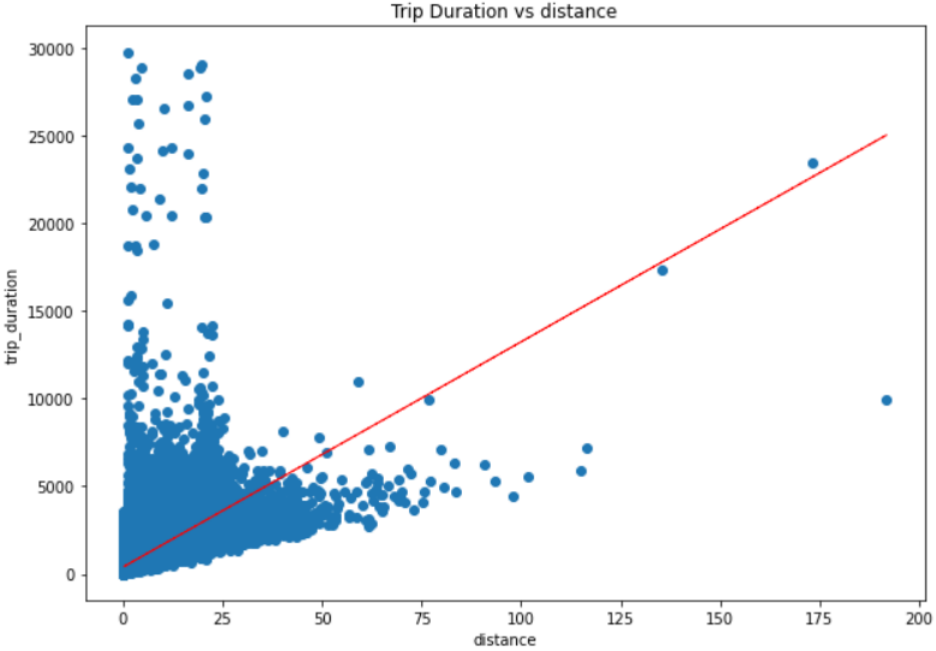
Bivariate Analysis

Trip Duration vs Distance

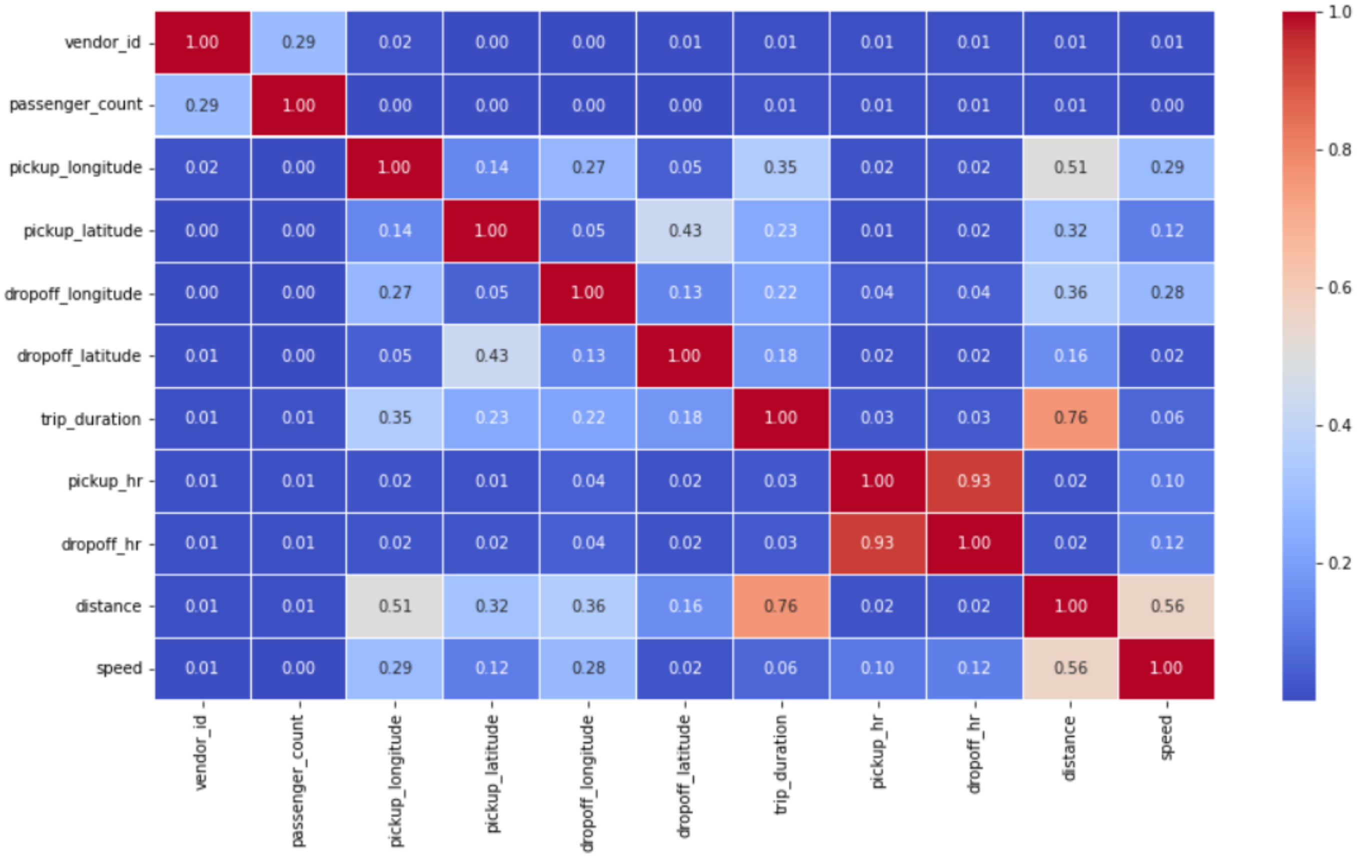
Raw Data



After Removing Outliers



Correlation Heat Map



	vendor_id	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	trip_duration	pickup_hr	dropoff_hr	distance	speed
vendor_id	1.00	0.29	0.02	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01
passenger_count	0.29	1.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.00
pickup_longitude	0.02	0.00	1.00	0.14	0.27	0.05	0.35	0.02	0.02	0.51	0.29
pickup_latitude	0.00	0.00	0.14	1.00	0.05	0.43	0.23	0.01	0.02	0.32	0.12
dropoff_longitude	0.00	0.00	0.27	0.05	1.00	0.13	0.22	0.04	0.04	0.36	0.28
dropoff_latitude	0.01	0.00	0.05	0.43	0.13	1.00	0.18	0.02	0.02	0.16	0.02
trip_duration	0.01	0.01	0.35	0.23	0.22	0.18	1.00	0.03	0.03	0.76	0.06
pickup_hr	0.01	0.01	0.02	0.01	0.04	0.02	0.03	1.00	0.93	0.02	0.10
dropoff_hr	0.01	0.01	0.02	0.02	0.04	0.02	0.03	0.93	1.00	0.02	0.12
distance	0.01	0.01	0.51	0.32	0.36	0.16	0.76	0.02	0.02	1.00	0.56
speed	0.01	0.00	0.29	0.12	0.28	0.02	0.06	0.10	0.12	0.56	1.00

- **Pickup_hr** and **dropoff_hr** are showing high correlation. We will be dropping **dropoff hr** from the dataset.
- **Speed** and **distance** is also showing correlation we will be dropping **speed** from the dataset.

Modelling

Evaluation Metrics

While data preparation and training a machine learning model is a key step in the machine learning, it’s equally important to measure the performance of this trained model. How well the model generalizes on the unseen data.

Evaluation metrics explain the performance of a model. An important aspect of evaluation metrics is their capability to discriminate among model results. Idea of building machine learning models works on a constructive feedback principle. Build a model, get feedback from metrics, make improvements and continue until a desirable accuracy is achieved.

Following are the Evaluation Metrics used in this project -

- **MSE**
- **RMSE**
- $\boldsymbol{R^2}$ **-** It describes how good our model is as compared to a no brainer model that just predicts the mean value of target from the train set as predictions.
- **Adjusted** $\boldsymbol{R^2}$ **-** R^2 suffers from the problem that the score improves on increasing the number of independent variables even though the model is not improving much.
  Adjusted R^2 adjusts for the increasing predictor and shows improvement only if there is a real improvement.

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \widehat{y}\right)^2}_{\text{The square of the difference between actual and predicted}}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

$$R^2 = 1 - \frac{\text{MSE(model)}}{\text{MSE(baseline)}}$$

$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1}\right)\times(1-R^2)\right]$$

where:

n = number of observations

k = number of independent variables

R_a^2 = adjusted R^2

Modelling

Machine Learning Models Used and Results

Features used in Training the Model

Features
vendor_id
passenger_count
pickup_hr
distance
store_and_fwd_flag_N
store_and_fwd_flag_Y
pickup_day_Friday
pickup_day_Monday
pickup_day_Saturday
pickup_day_Sunday
pickup_day_Thursday
pickup_day_Tuesday
pickup_day_Wednesday
pickup_timeofday_Afternoon
pickup_timeofday_Evening
pickup_timeofday_Morning
pickup_timeofday_Night

Models Used with respective scores for test data

Models	MSE	RMSE	R2	Adjusted R2
Linear Regression	171972.702	414.696	0.610	0.610
Lasso	171973.144	414.696	0.610	0.610
Ridge	171973.143	414.696	0.610	0.610
Elasticnet	171973.159	414.696	0.610	0.610
Polynomial Features	160417.946	400.522	0.636	0.636
Decision Tree	131198.160	362.213	0.702	0.702
Random Forest	131254.235	362.290	0.702	0.702
XG Boost	128167.600	358.005	**0.717**	**0.717**
Gradient Boosting	128619.268	358.635	0.708	0.708

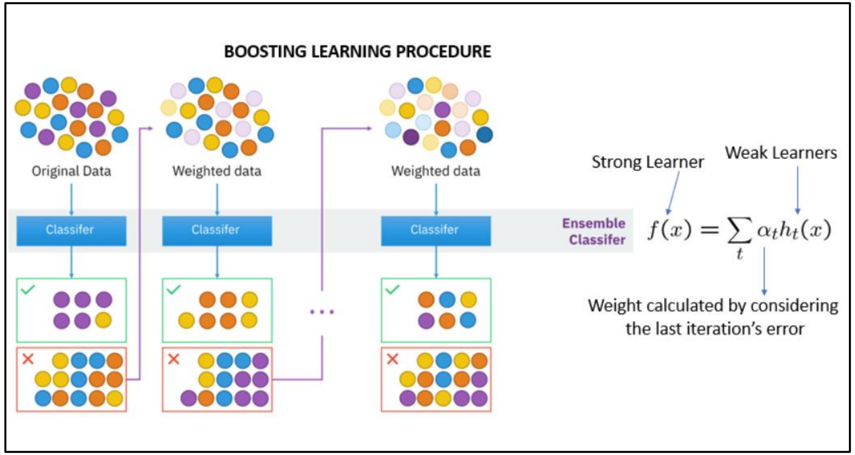
From the above table it is evident that XG Boost & Gradient Boosting are the best performing models.

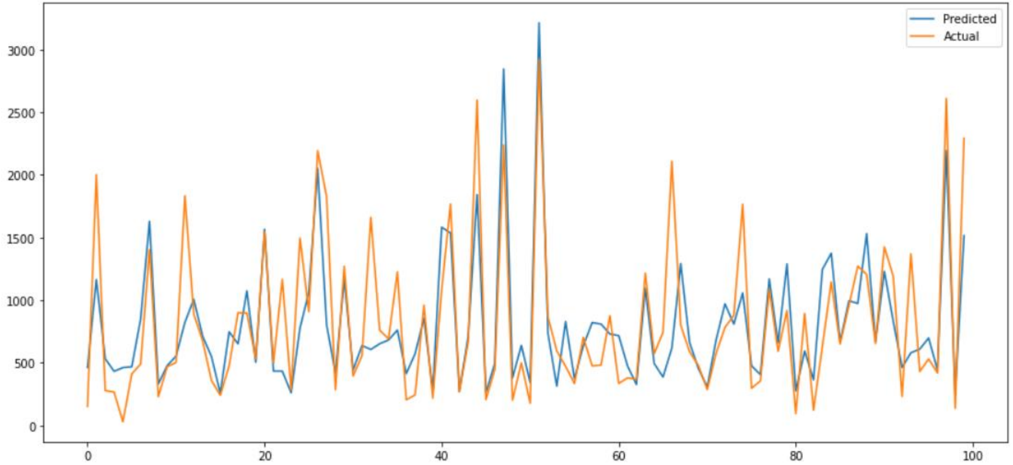
Modelling

Gradient Boosting

Boosting fit a sequence of weak learners – models that are only slightly better than random guessing, such as small decision trees – to weighted versions of the data. More weight is given to examples that were misclassified by earlier rounds.

Gradient boosting is one of the boosting methods where we create multiple weak models (decision trees) and combine them to get better performance model.
It build models sequentially and these subsequent models try to reduce the errors of the previous model by building the new model on the errors or residuals of the previous model.
The objective is to minimize a loss function by adding weak learners using gradient descent.


BOOSTING LEARNING PROCEDURE
Original Data
Weighted data
Weighted data
Classifer
Classifer
Classifer
Ensemble
Classifer
Strong Learner
Weak Learners
$f(x) = \sum_t \alpha_t h_t(x)$
Weight calculated by considering the last iteration's error



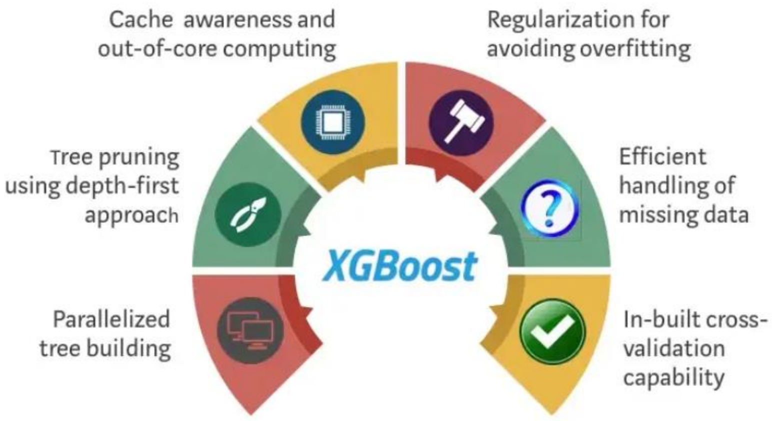
Predicted vs Actual Results for First 100 rows of Test Data

Modelling

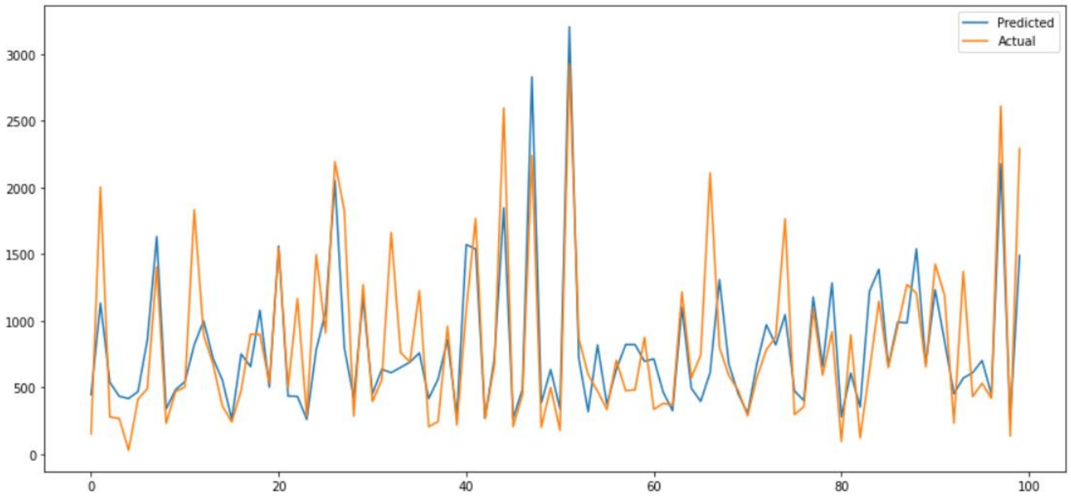
XG Boosting

Extreme Gradient Boosting (XGBoost) is just an extension of gradient boosting framework.

XGBoost improves upon the base GBM framework through systems optimization and algorithmic enhancements.


Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient handling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

How XGBoost optimizes standard GBM algorithm

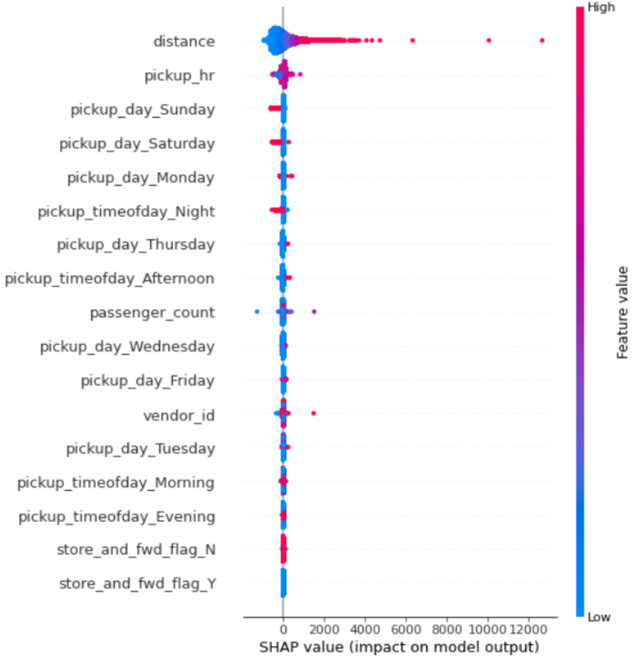


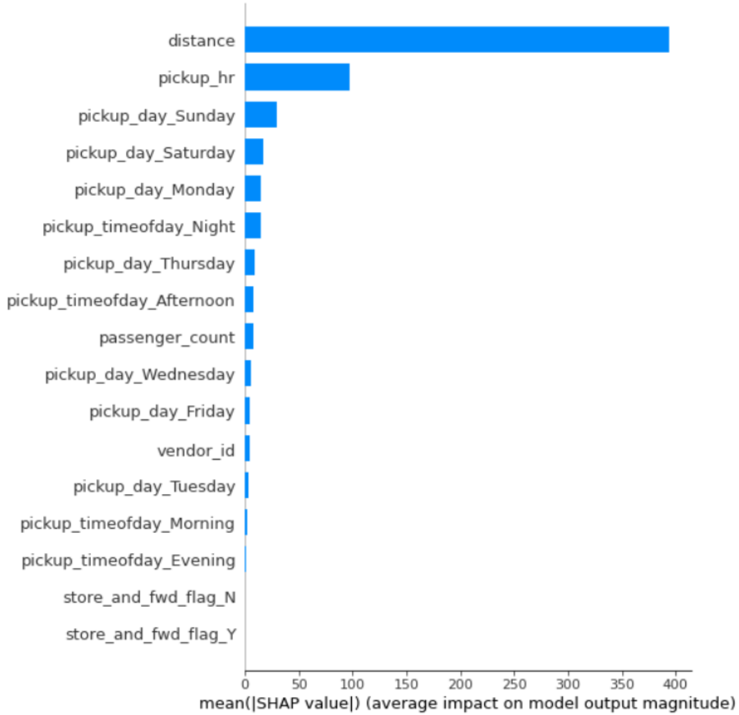
Predicted vs Actual Results for First 100 rows of Test Data

Model Explainability

XG Boost Model Explainability Using SHAP

SHAP (Shapley Additive exPlanations) is based on game theory - how to share a reward among team members in a cooperative game?
The goal of SHAP is to explain the prediction by computing the contribution of each feature to the final result.

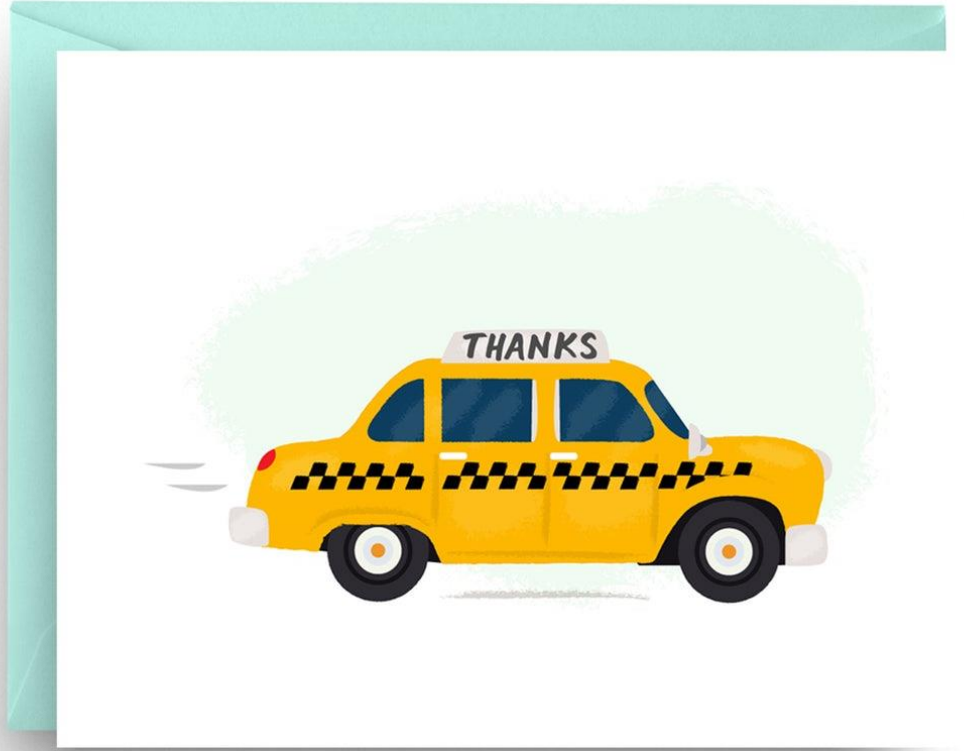




Distance and Pickup_hr are the most impacting feature in predicting the trip duration.

Conclusion

- We have predicted the Trip Duration for NYC Taxi dataset.
- Pickup and Dropoff latitude and longitude are used for getting the distance of the trip.
- Most of the trips are done on Friday and Saturday, weekends people prefer to go more on trips.
- Most of the trips are done between 6:00 PM to 7:00 PM, reason might be this is the time when people will leave from offices and they are taking cab on their way back home or on Friday since we saw is the busiest day we can infer that most of the people are going for trips after office on Friday.
- Trips done by Vendor 2 is bit more than Vendor 1.
- Average trip duration is most in the afternoon, lowest in the morning and average during evening and night.
- MSE, RMSE, R2 and adjusted R2 are the metrics used to evaluate the machine learning models.
- Hyperparameter tuning and Gridsearch is done to get the best fit for the models.
- XG Boost is the best performing model with R2 and adjusted R2 of 0.717.
- SHAP is used to explain the results of XG Boost regressor.
- Distance is the most impacting feature in predicting Trip Duration.
- Pickup_hr is also contributing a fair role in predicting Trip Duration as the traffic condition at various times will decide how much time will it take to complete the trip.


THANKS