Capstone Project - 2
NYC Taxi Trip Time Prediction

Team members
Gauri Agarwal
Saurav kumar
(Data Panthers)

Index

- Defining problem statement
- Data cleaning and feature engineering
- Feature Selection
- Prepare Dataset for modeling
- Apply Model
- Validation and Hyperparameter tuning
- Challenges
- Conclusion

Problem Statement

First of all before going through any code or analysis we must know what is the reason for doing this analysis.

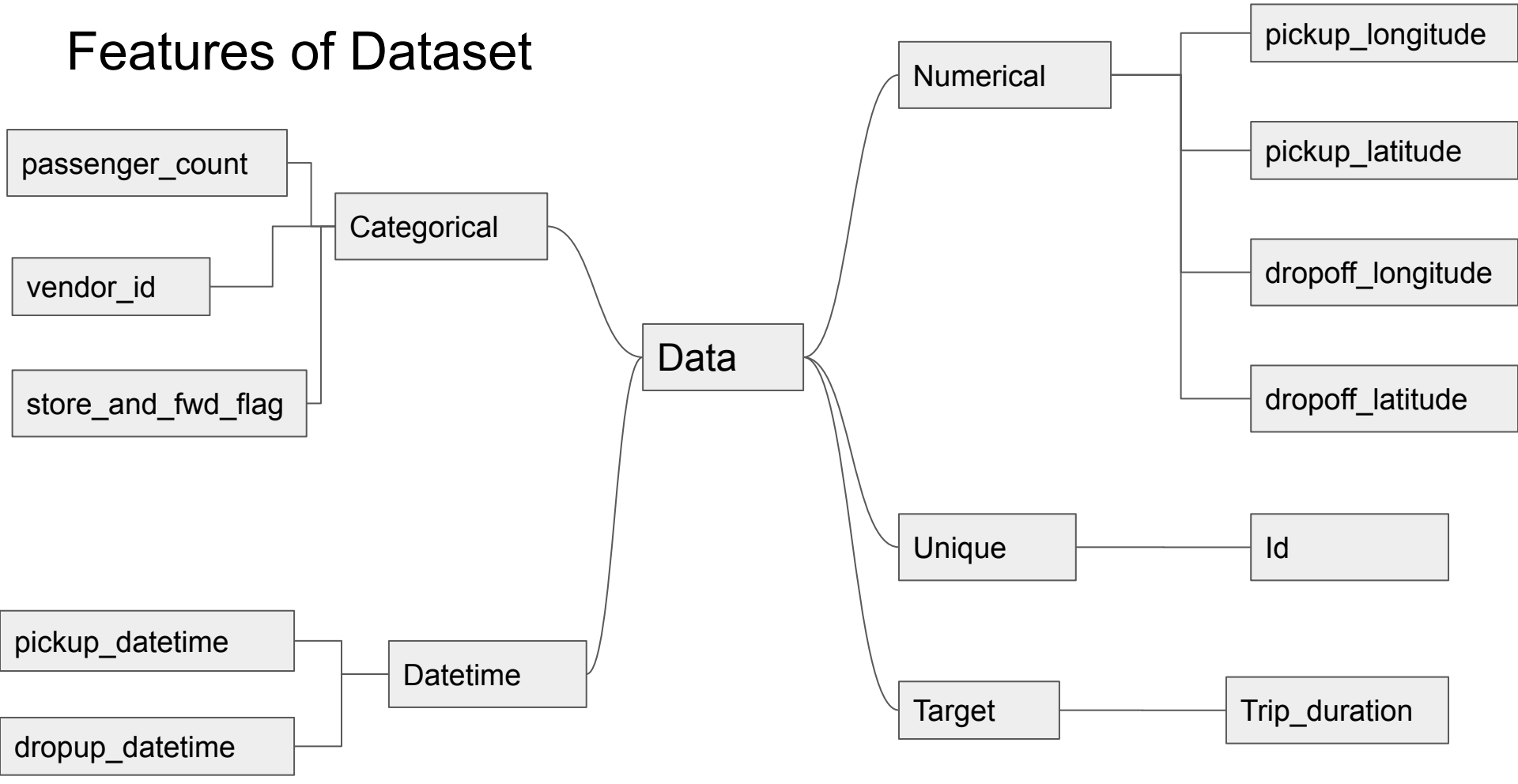
Data provided to us have many attributes which refer to a particular trip, our task is to remove noise from that data, and then find relation among the different attributes, to visualize the behavior of attribute or relation of two or more attributes using eda. To gain understanding from the data, we will use Python to undertake exploratory data analysis.

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

Data pipeline

- **Data Processing 1 :** In this first we remove the unnecessary feature like customer id, and remove the outliers
- **Feature Engineering :** In this we go through each feature, add new feature, change the column containing datetime value, do encoding on categorical variable.
- **EDA:** In this part we do some exploratory data analysis on the feature selected in part 1 and 2 to see the trend.
- **Create a model :** First we create a baseline model, then slowly increase the model complexity for better performance.

Features of Dataset



Data Summary

- 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).
- Important Features in NYC taxi trip 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

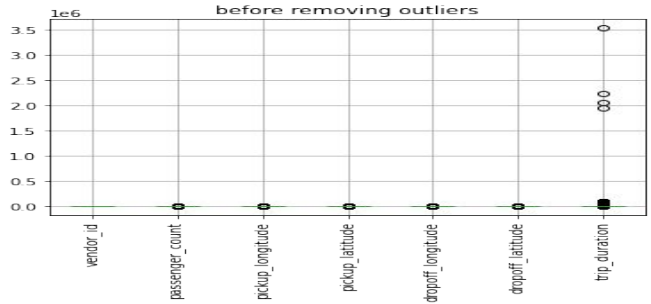
Data Summary

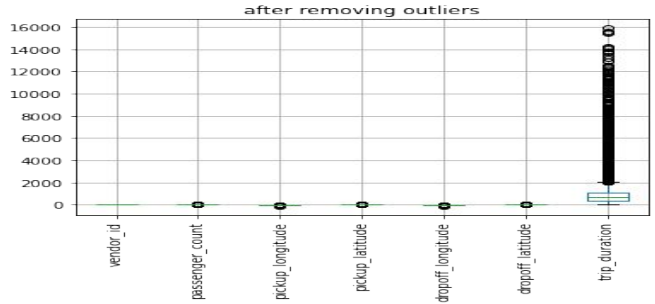
- Some important Features in NYC taxi dataset
  - dropoff_latitude - the latitude where the meter was disengaged
  - store_and_fwd_flag - This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server - Y=store and forward; N=not a store and forward trip
  - trip_duration - duration of the trip in seconds

Data Cleaning and Feature engineering

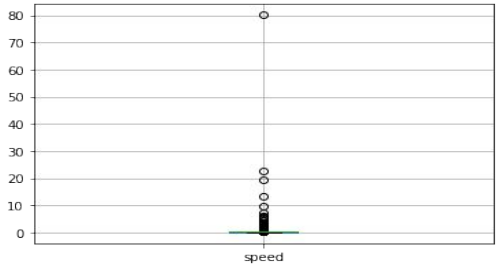
- Given dataset has neither have any null value nor any duplicate.
- In adding columns there is nothing to do with id, vendor_id, passenger_count, dropoff_datetime,store_and_fwd_flag and trip_duration, so these columns are left untouched in this section.
- In **pickup_datetime:**
    - We first separate date and time and then separate different sections of date and time, in date, month, year and hour, we do not consider minutes and seconds as they are not that much important for prediction
    - Also we added which day of week it is and them added a column is_weekend which checks whether a column is weekend or not, as it may be possible that more traffic is observed during office days.
    - Then we added a column which shows among four shift which shift of day taxi is booked for, as different shifts may have different trip_duration for same trip.
- Using all the four columns of latitude and longitude we calculate the total_distance between initial and final position using Haversine' formula.

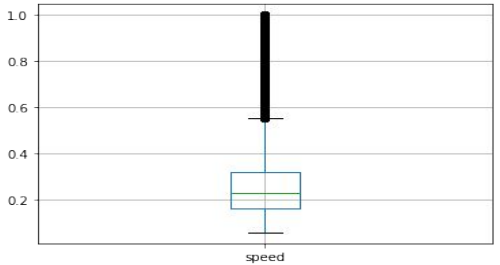
Removing Outliers and boxplot





- We have outliers presents in trip duration. So we remove it by using empirical rule.

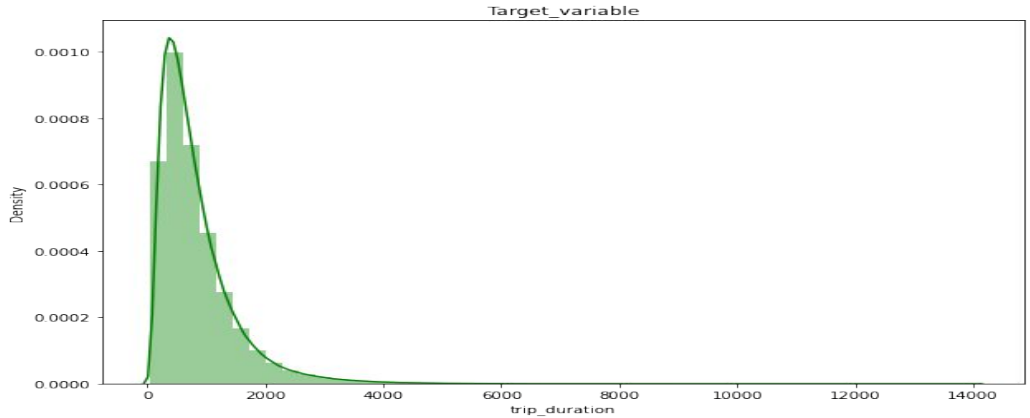


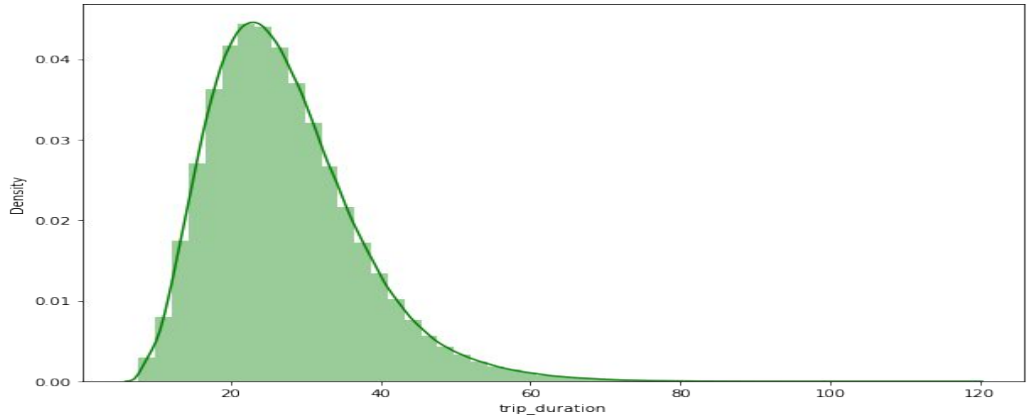


- Also we have some unnecessary data like people travel 30-40 km in 10 to 15 min, so we removed such rows from our data by adding new feature speed and removing outliers from it.

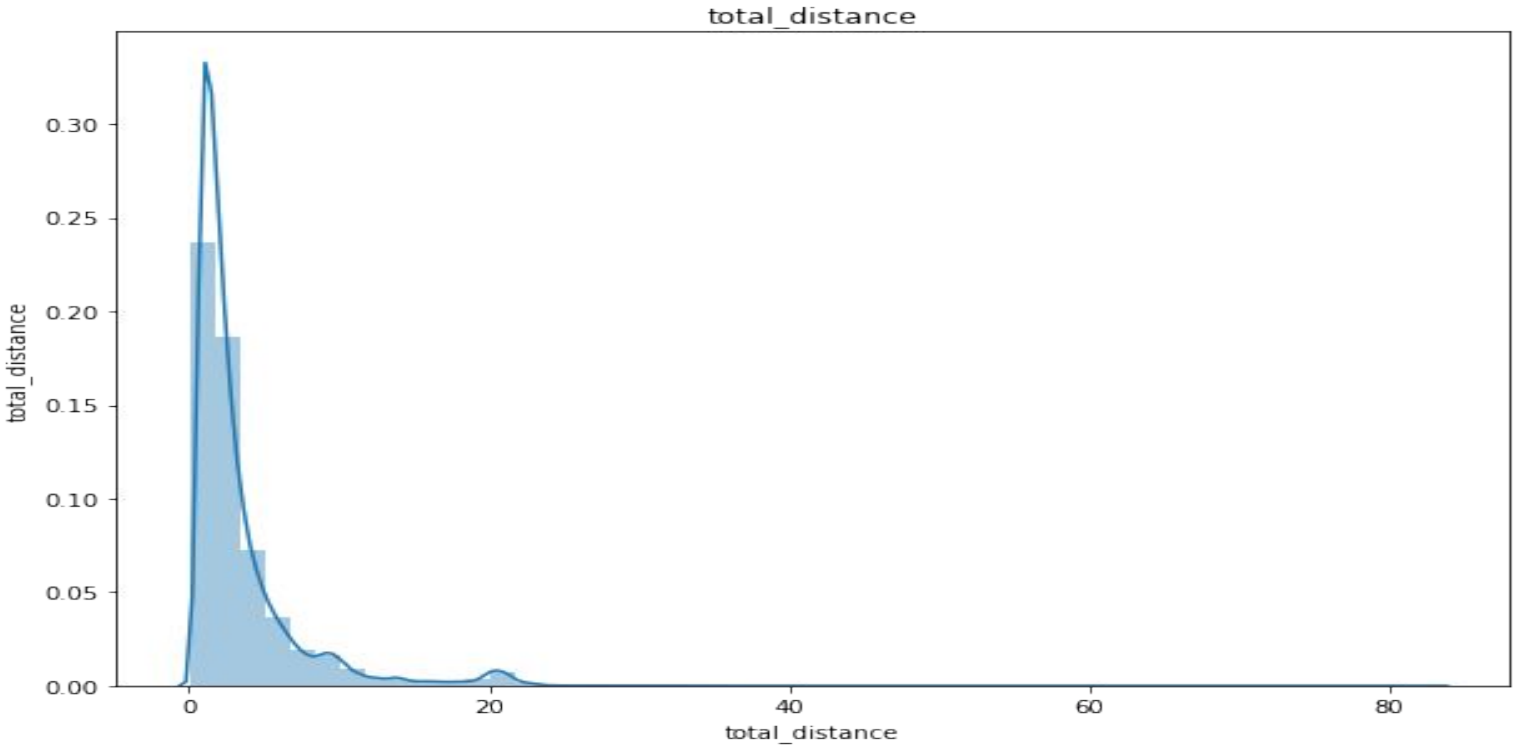
Dependent variable distribution

- Our target variable that is trip duration column are highly right skewed. So we will use square root transformation to convert the target variable into normal distribution.



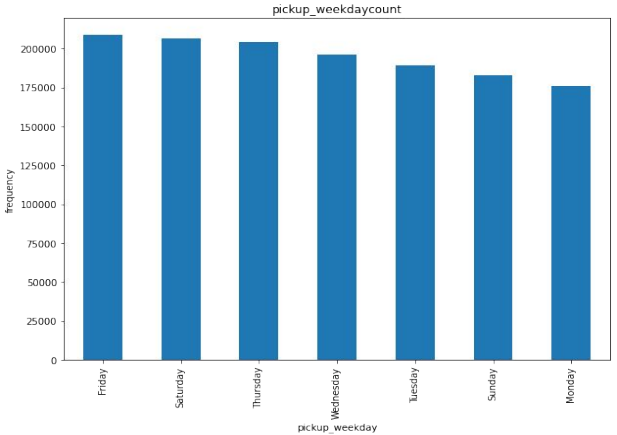


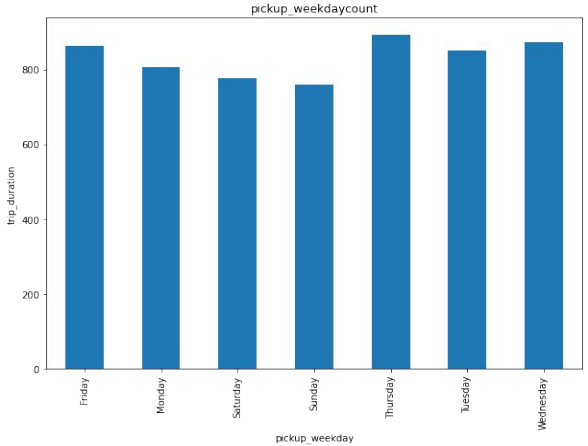
total_distance

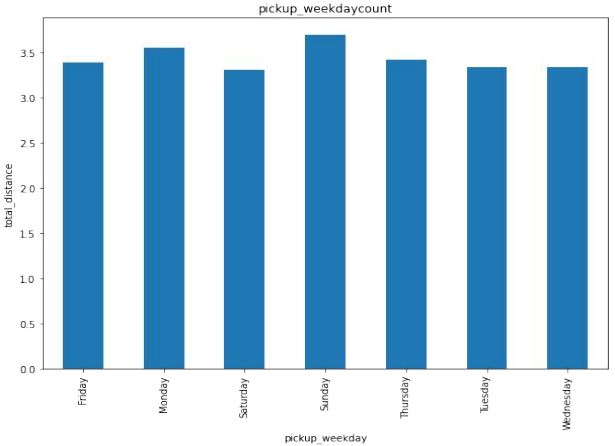


- It is highly right skewed, this implies that people booked taxi for short distance. After 20km booking data is very low.

pickup_weekday Variable

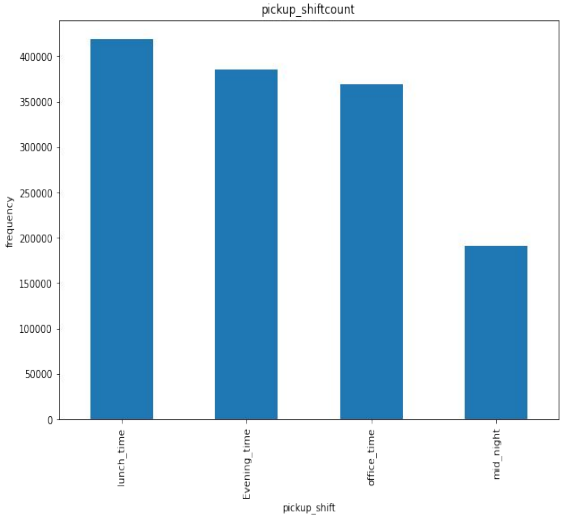


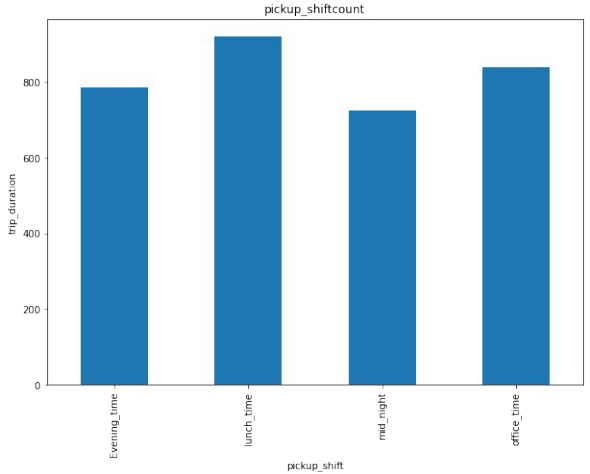


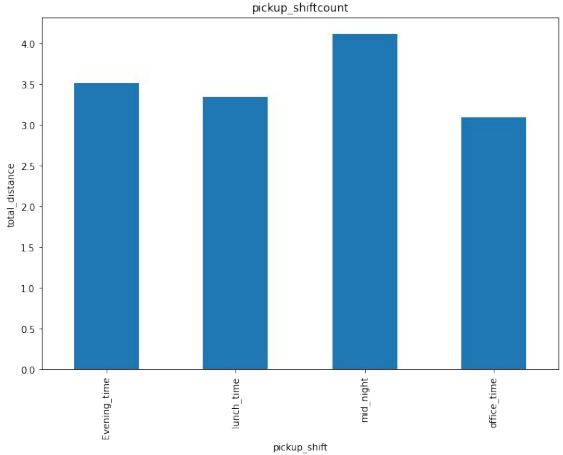


- Mostly taxi was booked by customer in friday and saturday and least booked in sunday and monday.
- Time duration is longer in Thursday and shorter in sunday.
- People travel long distance in sunday.

pickup_shift Categorical Variable

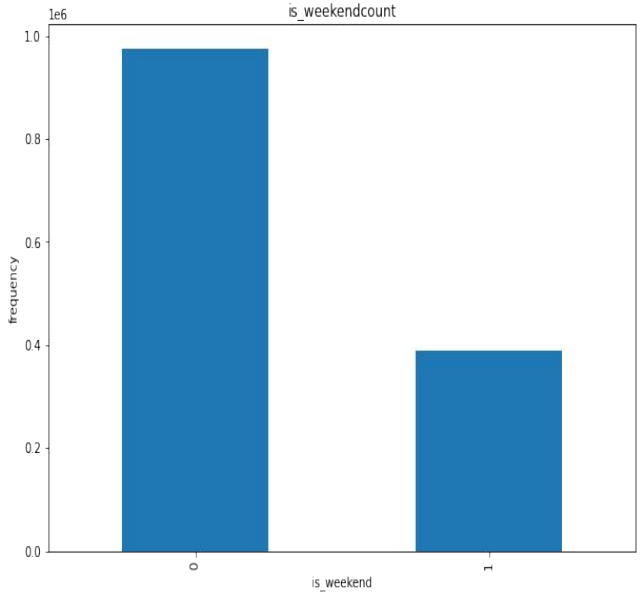


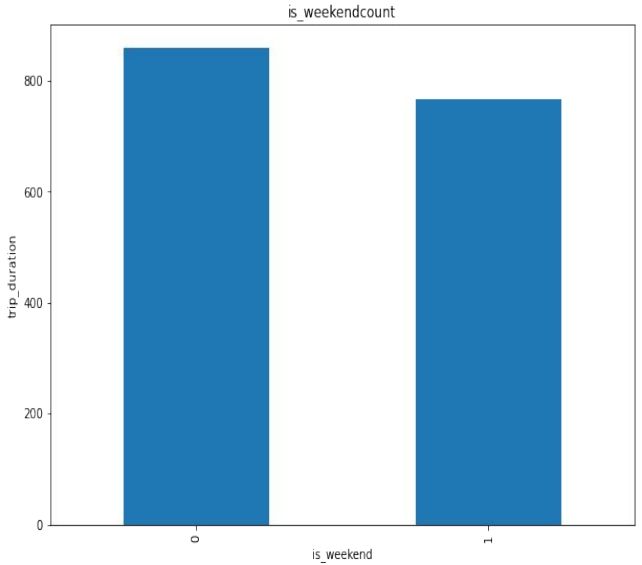


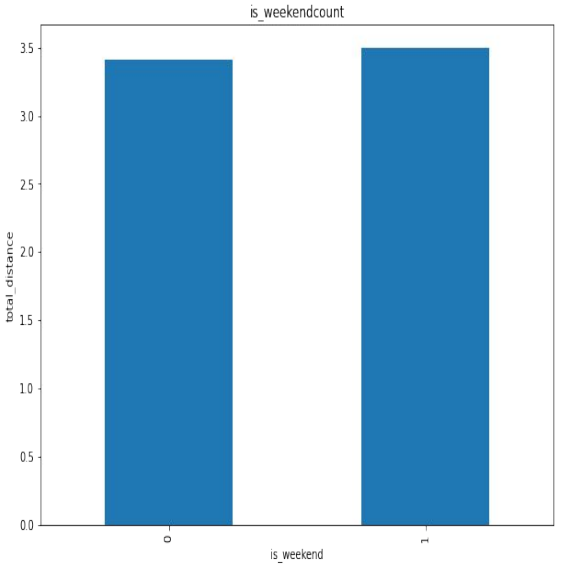


- Mostly taxi was booked during lunch time and usually trip duration is longer during lunch time. That implies there is too much traffic during lunch time
- Trip duration is also longer in office time.
- Customer prefers night time for travelling long distance.

Is_weekend Variable

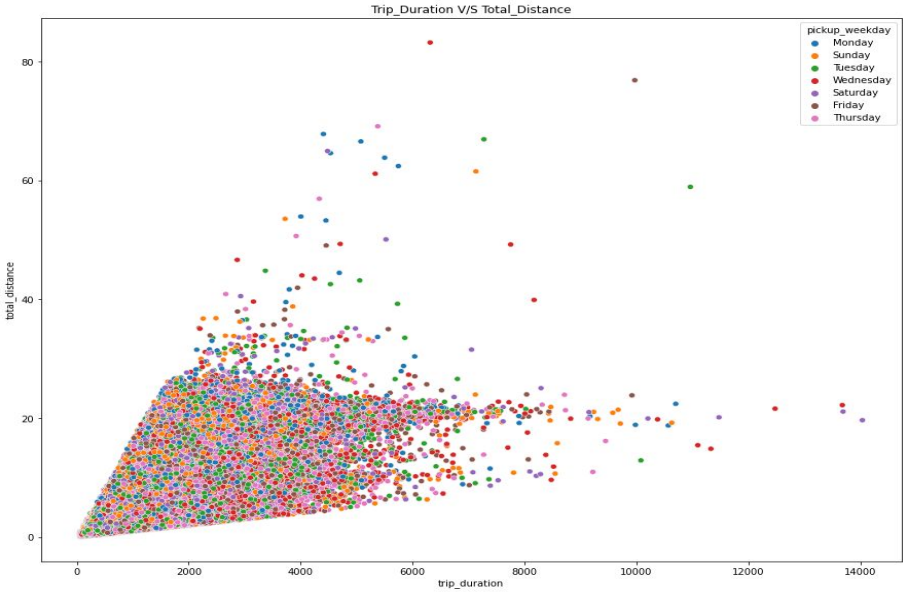


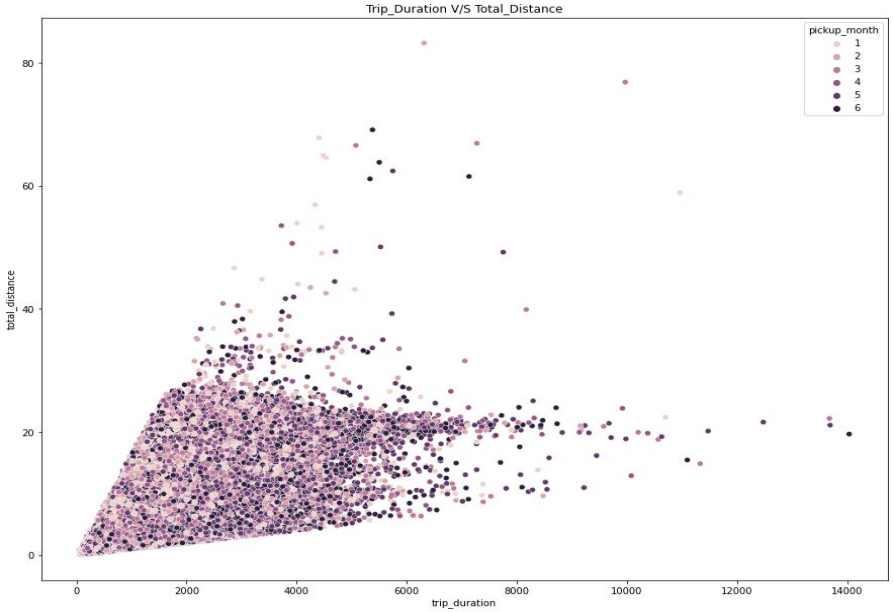




- As taxi was less booked in weekend but customer travel time and distance was increases.

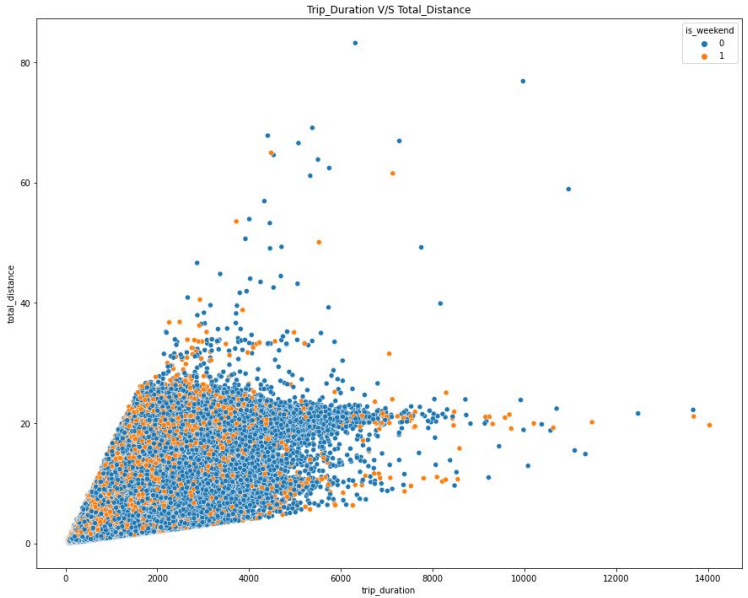
EDA

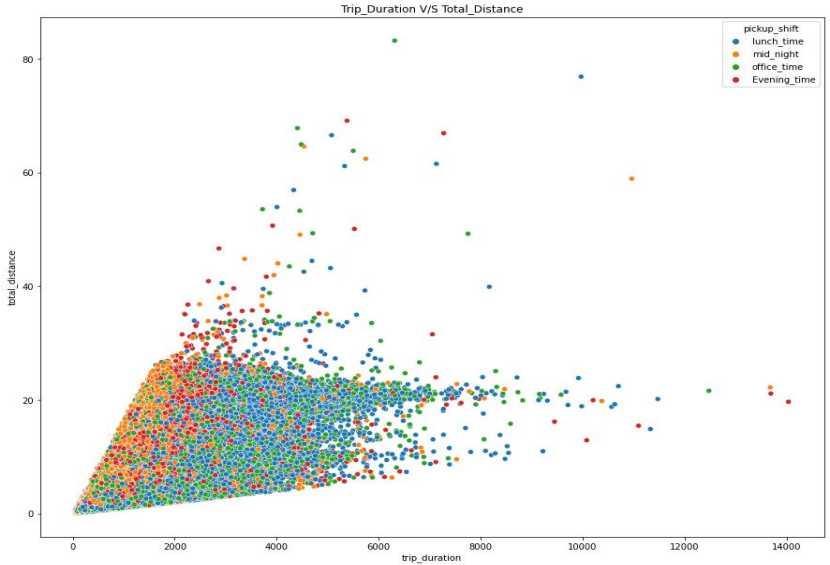




- Pickup_weekday and pickup_month do not affect the trip duration much as different hues are mixed together.

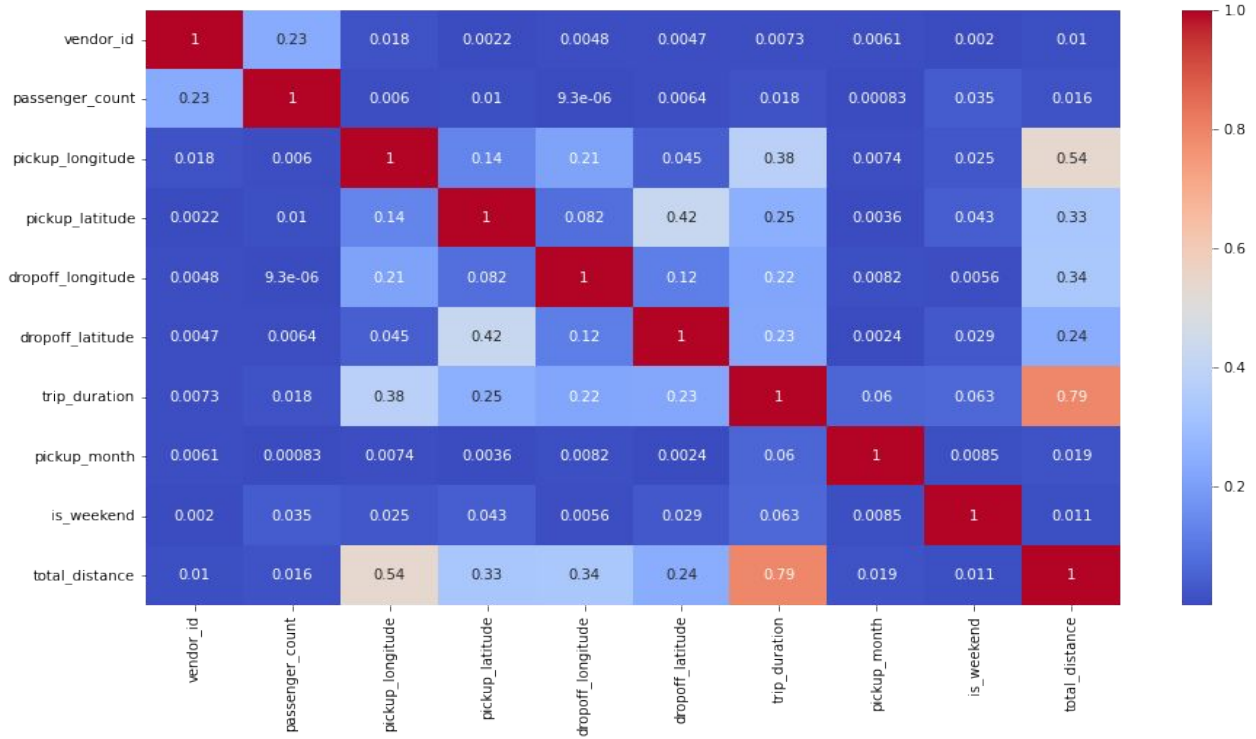
EDA(continue)





- pickup_shift and pickup_isweekend shows some boundaries through which we can separate different colours so these attributes are important
- Conclusion: First two features do not bother the trip duration much, as we note that all the colors are mixed , and there is no particular boundary which tends to separate them so we drop the first two features

Heat Map



- From here we note that there is a good correlation between trip_durationa and total distance, so we conclude that duration mainly depends on distance and not much on the path followed or on the initial and final positions.

Preparing data for Linear regression

- Linear regression model is affected by the multicollinearity, so we will remove multicollinearity using VIF
- VIF measure the multicollinearity between the independent variable. We will drop the multicollinear feature only for linear regression model. Other model like Decision tree, GBM, XGB are not affected by multicollinearity

	columns	vif_values
0	vendor_id	1.083433e+01
1	passenger_count	3.236760e+00
2	pickup_longitude	3.599453e+06
3	pickup_latitude	2.610077e+06
4	dropoff_longitude	3.228590e+06
5	dropoff_latitude	1.917796e+06
6	is_weekend	1.405791e+00
7	pickup_shift	2.464695e+00
8	total_distance	1.885683e+00

	columns	vif_values
0	vendor_id	4.627328
1	passenger_count	3.105256
2	is_weekend	1.366131
3	pickup_shift	2.169271
4	total_distance	1.708084

Preparing data for Linear regression

Task : Linear Regression

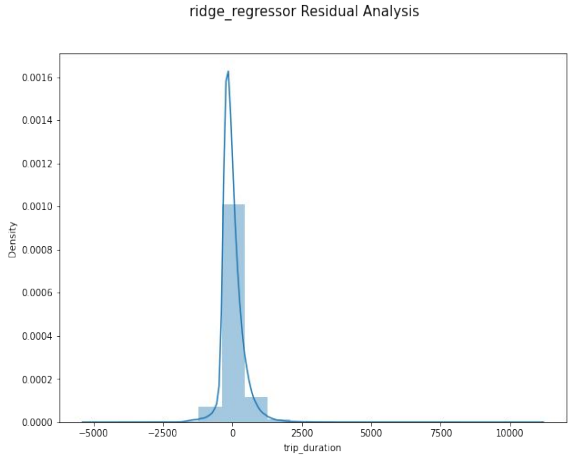
Train_set : (954998,5)

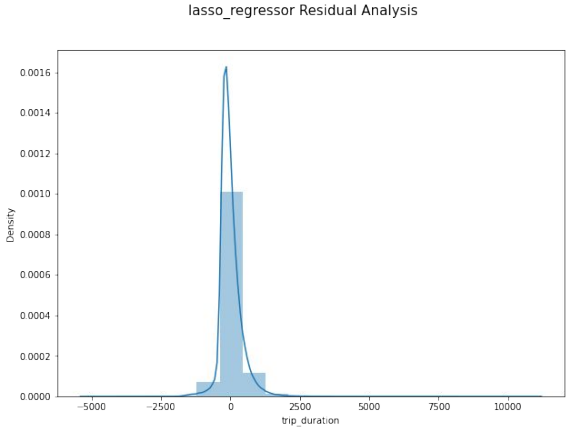
Test_set : (409285,5)

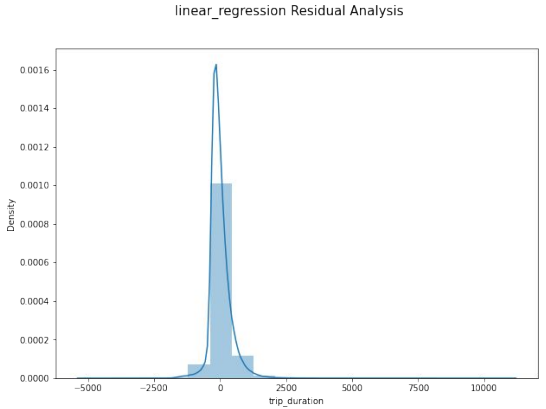
vendor_id	passenger_count	total_distance	is_weekend	pickup_shift
2	1	1.500127	0	1
1	1	1.807443	1	2
2	1	6.391944	0	3
2	1	1.487091	0	0
2	1	1.189863	1	1
...	...	...	...	...
2	4	1.226394	0	1
1	1	6.056322	1	3
2	1	7.832994	0	2
1	1	1.093735	0	1
1	1	1.135258	0	1

On Applying some basic models:

	Model_Name	Train_MSE	Test_MSE	Train_RMSE	Test_RMSE	Train_r2_score	Test_r2_score	Train_Adjusted_r2_score	Test_Adjusted_r2_score
0	linear_regression	155379.32	154694.14	394.18	393.31	0.63	0.63	0.63	0.63
1	lasso_regressor	155379.32	154694.14	394.18	393.31	0.63	0.63	0.63	0.63
2	ridge_regressor	155379.32	154694.14	394.18	393.31	0.63	0.63	0.63	0.63







Observation 1

- As seen in the above table Linear regression and regularized linear regression is not giving some great result.

- Residual distribution plot for linear regression is slightly right skewed.

Preparing data for other Models

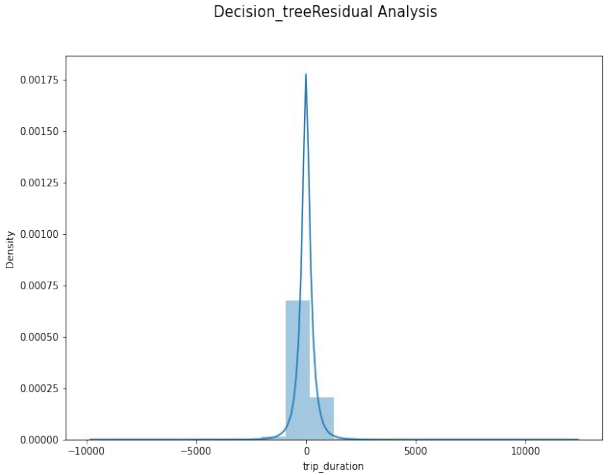
- Task : Regression
- Train_set : (954998,10)
- Test_set : (409285,10)

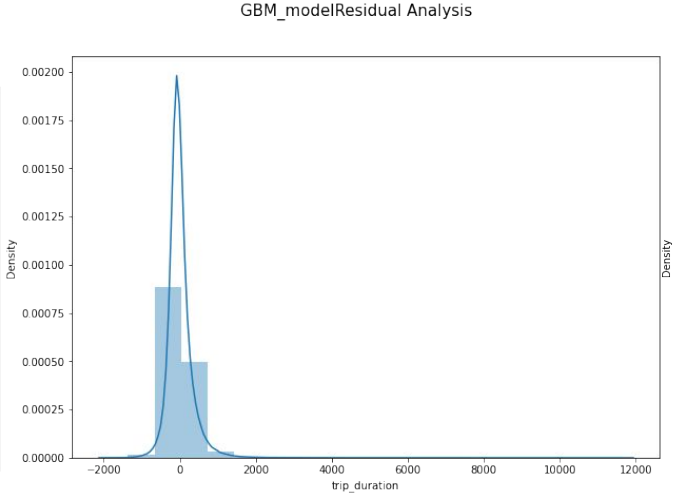
vendor_id	passenger_count	total_distance	store_and_fwd_flag	is_weekend	pickup_shift	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude
2	1	1.500127	0	0	1	-1.291232	0.711535	-1.290926	0.711494
1	1	1.807443	0	1	2	-1.291202	0.711022	-1.291535	0.710893
2	1	6.391944	0	0	3	-1.291178	0.711465	-1.291637	0.710525
2	1	1.487091	0	0	0	-1.291719	0.710698	-1.291758	0.710466
2	1	1.189863	0	1	1	-1.291073	0.711976	-1.291071	0.711789
...	...	...	...	...	...	...	...	...	...
2	4	1.226394	0	0	1	-1.291233	0.711144	-1.291455	0.711050
1	1	6.056322	0	1	3	-1.291560	0.711176	-1.291023	0.712034
2	1	7.832994	0	0	2	-1.290830	0.711550	-1.291621	0.710478
1	1	1.093735	0	0	1	-1.291231	0.711205	-1.291101	0.711346
1	1	1.135258	0	0	1	-1.291187	0.711776	-1.291069	0.711930

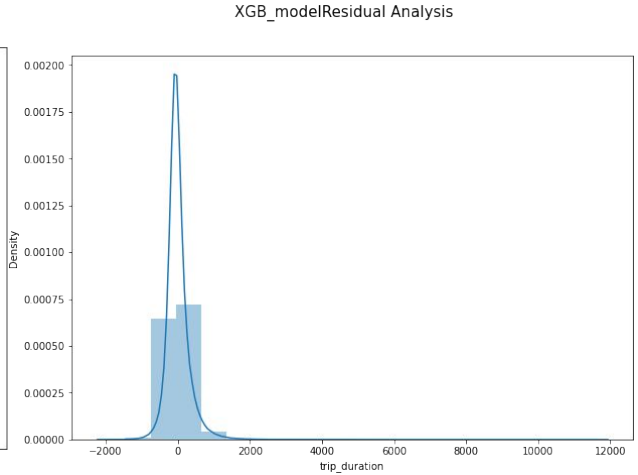
ows × 10 columns

Model Selection

	Model_Name	Train_MSE	Test_MSE	Train_RMSE	Test_RMSE	Train_r2_score	Test_r2_score	Train_Adjusted_r2_score	Test_Adjusted_r2_score
0	Decision_tree	0.00	171586.50	0.00	414.23	1.00	0.59	1.00	0.59
1	GBM_model	105790.84	106068.94	325.26	325.68	0.75	0.75	0.75	0.75
2	XGB_model	104994.05	105347.27	324.03	324.57	0.75	0.75	0.75	0.75







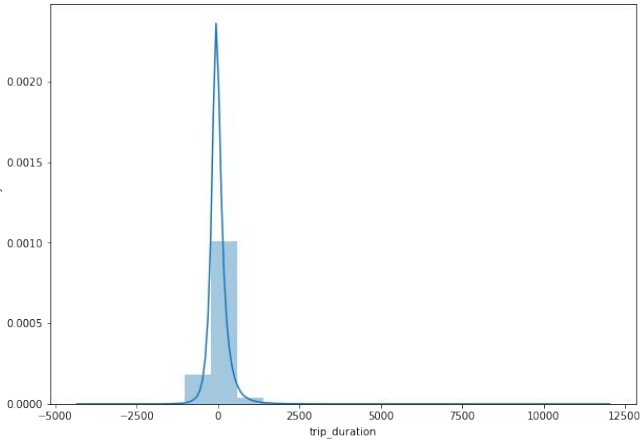
Observation 2

- From above table, we clearly observe that decision tree is overfitted. To overcome this problem we will tune the hyperparameter of decision tree.
- XGBRegressor and GBM Regression are performing well with same with r2 score 0.75, but we can improve their performance by hyperparameter tuning.
- Residuals are normally distributed for decision tree, while for XGB and GBM are right skewed.

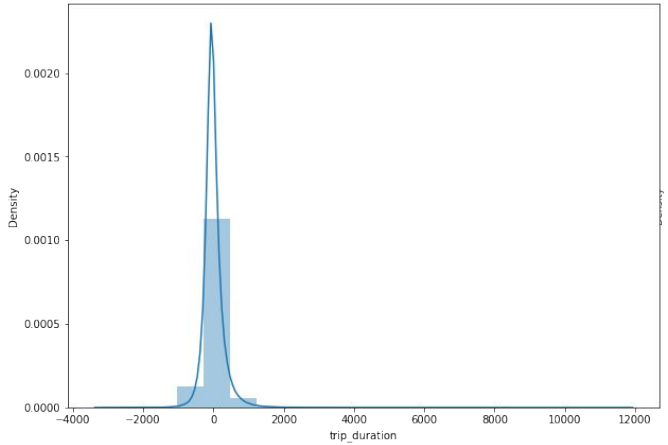
Model validation and tuning

Model_Name	Train_MSE	Test_MSE	Train_RMSE	Test_RMSE	Train_r2_score	Test_r2_score	Train_Adjusted_r2_score	Test_Adjusted_r2_score	best_estimat
Decesion_tree	94189.12	98469.56	306.90	313.80	0.78	0.76	0.78	0.76	DecisionTreeRegressor(max_depth=' min_impurity_decrease=0.1 min_samples_split=6(
XGBRegressor	75816.54	87956.66	275.35	296.57	0.82	0.79	0.82	0.79	XGBRegressor(learning_rate=0.5, max_depth= subsample=0
GBMRegressor	77060.63	86016.65	277.60	293.29	0.82	0.79	0.82	0.79	DecisionTreeRegressor(criterion='friedman_ms max_depth=9

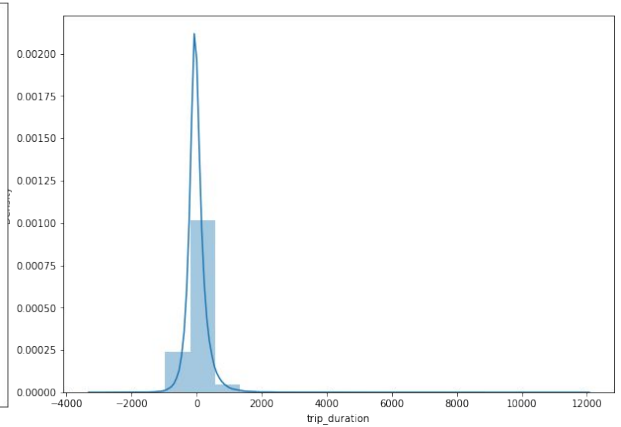
GBMRegressor Residual Analysis



XGBRegressor Residual Analysis



Decesion_tree Residual Analysis



Observation 3

- On hyperparameter tuning decision tree overcome with the overfitting problem with train r2 score is 0.78 and test r2 score is 0.76. The best hyperparameter estimator is, 'max_depth': 15, 'min_impurity_decrease': 0.1, 'min_samples_split': 600.
- GBM and XGB have nearly same performance.
- Best hyperparameter estimator for XGB is, 'learning_rate': 0.5, 'max_depth': 7, 'subsample': 0.9
- Best hyperparameter estimator for GBM is, 'n_estimators': 90, 'max_depth': 9
- Gradient Boosting Regressor is performing slightly better than other two model.
- Finally we select GB regressor having R2 score of nearly 80 for test data.

Challenges

- Big dataset, take very long time in computation.
- Some Models like Randomforest we are unable to perform on this dataset, as it takes too much time.
- Dealing with outliers is major challenge for us.
- The data mostly have information about shorter trips, so our model is able to make correct predictions for a shorter trip or for a average duration trip, but for very large trips the predictions are not very much accurate
- We do not used black box model, as we want some transparency to judge what all reasons are affecting our trip_duration, so we limit ourselves to grey box models.

Conclusion

- Total_distance, pickup_shift, is_weekend feature were found to be most relevant for predicting the trip duration for NYC taxi.
- From pickup_shift it is clearly visible that, at lunch time taxi takes long time to cover short distance, whereas at midnight taxi takes short time to cover long distance. This implies people face lots off traffic during lunch time.
- In weekends people usually book a taxi for a longer trip.
- Most of the taxis do not have store and forward flag but for long duration a taxi with store and forward flag is preferred.
- Our dependent column taxi trip mostly have data of shorter trips.

Conclusion

- Linear regression model does not perform good in this dataset as very few dependent variable is strongly correlated to independent variable. The XGB and GBM provide substantial improvement in predicting the trip duration. The root mean square error is less than 300 seconds and r2 and adjusted r2 score is 0.79.
- So we used XGboost and Gradient boost model for prediction,. This model can also improve by finer tuning of hyperparameters.

Thank you