Capstone Project – 2

Supervised ML - Regression

NYC Taxi Trip Time Prediction

Team Members

Prerit Tyagi

Saurabh Waghmare

Shivraj Y Saude

Presentation Outline:

- Problem Statement
- Introduction
- Exploring the dataset
- Methodology
- EDA and Data Processing
- Optimal Model for testing & Evolution
- Challenges
- Scope of Improvement
- Conclusion

Problem Statement:

Your task is to build a model that predicts the total ride duration of taxi trips in New York City. Your 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.

Introduction:

The data is the travel information for the New York taxi. The prediction is using the regression method to predict the trip duration depending on the given variables. The variables contains the locations of pickup and drop-off presenting with latitude and longitude, pickup date/time, number of passenger etc. The design of the learning algorithm includes the preprocess of feature explanation and data selection, modeling and validation. To improve the prediction, we have done several test for modeling and feature extraction.

Exploring the Dataset

Data Summary:

- **Data Set Name:-** NYC Taxi Data.csv (The training dataset we have.)

- **Statistics:-**
  - Rows - 1458644
  - Features - 11 (Including Target)
  - Target - Trip Duration Important

- **Column:-** ‘id’, 'vendor_id', 'pickup_datetime', 'dropoff_datetime', 'passenger_count', 'pickup_longitude', 'pickup_latitude', 'dropoff_longitude', 'dropoff_latitude', 'store_and_fwd_flag', 'trip_duration’.

Data Menu:

- **Independent Variables: -**
  - 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.
- **Target Variable:–**
  - trip_duration — duration of the trip in seconds

Data Pipeline:

AI

- **Data Wrangling :**
  - Check for null values
  - Removed outlier
- **EDA :**
  - Exploratory Data Analysis of features
  - Target to extract a piece of information

- **Modelling:**
  - Encode categorical variables
  - Create new features from existing
  - Standardize data
  - Split data and train the model
  - Evaluate the model against the validation set with MSE, RMSE, R2 score, etc error metrics

- **Tools Used:**
  - Google Colab Research (Python)

Attribute Information: Dtype and Null Values

```
#Attribute information

df.info()
```

```
<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
```

```
#checking missing values

df.isnull().sum()
```

```
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
```

Attribute Information: Unique Values

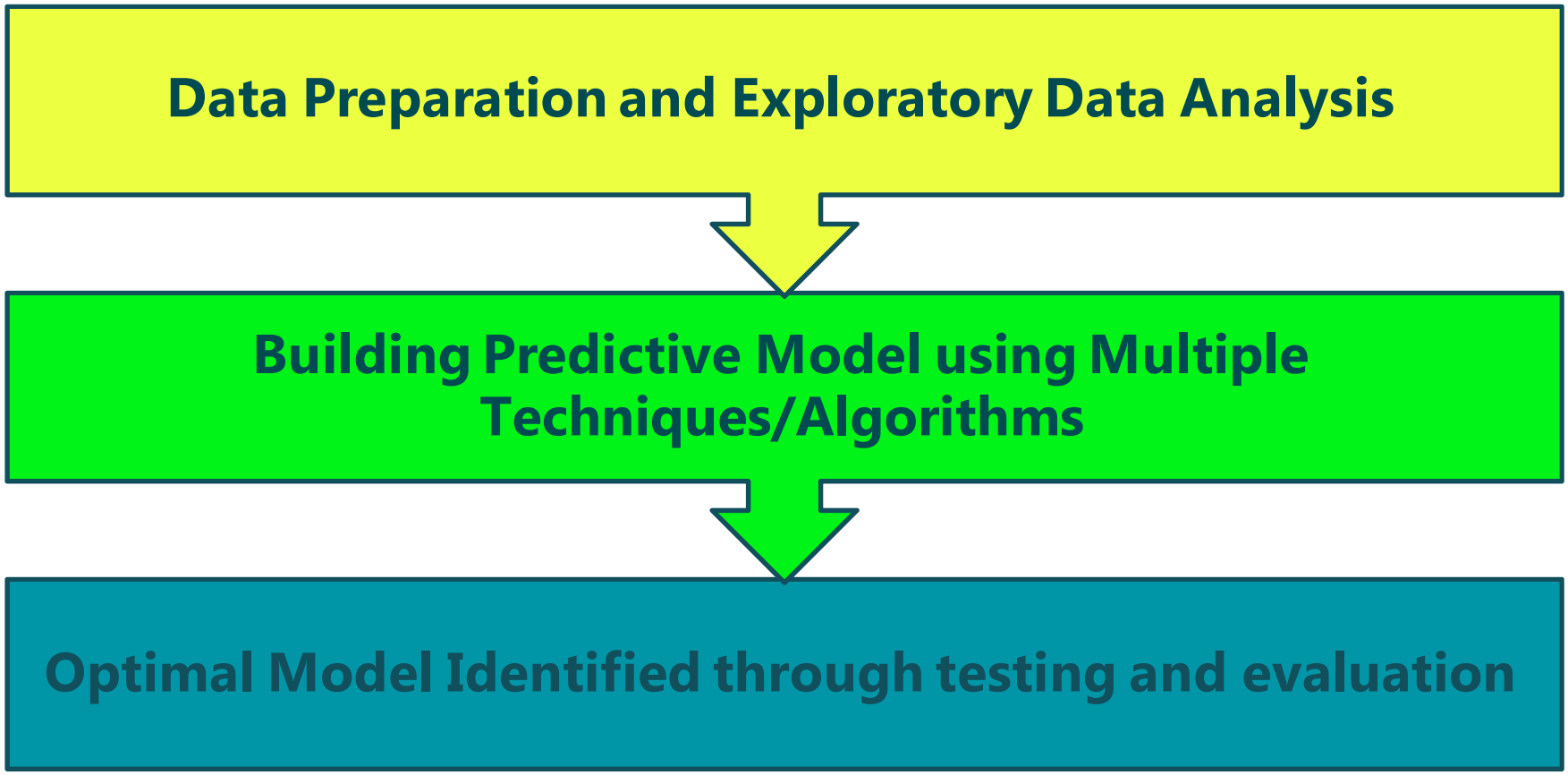
```
# Let us check for unique values of all columns.

print(df.nunique().sort_values())
```

```
vendor_id                2
store_and_fwd_flag       2
passenger_count         10
trip_duration         7417
pickup_longitude     23047
dropoff_longitude    33821
pickup_latitude      45245
dropoff_latitude     62519
pickup_datetime    1380222
dropoff_datetime   1380377
id                 1458644
dtype: int64
```

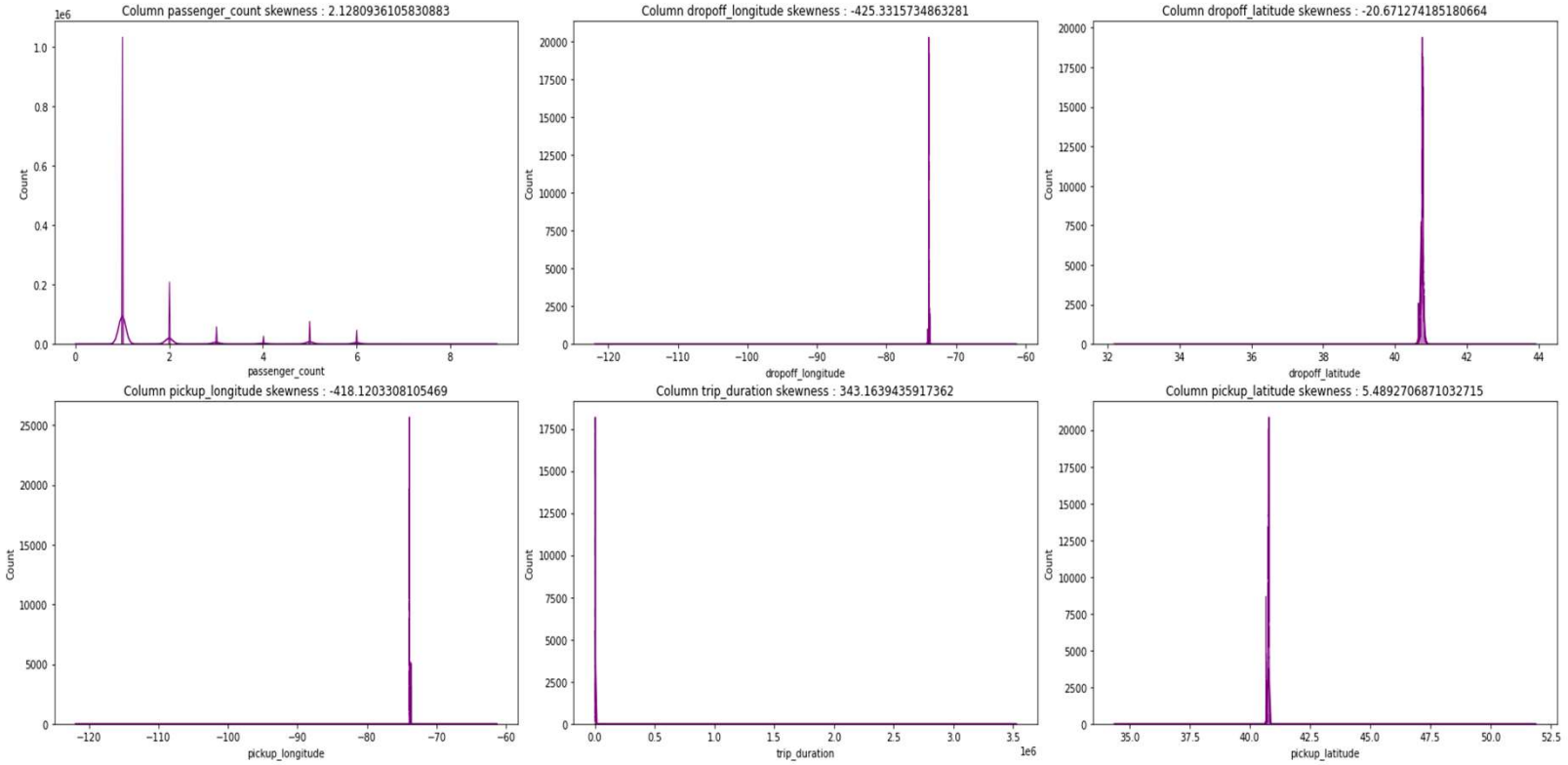
METHODOLOGY

Approach:


Data Preparation and Exploratory Data Analysis
Building Predictive Model using Multiple Techniques/Algorithms
Optimal Model Identified through testing and evaluation

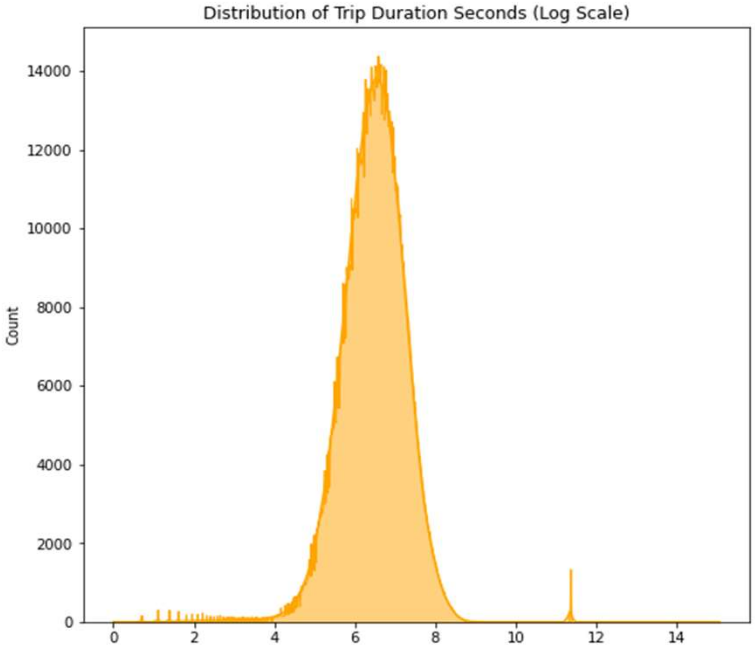
Distribution of Features:

- **Extreme skewed distribution**

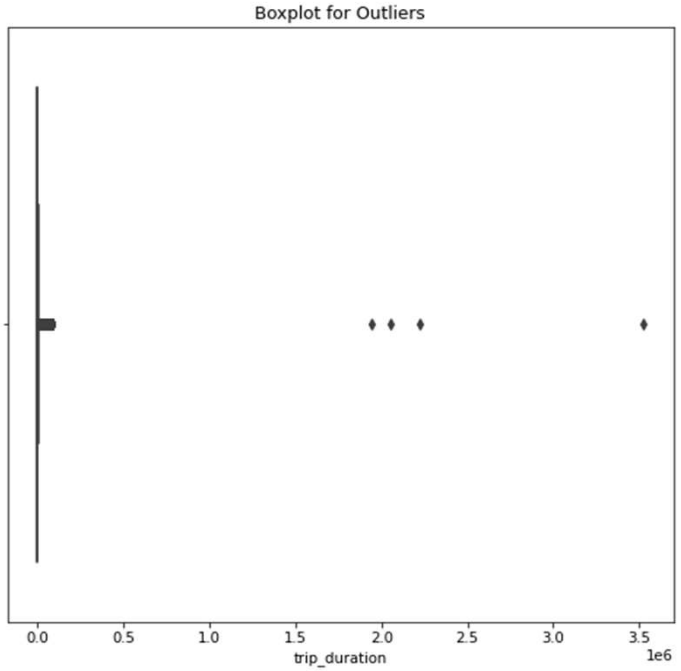


Target Distribution (Log Scale)

- Most of the trips between ~54 sec (exp 4) and 82 mins (exp 8)



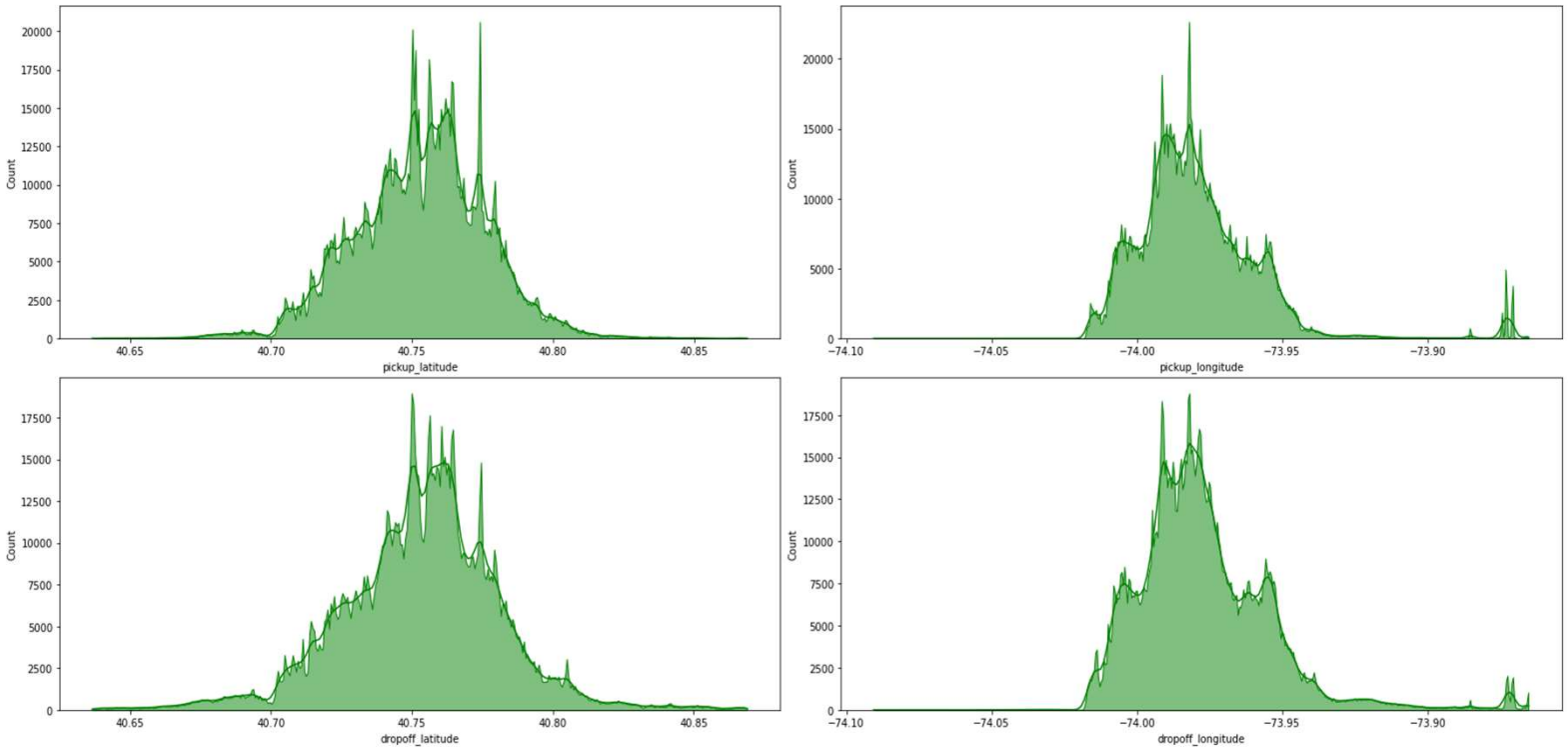
Normal Distribution in Log Scale



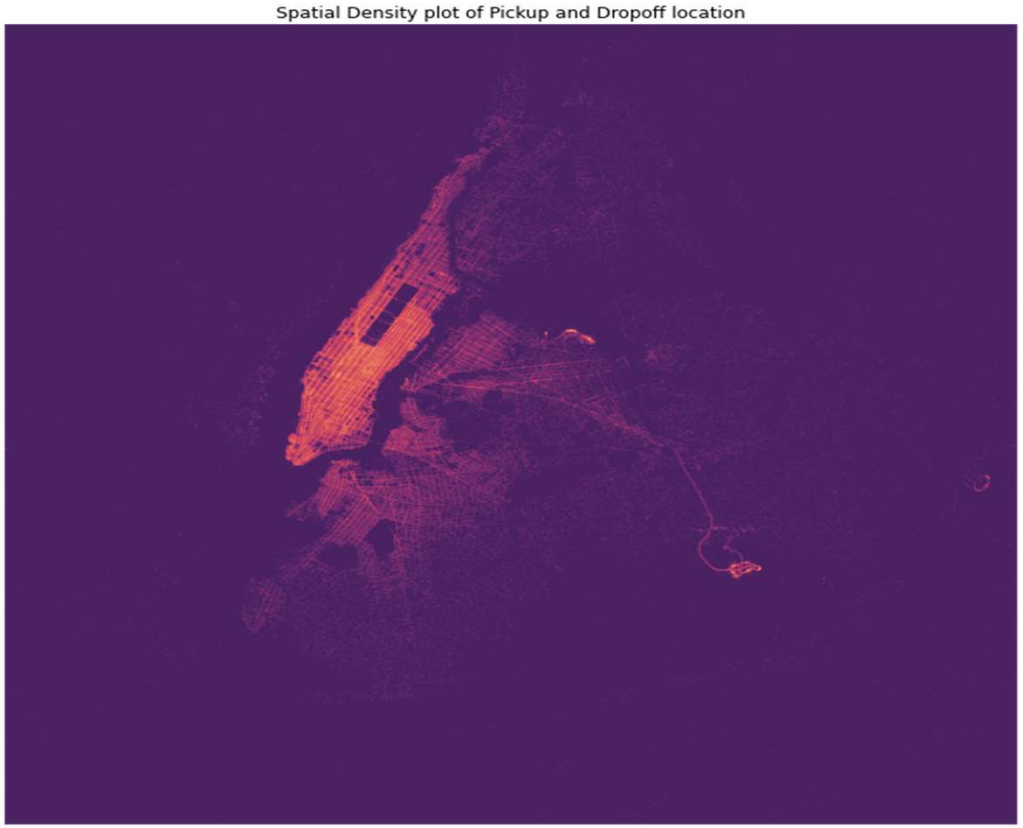
Clear outliers visible

Distribution of Co-ordinates:

- Latitude distributes within 40 to 41
- Longitude distributes within -74 to -73

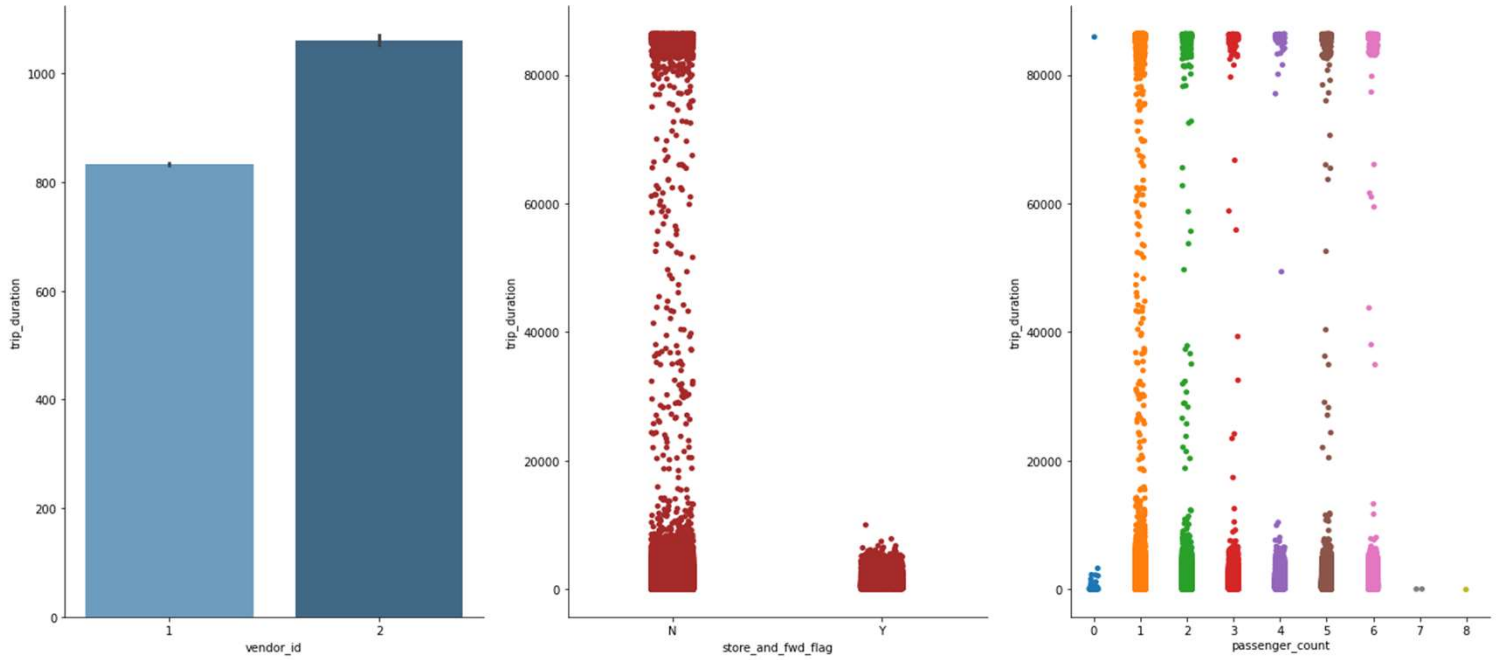


Spatial Density Graph of Location:



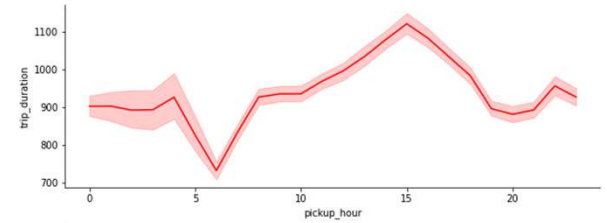
Bivariate Analysis on Target:

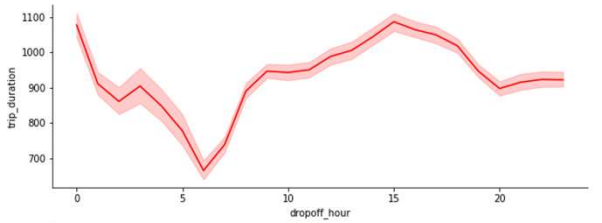
- ❑ Vendor 2 seems to prefer longer trips.
- ❑ Most of the ride has connectivity to the server.
- ❑ Several Rides with 0 Passenger.

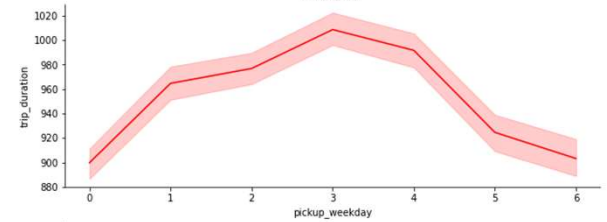


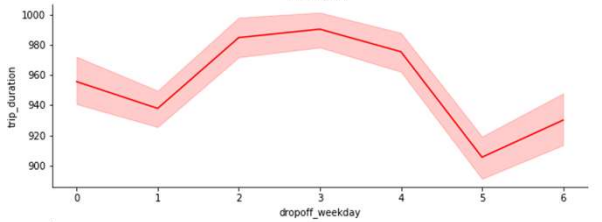
Bivariate Analysis of against Datetime:

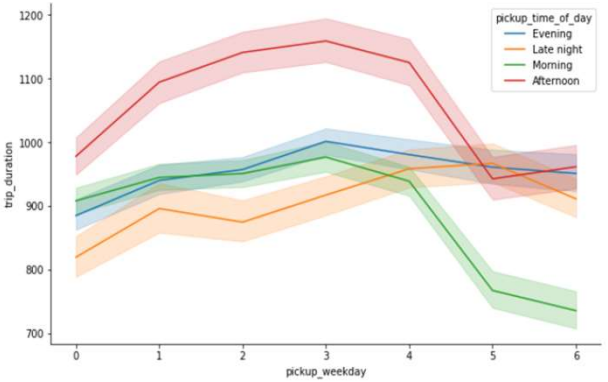
- High requests from middle of day.
- Highest duration on middle of weeks.
- With more distance travelled, trip duration on midnight is relatively low

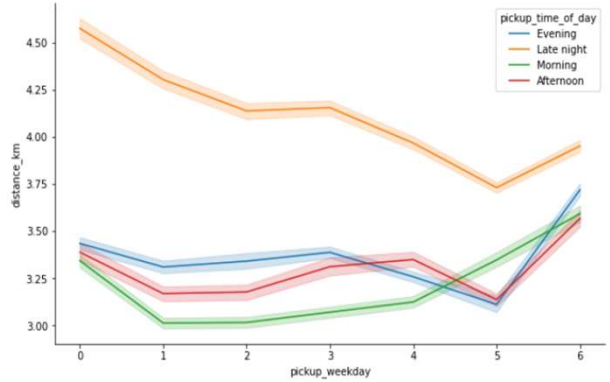




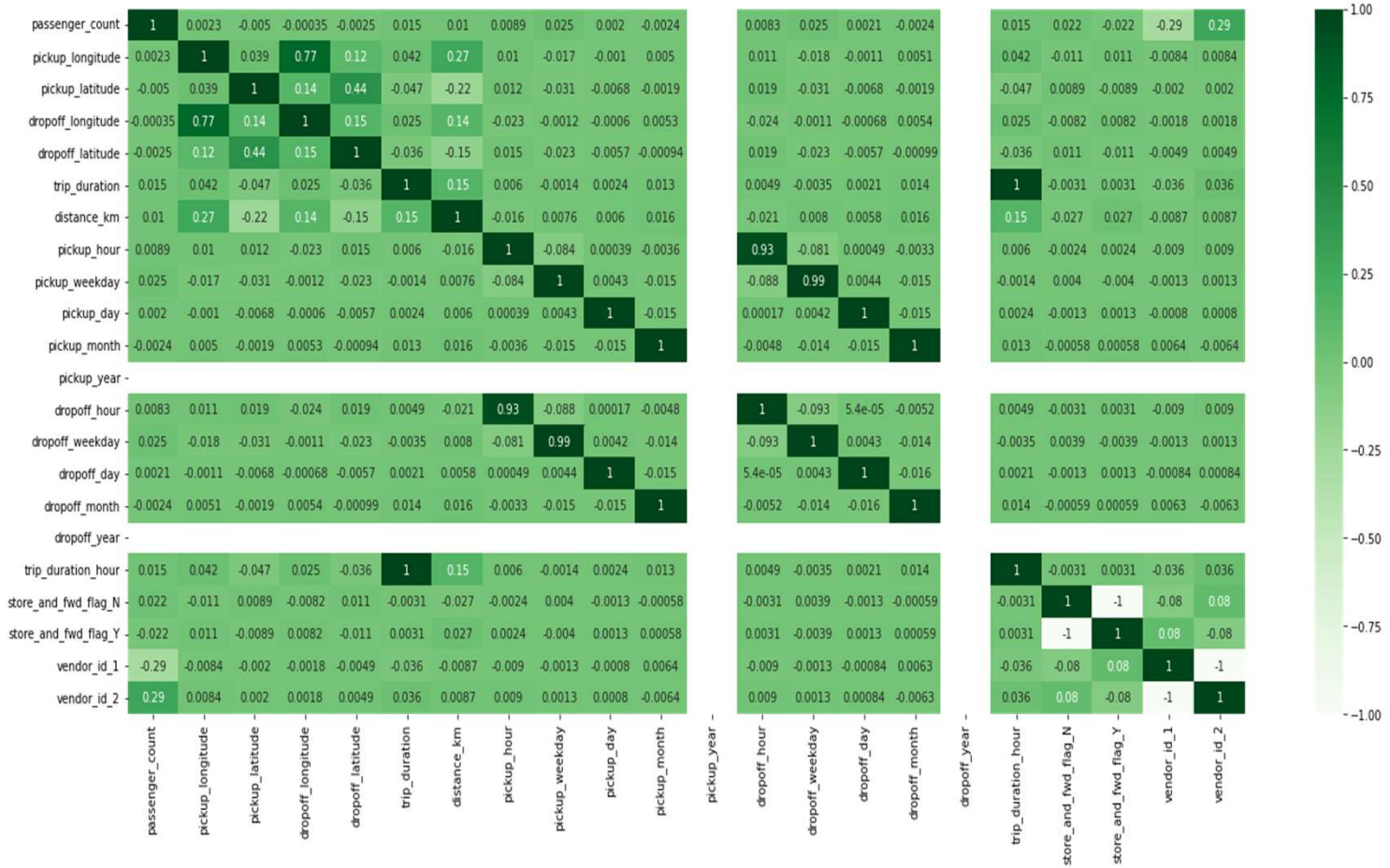






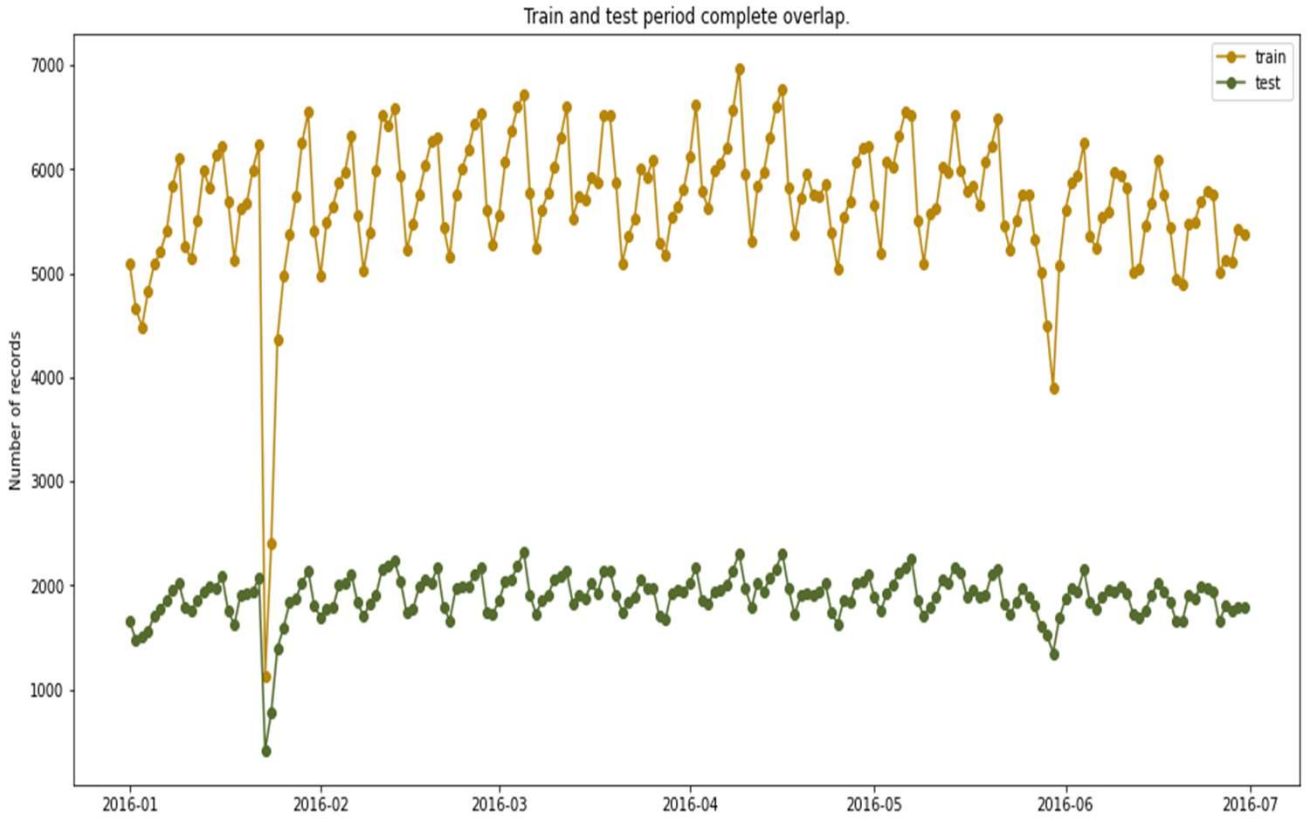


Correlation Matrix:



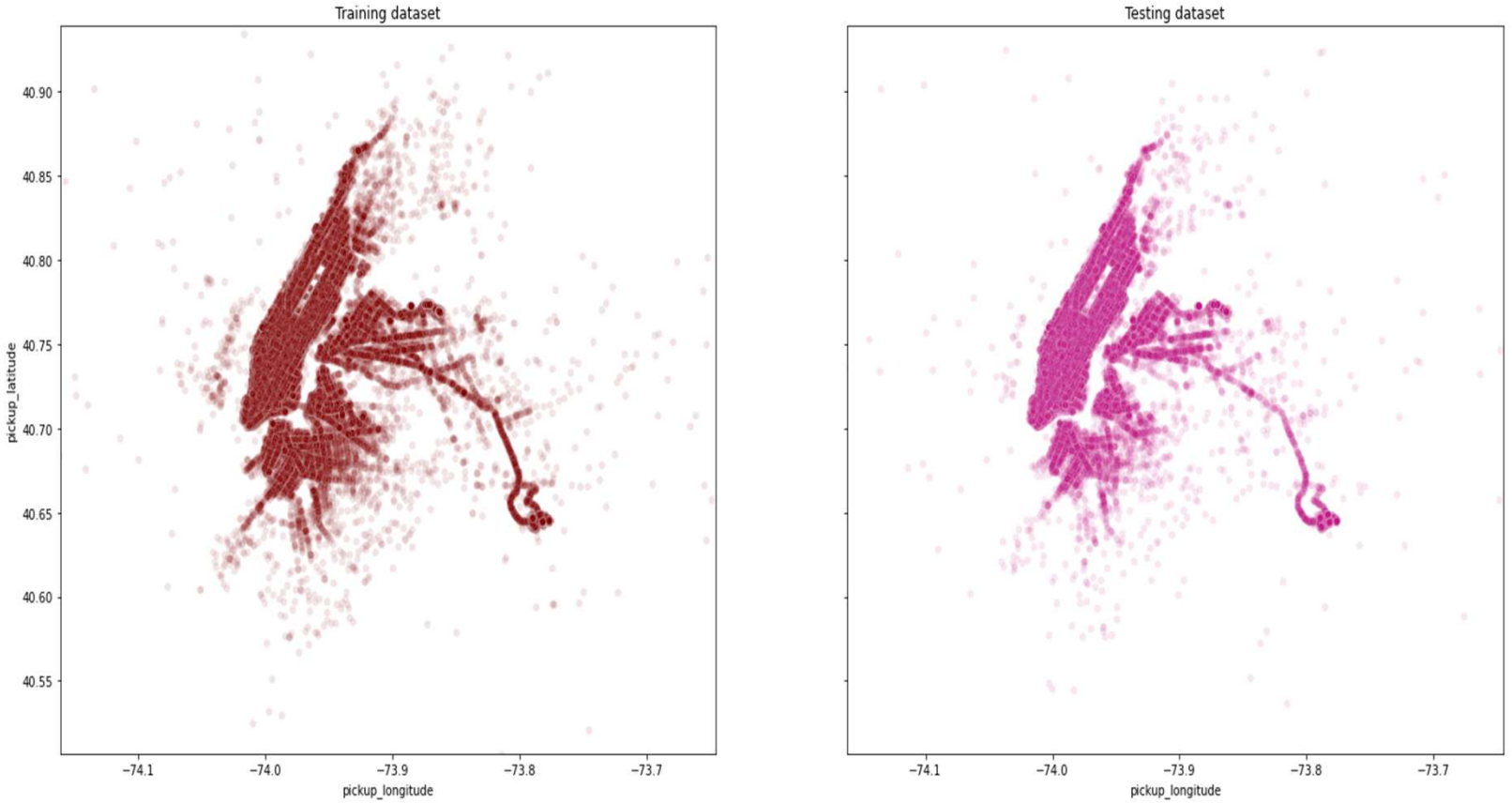
- **The highest correlation for the target is with distance km i.e. -0.15**

Comparison of Train-Test:



- **Train Set: 1039279**
- **Test Set: 346427**
- **Validation Set: 72932**

Scatterplot of Train - Test Data:



Model Selection:

- ***Best r2 score ~0.7756 Obtained on LightBGM***

- **Linear Regression and Lasso Ridge returning score with 0.30 – 0.34 range.**
- **Decision Tree Regressor is slightly better with avg training score of ~0.45.**
- **Random Forest outperforms all previous scores, set a new best score of ~0.80 but at an extensive computational cost.**
- **LightGBM as expected gives quite similar or better performance with a score of 0.89 and 0.77 on the training and testing sets respectively.**

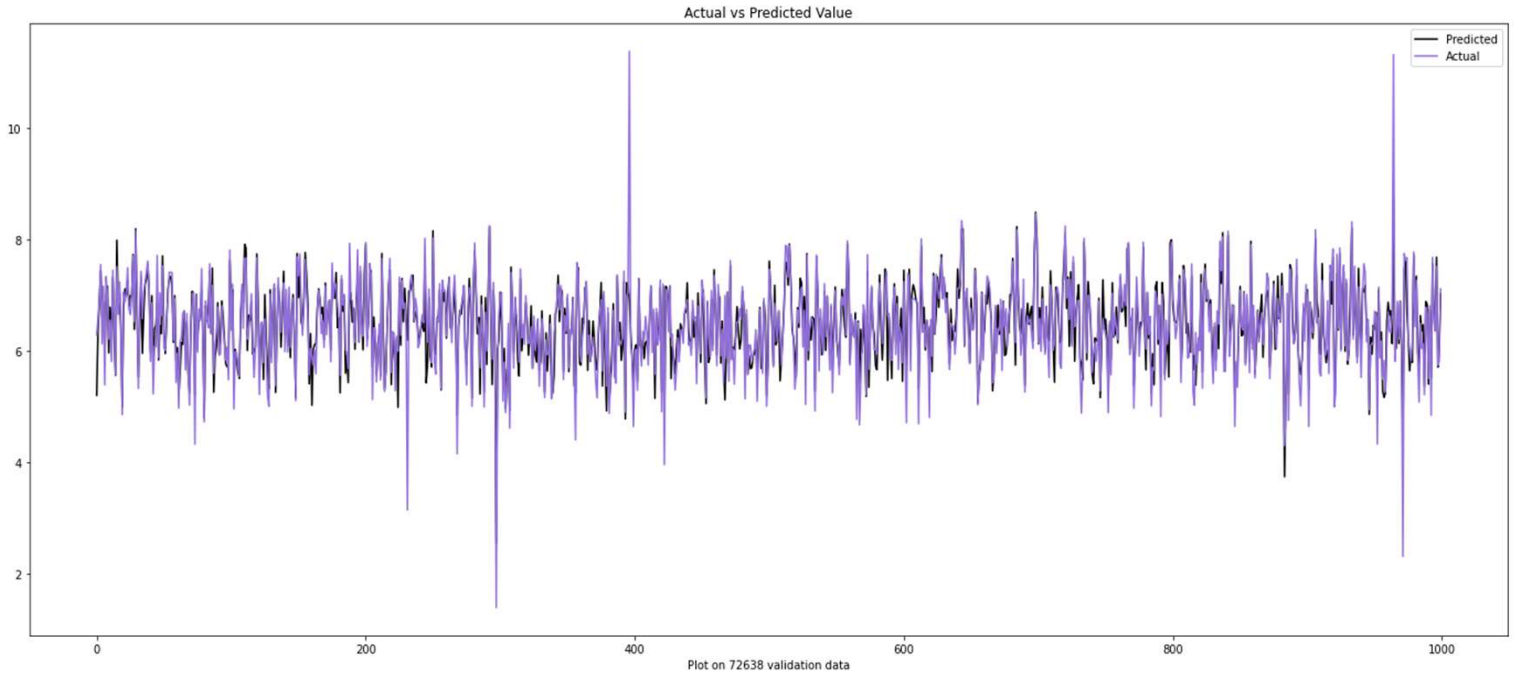
Model Evaluation:

- Best r2 score ~0.7756 Obtained on LightGBM.
- Best rmse score~0.3721 Obtained on LightGBM

	model	best_score_on_train	r2_score_on_test	adjusted_r2_on_test	mse_on_test	rmse_on_test	best_params
0	linear_regression	0.343620	0.326096	0.326072	0.416147	0.645095	{}
1	lasso	0.347218	0.323821	0.323797	0.417552	0.646183	{'max_iter': 1000, 'alpha': 0.01}
2	ridge	0.343620	0.326096	0.326073	0.416147	0.645095	{'alpha': 1}
3	decision_tree	0.467710	0.470893	0.470875	0.326733	0.571605	{'splitter': 'best', 'criterion': 'squared_error'}
4	random_forest	0.830735	0.744038	0.744029	0.158061	0.397569	{'bootstrap': True, 'max_features': 'auto', 'max_depth': 50, 'min_samples_leaf': 10, 'min_samples_split': 15, 'n_estimators': 50}
5	lightgbm	0.891685	0.775582	0.775574	0.138582	0.372266	{'max_depth': 50, 'n_estimators': 1000, 'num_leaves': 500}

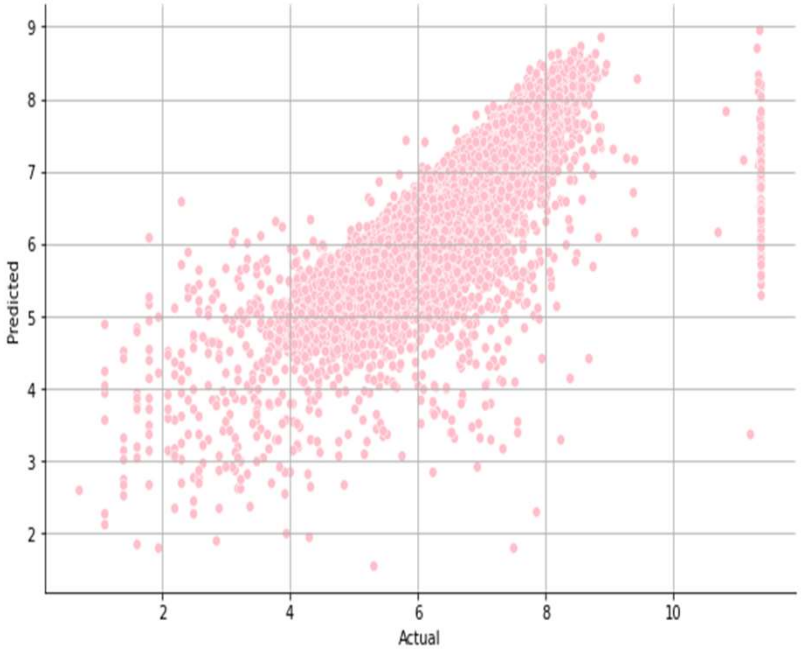
- Best parameter on LightGBM is max_depth 50 n_estimators 1000 num_leaves 500.
- The model tends to overfit with a large number of estimators or large sample splits.

Actual vs Predicted Value:

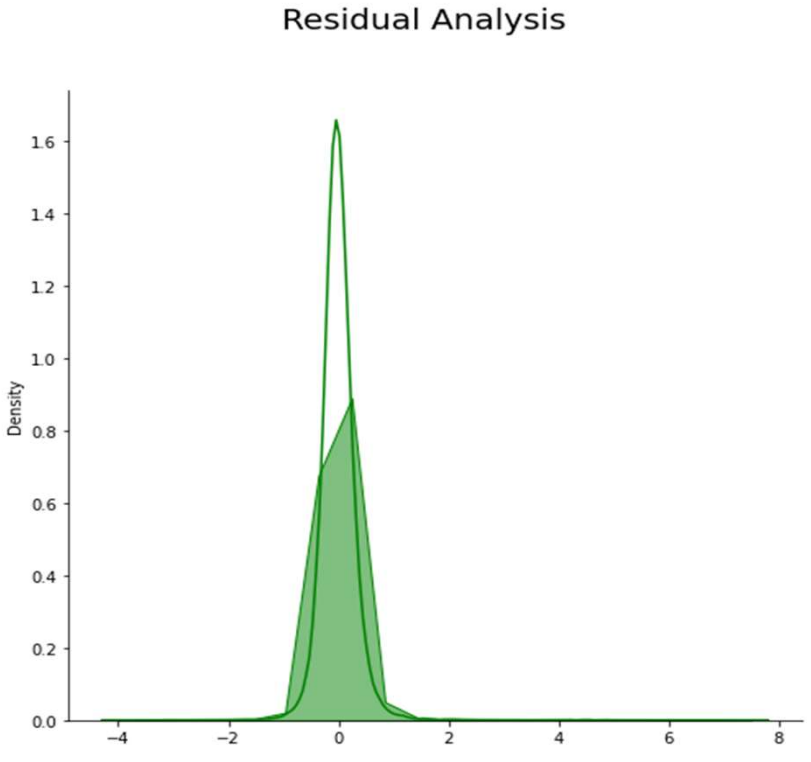


- Plotted from the predictions of 72932 validation data.

Model Evaluation:

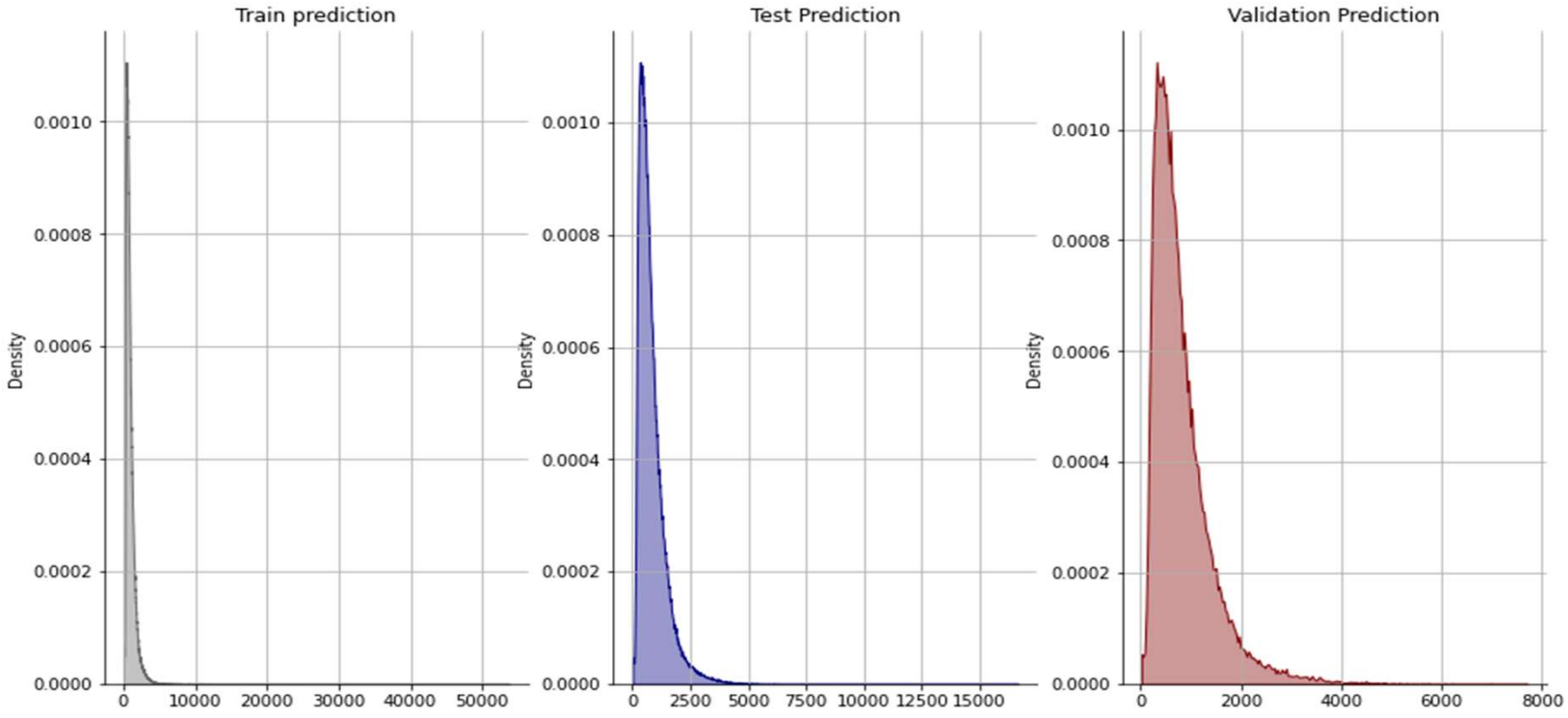


Plots with predicted Values



Residual Analysis

Histplot for density of predictions from all datasets:



Limitation:

- Real world data can be and is a messy one, it's also a no exception in our dataset. In our case, data as it was very raw, and we needed to clean it first to move further. It has many anomalies we noticed about the time duration and distance travelled, though we've removed it but still many more is present there, we live it as it is to add a bit noise.
- We don't have proper information of traffic and season, that'd be a great addition in this dataset. Also, the rating of cab or driver would also do great, people normally provide better rating with clean driving and punctuality in rush hours.

Challenges:

- With over 14 lakhs row presents, handling a huge chunk of data is a challenge itself. Also, the memory usage of this pile of data is very high so there's a chance of the system runs out of memory during model training or some advanced plotting.
- The computation time and cost both are relatively high.
- The dataset contains many obvious outliers and noise.

Scope of Improvement:

- We can add more features to our dataset, like ratings, driver's ability on some scale or experience (experienced driver is normally better at driving than newbies), seasons information or can feed live traffic data. Also, more extensive EDA is required with help of external datasets.

- Try out XGBoost model, done hyper parameter tuning over XGBoost model. XGBoost always gives more importance to functional space when reducing the cost of a model while Random Forest tries to give more preferences to hyperparameters to optimize the model. Also we need to perform PCA to reduce dimensions.

Conclusion:

- Observed which taxi service provider is most Frequently used by New Yorkers.
- Found out few trips which were of duration 528 Hours to 972 Hours, possibly Outliers.
- Passenger count Analysis showed us that there were few trips with Zero Passengers and One trip with 7,8 and 9 passengers.
- The monthly trip analysis gives us an insight of Months – March and April marking
  the highest number of Trips while January marking the lowest, possibly due to Snowfall.
- Taxi giants such as UBER and OLA can use the same data for analyzing the trends that vary throughout the day in the city. This not only helps in better transport analysis but also helps the concerned authorities in planning traffic control and monitoring.

Thankyou