



NYC Taxi Trip Time Prediction

Presenter Name :

Vikas panchal

Naveen Kumar Batta



CONTENT

- Introduction
- Problem statement
- Data summary
- Exploratory Data Analysis (EDA)
- Feature Engineering & Selection
- Building and Evaluating Model
- Conclusion



Introduction

- In New York City, due to traffic jams, construction or road blockage etc. user will need to know how much time it will take to commute from one place to other.
- Increasing popularity of app-based taxi such as ola or uber and there competitive pricing levels made user decisive to choose based on trip pricing and duration.
- Taxi Drivers also have to choose best route having lesser trip time.
- So here we will be building a model which will be predicting the trip duration of taxies running in NewYork. This prediction will help customers to select the taxi based on trip duration and driver to select optimum route to their destination.

Problem Statement

We have the dataset Which is based on the 2016 NYC Yellow Cab trip record data made available in Big Query on the 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. Based on individual trip attributes, we should predict the duration of each trip in the test set.



DATA SUMMARY

- 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
- trip_duration - duration of the trip in seconds

BASIC EXPLORATION

- The dataset contains 1458644 rows and 11 columns.
- Two categorical features 'store_and_fwd_flag' and 'vendor_id'
- Outliers present in all numerical features
- Data formatting steps required for datetime features
- No null values present
- Passenger_count, Vendor_id and trip_duration are having integer value.
- pickup_datetime, dropoff_datetime is a datetime variable
- pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude are real numbers having float as data type *store_and_fwd_flag and Id belongs to a string data type.

IMPORT DATASETS & FIND MISSING VALUES

```
[3] #reading dataset
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/data/Regression nyc taxi trip time prediction/NYC Taxi Data.csv')
df.sample(5)
```

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration
1162860	id1242066	2	2016-05-06 06:07:39	2016-05-06 06:18:43	2	-73.978767	40.724079	-73.983482	40.757011	N	664
423458	id0199665	2	2016-02-22 10:54:56	2016-02-22 10:57:19	6	-73.980461	40.759888	-73.977028	40.763176	N	143
174734	id3853876	2	2016-06-02 22:28:14	2016-06-02 22:49:52	1	-73.983109	40.762989	-73.979347	40.736591	N	1298
113526	id2186855	1	2016-04-21 19:07:59	2016-04-21 19:17:09	1	-73.966972	40.756680	-73.953171	40.780029	N	550
325198	id1625327	1	2016-04-14 07:33:41	2016-04-14 07:42:43	1	-73.961609	40.774094	-73.970467	40.762161	N	542

```
[6] #checking missing values
```

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

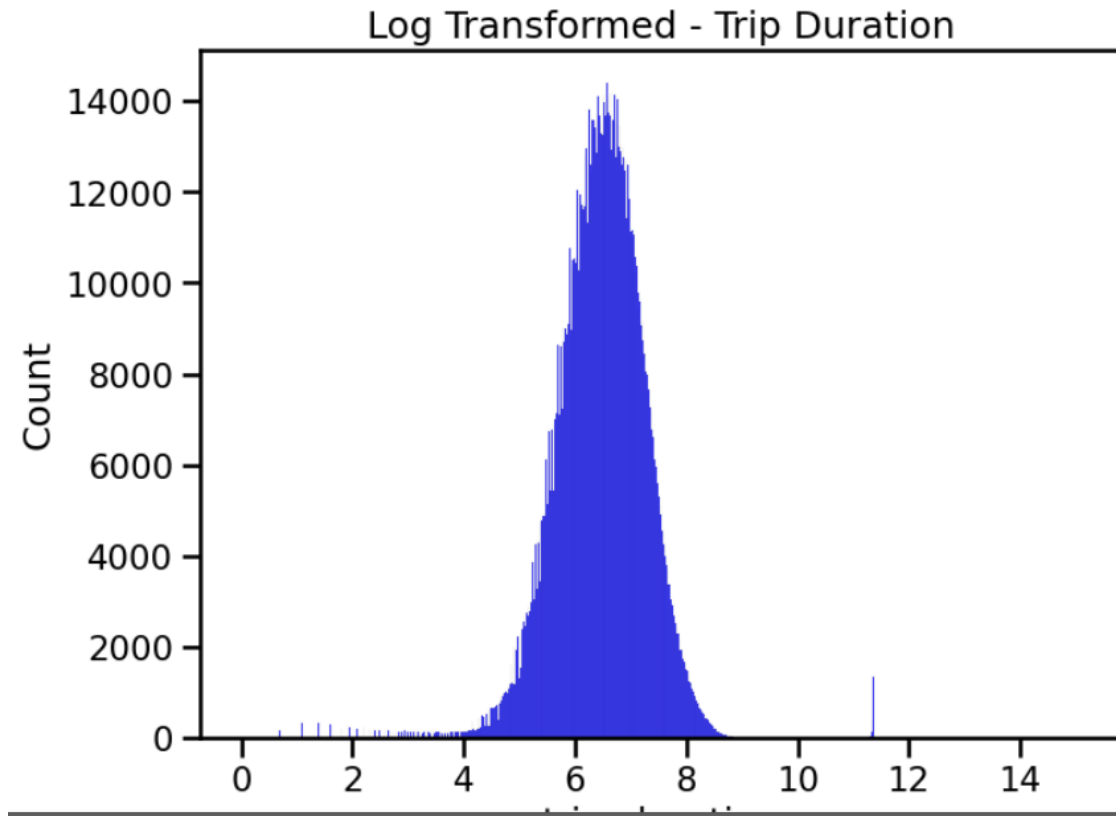
```
id                0
vendor_id         0
pickup_datetime   0
dropoff_datetime  0
passenger_count   0
pickup_longitude  0
pickup_latitude   0
dropoff_longitude  0
dropoff_latitude  0
store_and_fwd_flag 0
trip_duration     0
dtype: int64
```

- Quite good that our Dataset has no NULL values !!

```
df.info()
```

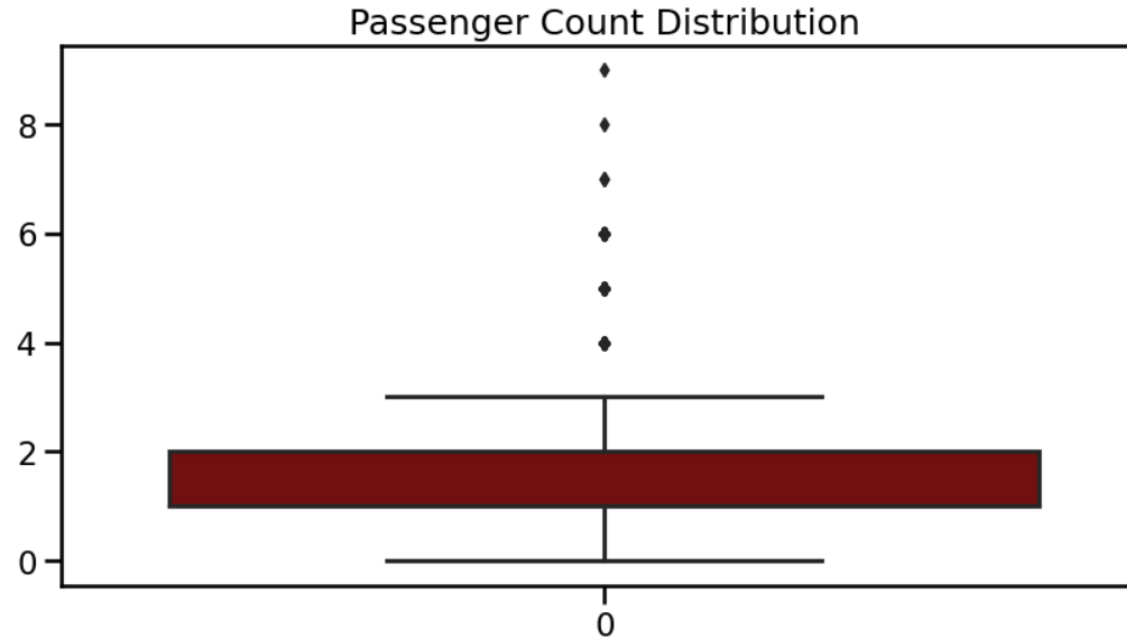
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1458644 entries, 0 to 1458643
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    1458644 non-null object
1   vendor_id            1458644 non-null int64
2   pickup_datetime      1458644 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
```

TRIP DURATION DATA ANALYSIS



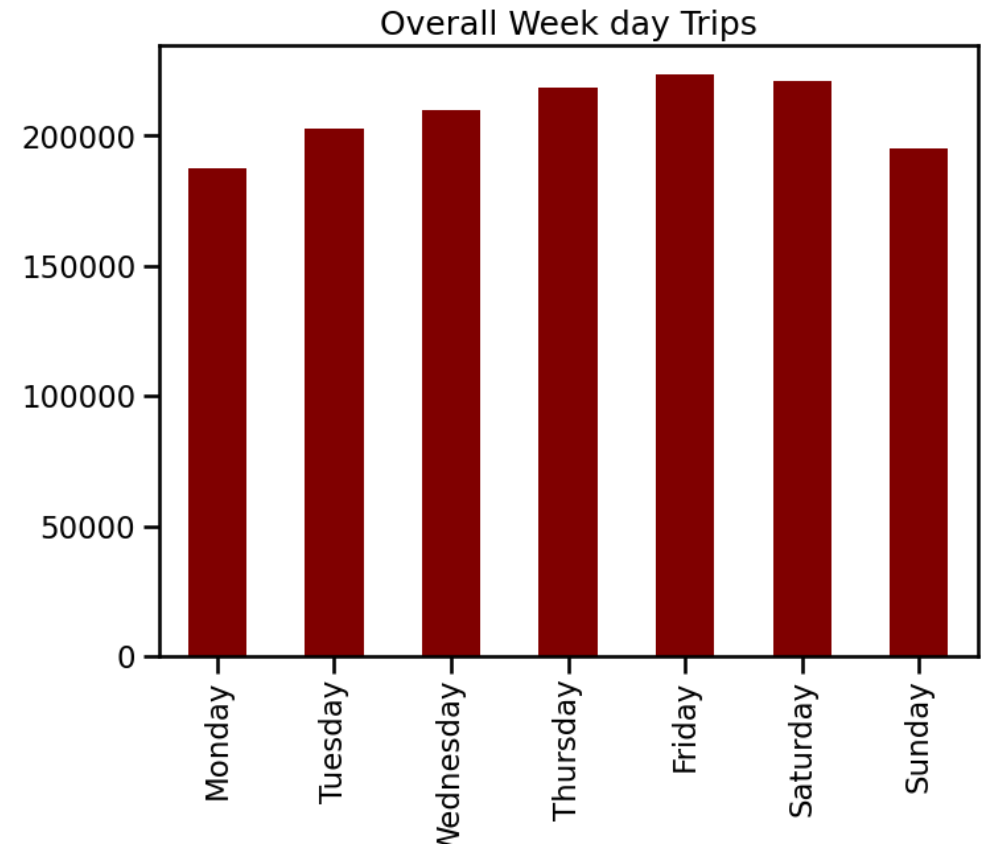
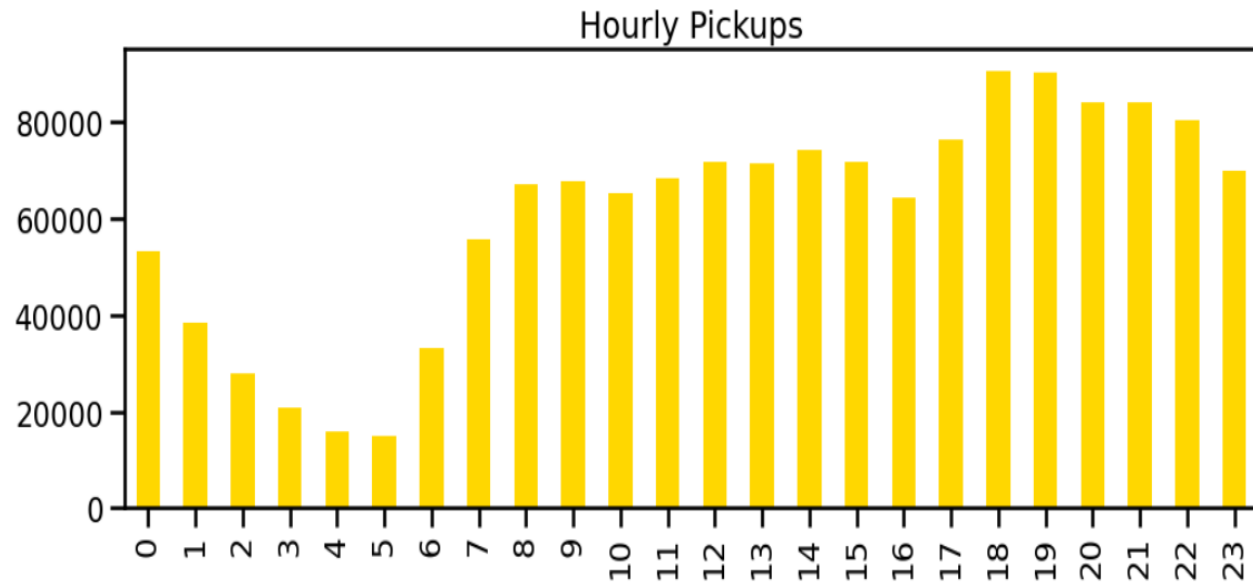
- Since our Evaluation Metric is RMSLE, we'll proceed further with Log Transformed "Trip duration".
- Log Transformation Smoothens outliers by proving them less weightage.

Passenger Count Distribution



- Most number of trips are done by 1-2 passenger(s).
- But one thing is Interesting to observe, there exist trip with Zero passengers, was that a free ride ? Or just a False data recorded ?
- Above 4 Passengers Indicate that the cab must be larger sized.

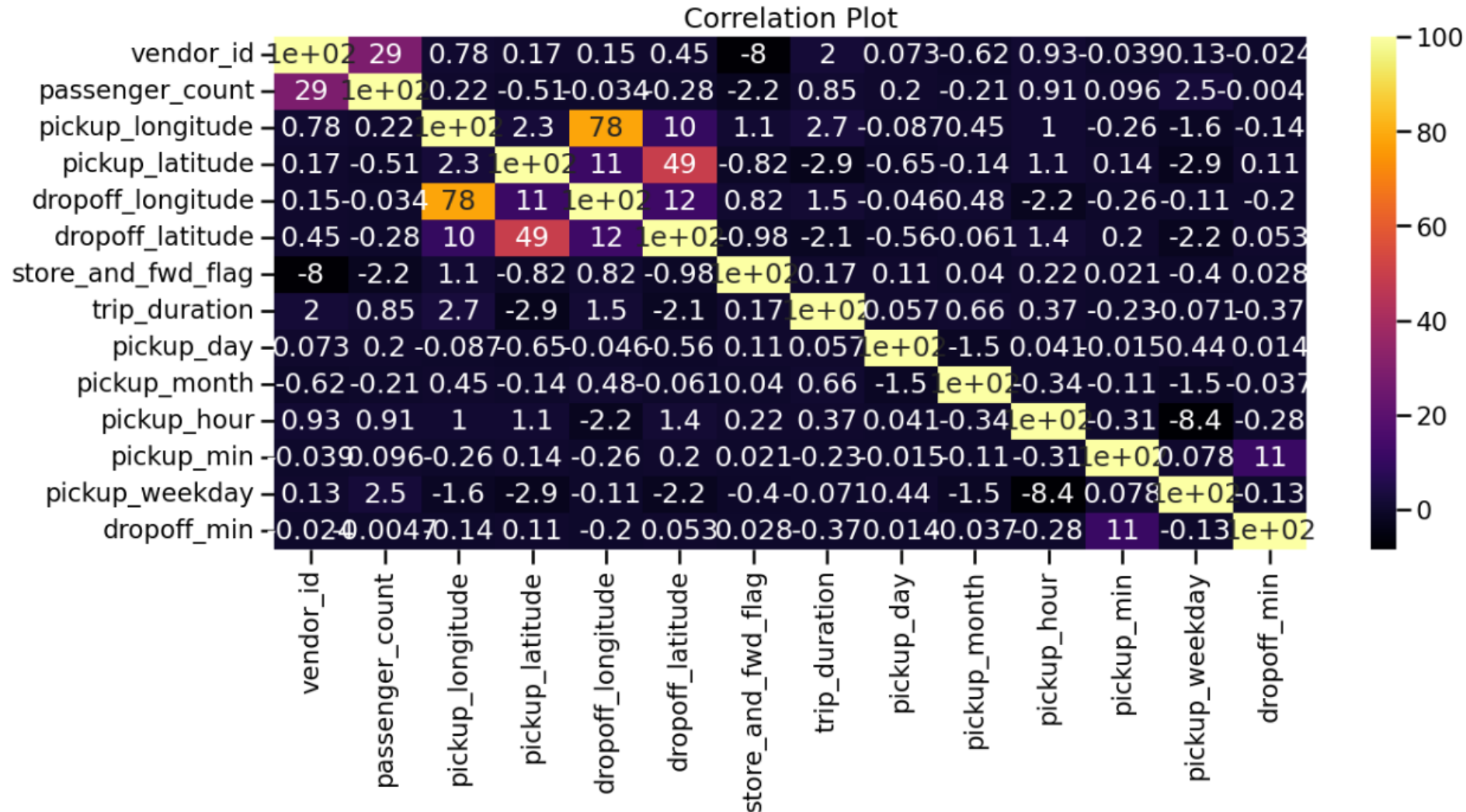
Pickups (Hourly & Weekdays)



- In which hour we get to see maximum pickups ? - Rush hours (5 pm to 10 pm), probably office leaving time.

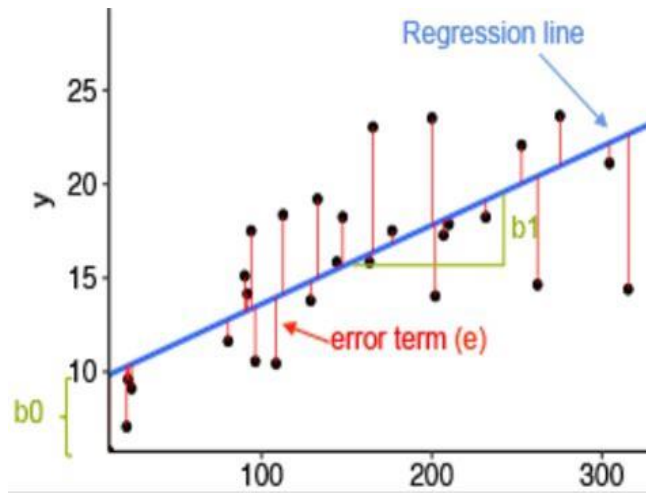
- Observations tells us that Fridays and Saturdays are those days in a week when New Yorkers prefer to come in the city. GREAT !!

Correlation Heatmap

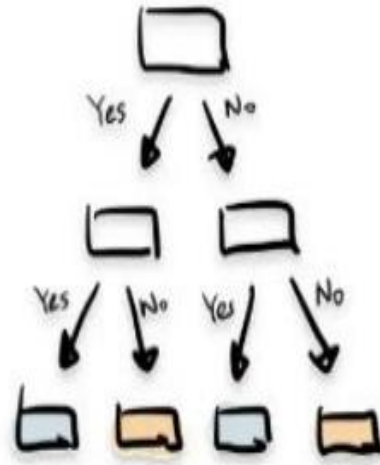


Machine Learning algorithms :

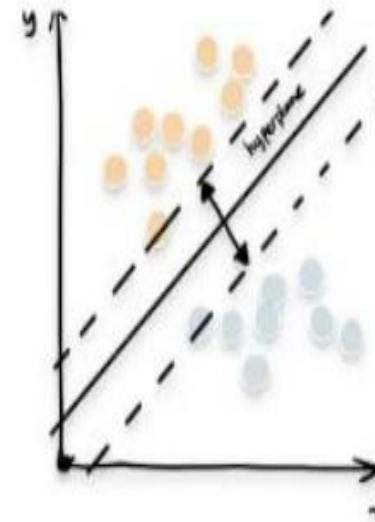
1. Linear Regression



2. Decision Tree



3. Support Vector Machine



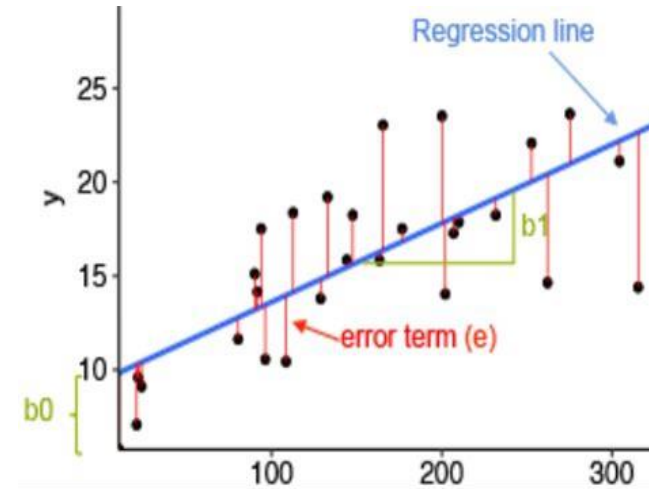
Linear Regression

Linear Regression is a regression of dependent variable on independent variable. It is a linear model that assumes a linear relationship between dependent (y) and independent variables (x). The dependent variable (y) is calculated by linear combination of independent variable (x).

$$y = b_0 + b_1 x_1 + b_2 x_2$$

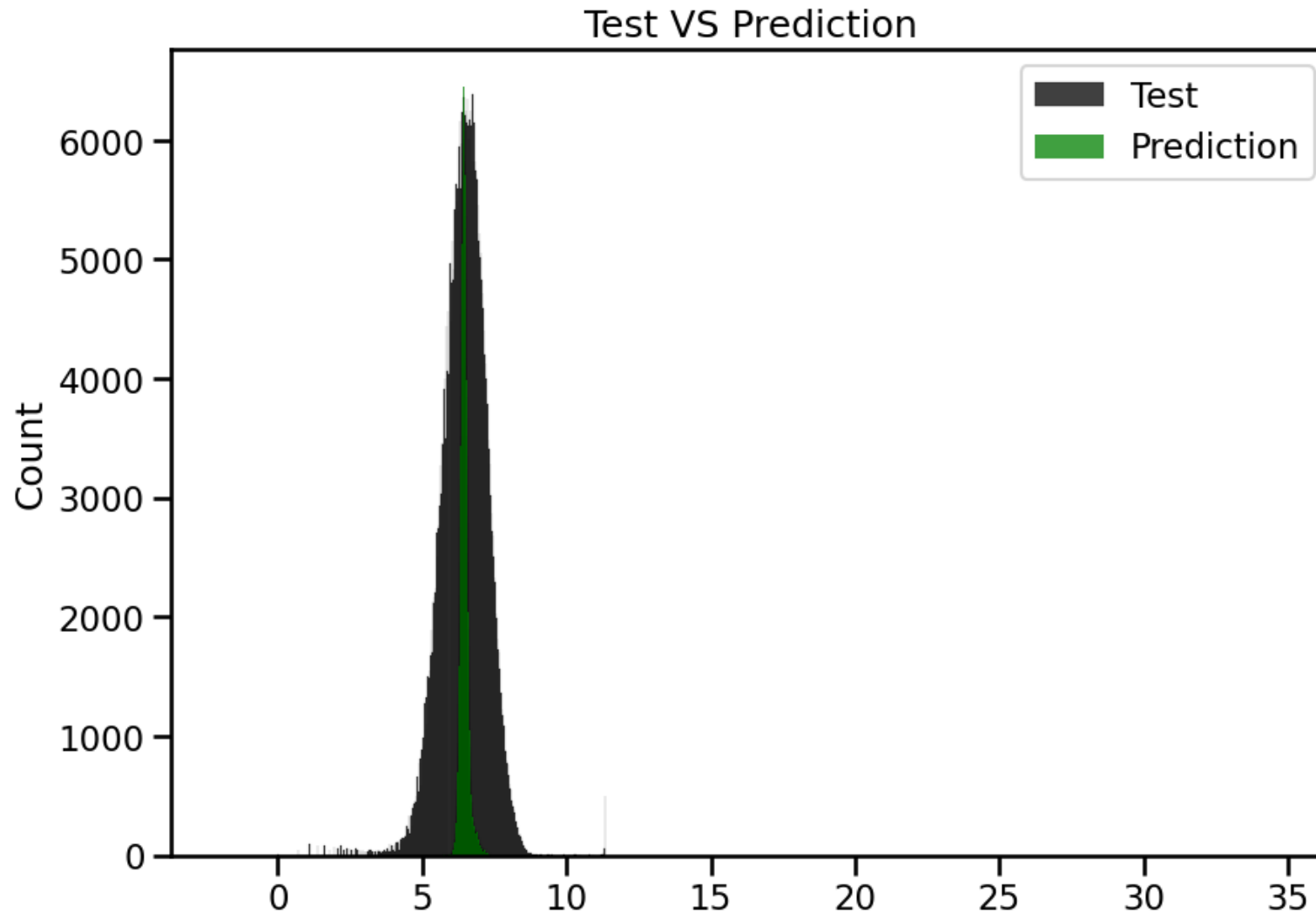
The cost function for linear regression is given by:
Mean of sum of square error

$$MSE = \frac{\sum (y_i - \hat{y}_i)^2}{n}$$



```
Training Score : 0.04244450511791209
Validation Score : 0.043892079955855645
Cross Validation Score : -0.048583442002601875
R2_Score : -23.10135437604674
```

prediction vs real data



Viz. we can clearly identify that the Linear Regression isn't performing good. The Actual Data (in Grey) and Predicted values (in Yellow) are so much differing. We can conclude that Linear Regression doesn't seem like a right choice for Trip duration prediction

Decision Tree Algorithm

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

```
#examining metrics
```

```
print ("Training Score : " , est_dt.score(X_train, y_train))
```

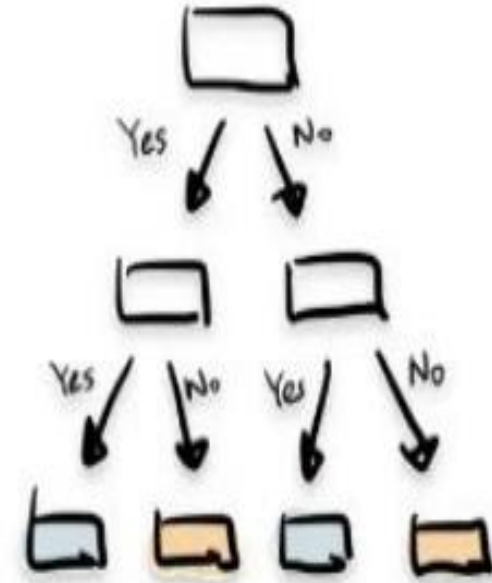
```
print ("Validation Score : " , est_dt.score(X_test, y_test))
```

```
print ("Cross Validation Score : " , cross_val_score(est_dt, X_train, y_train, cv=5).mean())
```

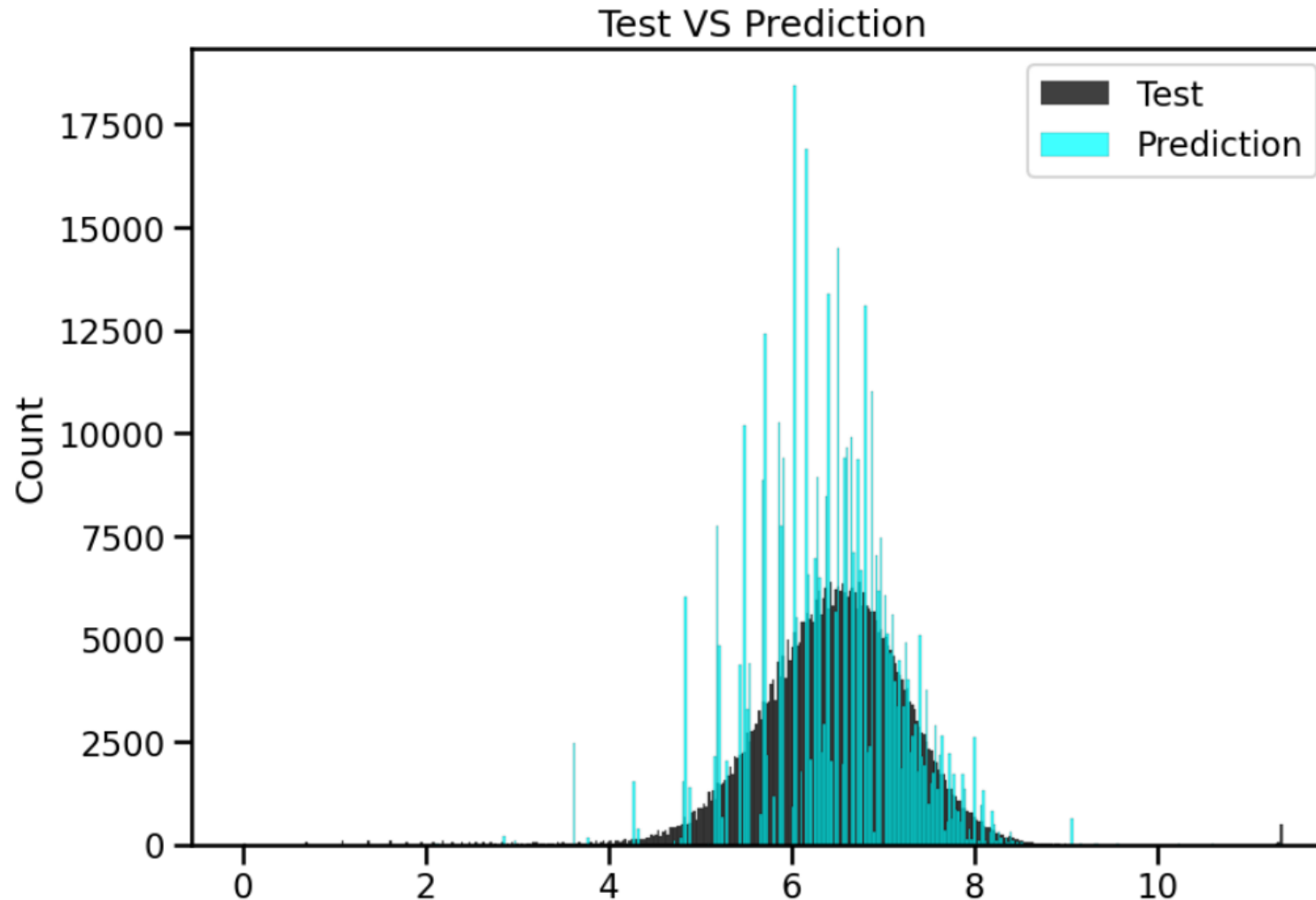
```
print ("R2_Score : " , r2_score(dt_pred, y_test))
```

```
print ("RMSLE : " , np.sqrt(mean_squared_log_error(dt_pred, y_test)))
```

```
Training Score : 0.9258236409034742  
Validation Score : 0.9167496153990037  
Cross Validation Score : 0.9136198482488055  
R2_Score : 0.9102582371818634  
RMSLE : 0.03756175968240503
```



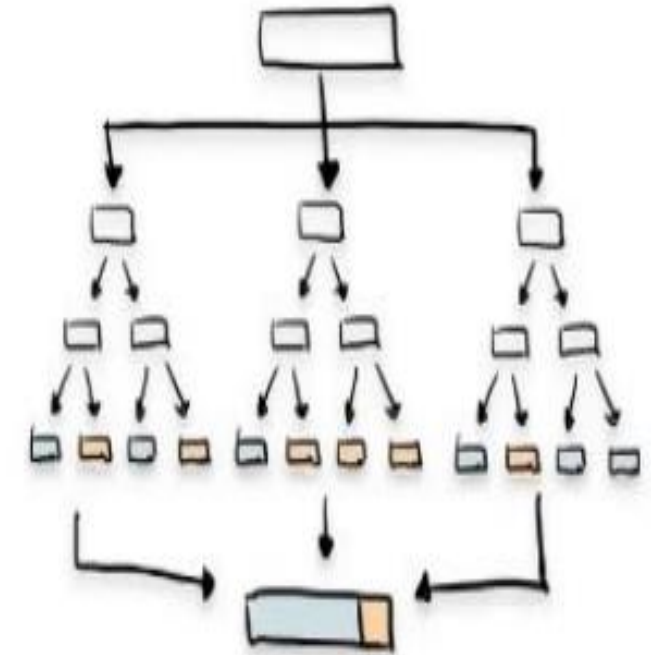
Decision Tree Algorithm :- prediction vs real data



From the above Viz. we can clearly identify that the Decision Tree Algorithm is performing good. The Actual Data (in Grey) and Predicted values (in Red) are as close as possible. We can conclude that Decision Tree could be a good choice for Trip duration prediction.

Random Forest Algorithm :

Random forest is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.



```
#examining metrics
```

```
print ("Training Score : " , est_rf.score(X_train, y_train))
```

```
print ("Validation Score : ", est_rf.score(X_test, y_test))
```

```
print ("Cross Validation Score : " , cross_val_score(est_rf, X_train, y_train, cv=5).mean())
```

```
print ("R2_Score : ", r2_score(rf_pred, y_test))
```

```
print ("RMSLE : ", np.sqrt(mean_squared_log_error(rf_pred, y_test)))
```

```
Training Score : 0.9305341702955872
```

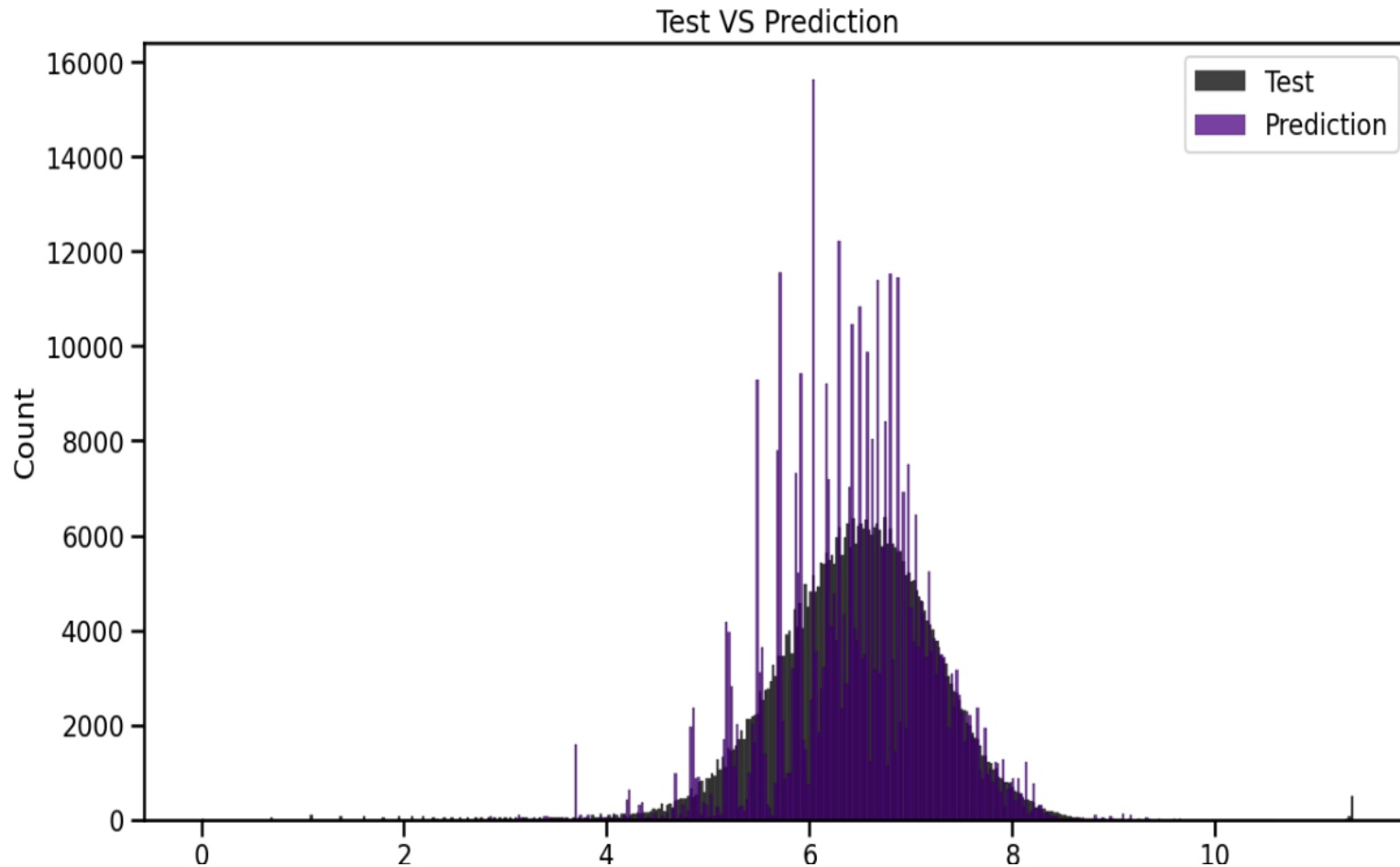
```
Validation Score : 0.9248838628922653
```

```
Cross Validation Score : 0.9242455425858781
```

```
R2_Score : 0.918176311630572
```

```
RMSLE : 0.035755791943398875
```

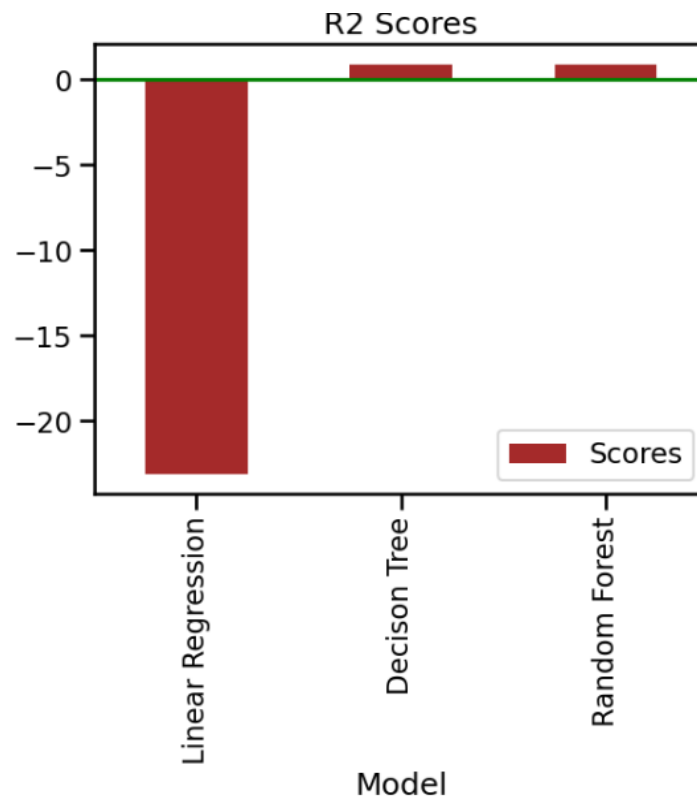
Random Forest Algorithm :- prediction vs real data



- From the above Viz. we can clearly identify that the Random Forest Algorithm is also performing good. The Actual Data (in Grey) and Predicted values (in Green) are as close as possible. We can conclude that Random Forest could be a good choice for Trip duration prediction.
- Similarly, we can Hyper tune Random Forest to get the most out of it.

R2 Scores Evaluation

- R2 Score or R-Squared is the proportion of the variance in the dependent variable that is predictable from the independent variable(s).
- Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y , disregarding the input features, would get a R^2 score of 0.0.



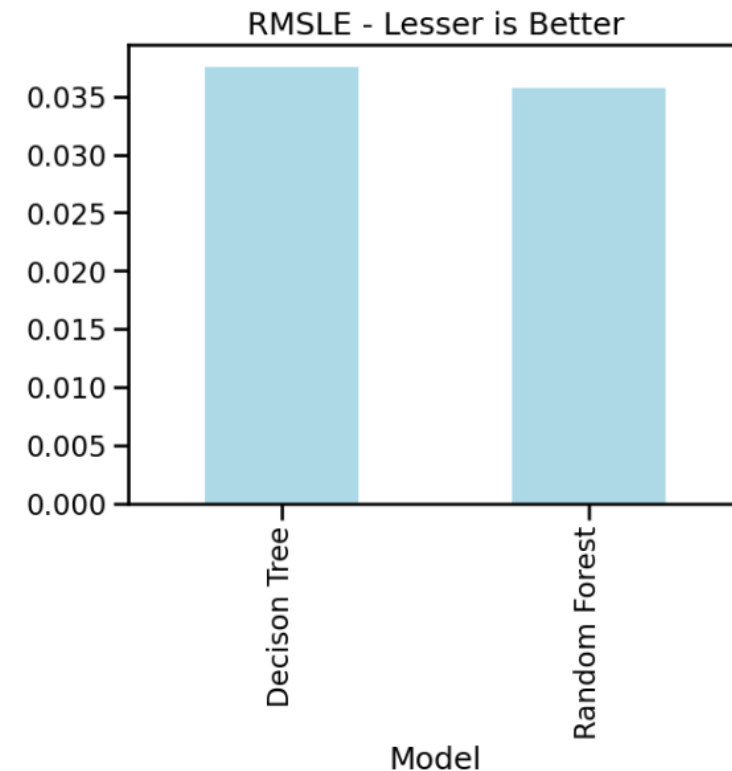
- Although , our Evaluation Metric isn't R2 Score but I'm just plotting them to check the Good Fit.
- We're getting good fit score for Decision Tree and Random Forest , i.e, close to 1.0

RMSLE Evaluation :

- RMSLE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.
- With RMSLE we explicitly know how much our predictions deviate.
- Lower values of RMSLE indicate better fit with lesser LOSS.

$$\epsilon = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(p_i + 1) - \log(a_i + 1))^2}$$

- ❖ Remember our NULL RMSLE : 0.1146 as a benchmark to beat.
- ❖ We can observe from above Viz. that our Decision Tree model and Random Forest model are good performers. As, Random Forest is providing us reduced RMSLE, we can say that it's a model to Opt for



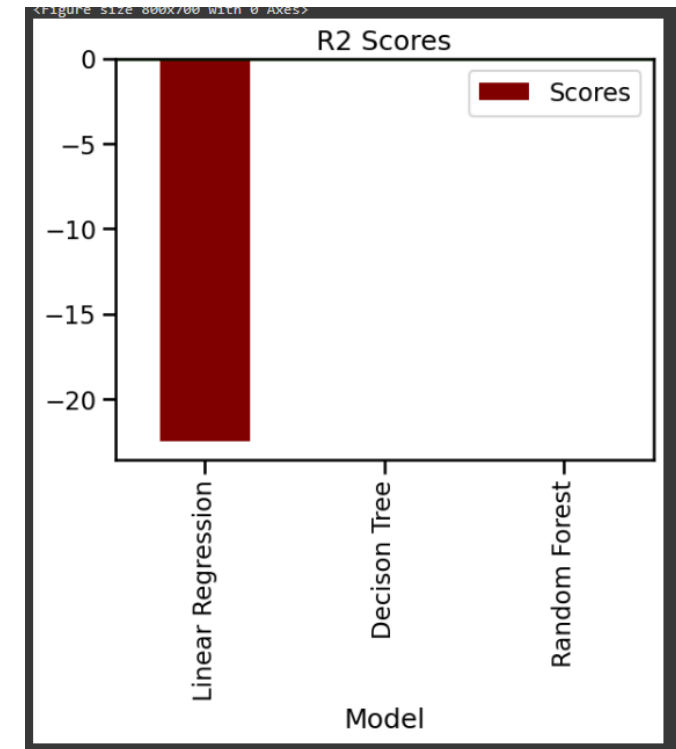
Second Approach - Without PCA (R2 Scores Evaluation) :

- Another approach we could go with is without PCA, just Standard Scaling Dataset and applying our Algorithms.
- The approach can give us better idea of what works better for us.
- This approach might take great amount of computational resources and time, it will be good if we can run this on Google's Collaboratory, that will eliminate huge computational stress on our system as the program will be running on Cloud

```
Training Score : 0.043473061426662074  
Validation Score : 0.045092484959132983  
Cross Validation Score : -0.048701119150320826  
R2_Score : -22.45617356992333  
RMSLE : 0.11228235950685247
```

```
Training Score : 0.4643053926882248  
Validation Score : 0.4579574624226187  
Cross Validation Score : 0.45745671106348684  
R2_Score : -0.16168520580612133  
RMSLE : 0.08783882677794577
```

```
Training Score : 0.4770512476779144  
Validation Score : 0.47142456987537174  
Cross Validation Score : 0.47093026897207546  
R2_Score : -0.17083406013290636  
RMSLE : 0.08692710581035555
```



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.
- With the help of Tableau, we're able to make good use of Geographical Data provided in the Dataset to figure prominent Locations of Taxi's pickup / dropoff points.
- In this project, we tried to predict the trip duration of a taxi in NYC.
- We are mostly concerned with the information of pick up latitude and longitude and drop off latitude and longitude, to get the distance of the trip.
- Hyperparameter tuning doesn't improve much accuracy.
- Also, found out some Trips of which pickup / dropoff point ended up somewhere in North Atlantic Sea.
- Passenger count Analysis showed us that there were few trips with Zero Passengers.
- Monthly trip analysis gives us a insight of Month – March and April marking the highest number of Trips while January marking lowest, possibly due to Snowfall




Challenges:

- Handling Large Dataset
- Feature Engineering
- Computation Time
- Optimising The Model



AImaBetter



 Vikas panchal
 +91 7089014472
 panchalvicky501@gmail.com

