NEW YORK CITY TAXI PRICE PREDICTION

Ankita Chaudhari

ankitachaudhari4699@gmail.com

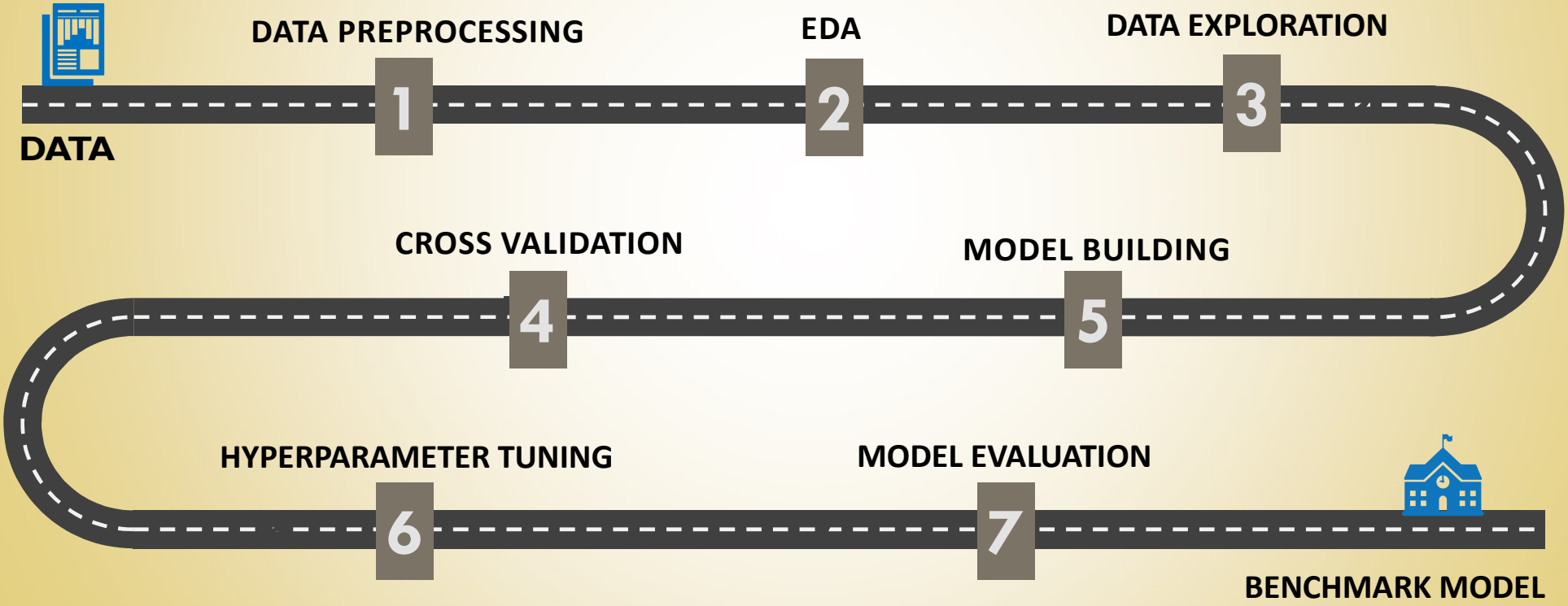
GIST OF DOMAIN

- **PROJECT BACKGROUND:** Taxis delivers service to lakhs of customers daily. Now it becomes really important to manage their data properly to come up with new business ideas to get best results. Eventually, it becomes really important to estimate the fare prices accurately.
- **OBJECTIVE:** In this project, we're looking to predict the fare for the New York city's taxi future transactional cases.
- **ABOUT DATASET:**The dataset is referred from Kaggle.

DATA DESCRIPTION

NUMERICAL	DESCRIPTION
key	Unique ID field
fare amount	Cost of each trip in USD
pickup_datetime	date and time when the meter was engaged
passenger count	The number of passengers in the vehicle
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

ROADMAP


DATA
DATA PREPROCESSING
1
EDA
2
DATA EXPLORATION
3
CROSS VALIDATION
4
MODEL BUILDING
5
HYPERPARAMETER TUNING
6
MODEL EVALUATION
7
BENCHMARK MODEL

DATA PREPROCESSING

- Removing non-critical columns
- Adding new features
- Calculating Distance in miles from the Longitude and Latitude
- Removing invalid records
- Duplicate values
- Missing data
- Outliers detection and treatment

EDA & DATA EXPLORATION INSIGHTS

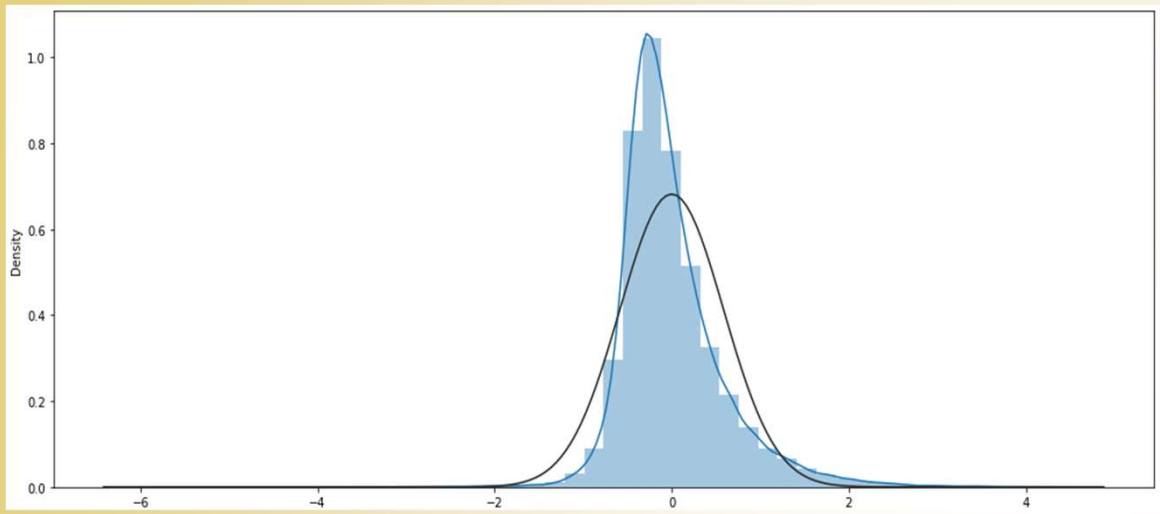
- We have 13 features in our dataset, In which 12 are Independent feature and 1 Dependent feature (i.e. fare amount).
- No negative fare amounts present.
- The average taxi fare amount is $8 Dollars.
- More number of booking is done for single passenger and maximum for 3 passenger.
- We cannot infer more details from latitude & longitude coordinates else then the maximum and minimum count.
- There are few data points with zero passenger count. In sometime we use to book taxi for goods transfer or it can also be cancelled taxi trips.

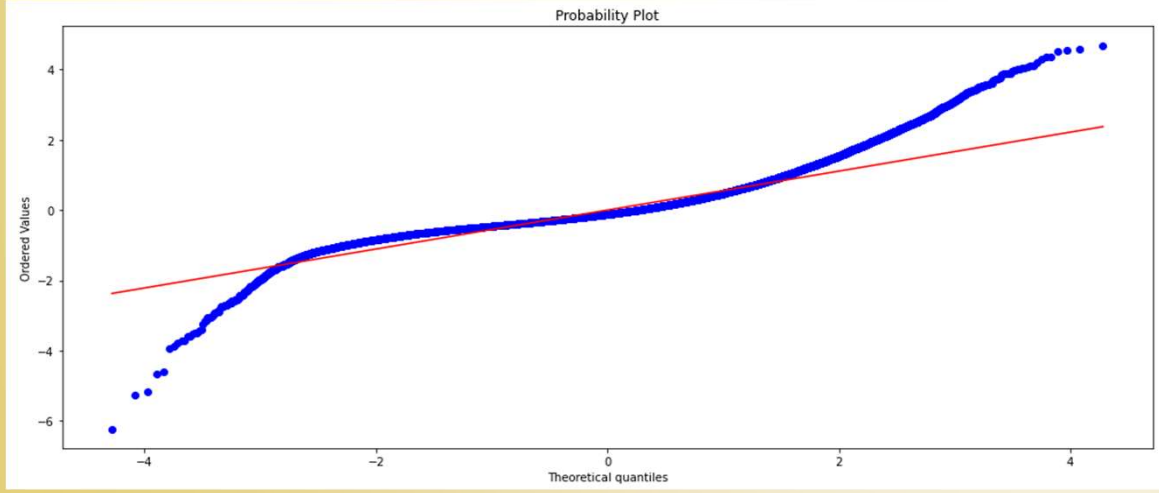
- We can observe the higher fare amount for passenger count of 3.
- Tuesday, Wednesday, Thursday, Friday has the maximum earning.
- Saturday, Sunday & Monday has minimum earning.
- The fare amount varies mostly between 5 to 10 USD.
- March, April, May & June are the highest grossing months. This can be due to tourism during the spring/summer season.
- 6 & 7 PM have the highest hour counts.
- Thursday is the busiest day of the week.

ASSUMPTIONS OF LINEAR REGRESSION

- Normality of Residuals
- There is no perfect multicollinearity.
- Test for Homoscedasticity
- No autocorrelation between the residuals.
- Linearity of Relationship

NORMALITY OF RESIDUALS

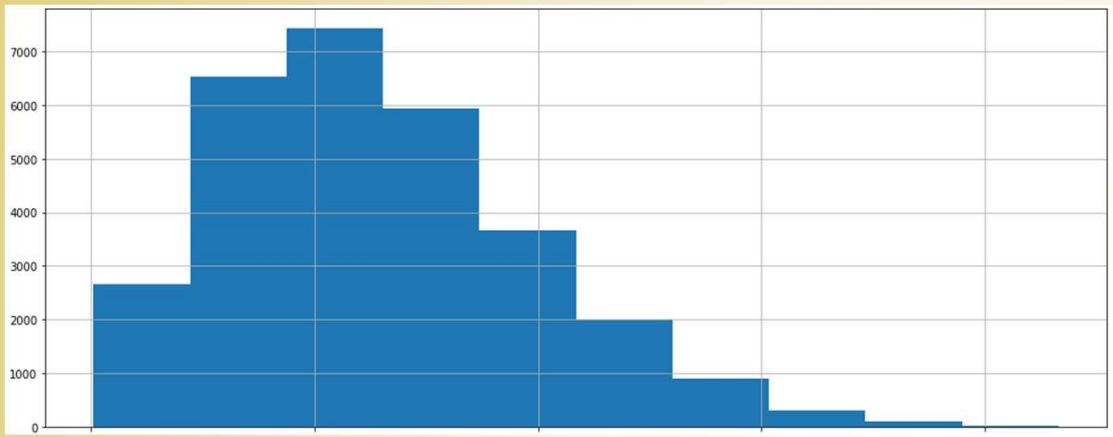


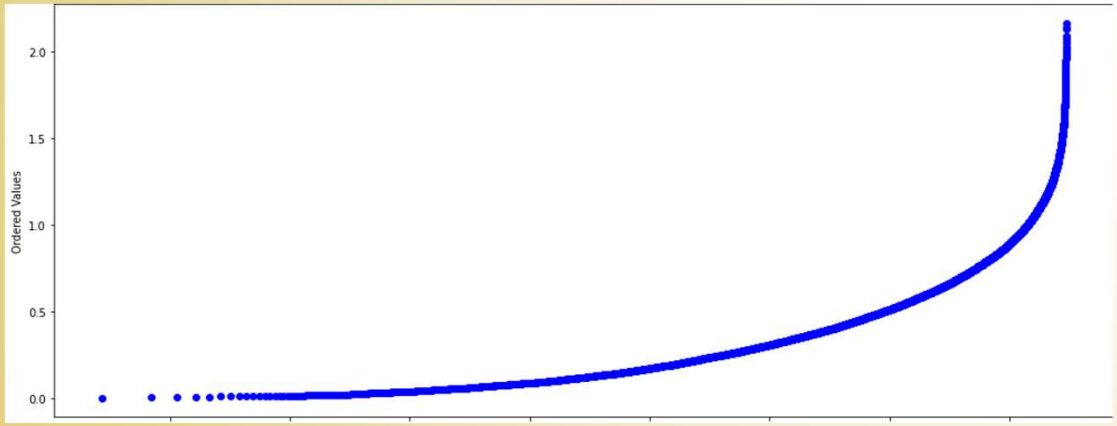



```
stat, p = jarque_bera(residuals)

# to print the numeric outputs of the Jarque-Bera test upto 3 decimal places
# %.3f: returns the a floating point with 3 decimal digit accuracy
# the '%' holds the place where the number is to be printed
print('Statistics=%.3f, p-value=%.3f' % (stat, p))

# display the conclusion
# set the level of significance to 0.05
alpha = 0.05

# if the p-value is greater than alpha print we accept alpha
# if the p-value is less than alpha print we reject alpha
if p > alpha:
    print('The data is normally distributed (fail to reject H0)')
else:
    print('The data is not normally distributed (reject H0)')
```

```
Statistics=107191.440, p-value=0.000
The data is not normally distributed (reject H0)
```

TRANSFORMATION USING SQRT METHOD

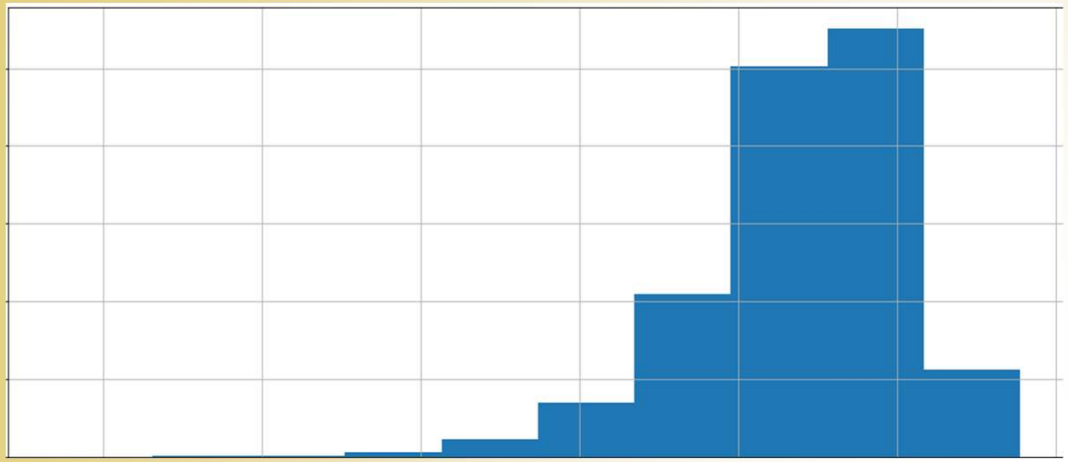


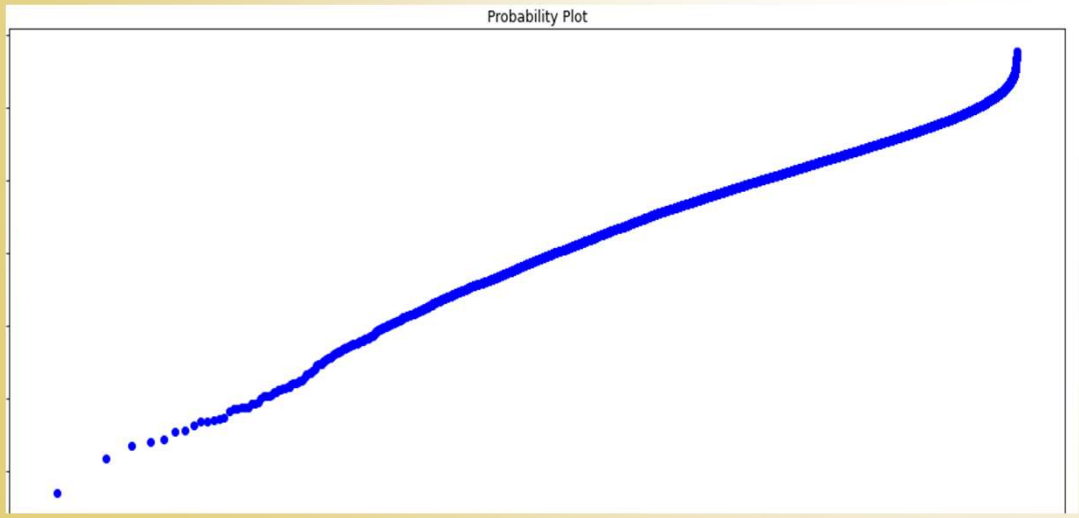



```
stat, p = jarque_bera(ab)

# to print the numeric outputs of the Jarque-Bera test upto 3 decimal places
# %.3f: returns the a floating point with 3 decimal digit accuracy
# the '%' holds the place where the number is to be printed
print('Statistics=%.3f, p-value=%.3f' % (stat, p))

# display the conclusion
# set the level of significance to 0.05
alpha = 0.05

# if the p-value is greater than alpha print we accept alpha
# if the p-value is less than alpha print we reject alpha
if p > alpha:
    print('The data is normally distributed (fail to reject H0)')
else:
    print('The data is not normally distributed (reject H0)')
```

```
Statistics=nan, p-value=nan
The data is not normally distributed (reject H0)
```

TRANSFORMATION USING LOG METHOD






```
stat, p = jarque_bera(cd)

# to print the numeric outputs of the Jarque-Bera test upto 3 decimal places
# %.3f: returns the a floating point with 3 decimal digit accuracy
# the '%' holds the place where the number is to be printed
print('Statistics=%.3f, p-value=%.3f' % (stat, p))

# display the conclusion
# set the level of significance to 0.05
alpha = 0.05

# if the p-value is greater than alpha print we accept alpha
# if the p-value is less than alpha print we reject alpha
if p > alpha:
    print('The data is normally distributed (fail to reject H0)')
else:
    print('The data is not normally distributed (reject H0)')
```

```
Statistics=nan, p-value=nan
The data is not normally distributed (reject H0)
```

THERE IS NO PERFECT MULTICOLLINEARITY

- We will use Variation Inflation factor (VIF) to check multicollinearity. The VIF of a Linear Regression is defined as VIF=1/1-R2.
- With VIF > 5 there is an indication that multicollinearity may present.
- With VIF > 10 there is certainly multicollinearity present between the independent and the target variables.
- As per our analysis, we observe that none of the features show strong multicollinearity.
- Therefore, we can conclude that there is No or (little) Multicollinearity in our dataset.

```
vf = [vif(Xc_scaled.astype(int).values, i) for i in range(Xc_scaled.astype(int).shape[1
pd.DataFrame(vf, index=Xc_scaled.astype(int).columns, columns=['vif'])
```

	vif
const	1.236570
pickup_longitude	1.419473
pickup_latitude	1.493493
dropoff_longitude	1.322377
dropoff_latitude	1.397138
passenger_count	1.005308
day	1.000507
dayofweek	1.012567
hour	1.007816
minute	1.000124
month	1.000566
distance_miles	1.012187

TEST FOR HOMOSCEDASTICITY

- Checking heteroscedasticity : Using Goldfeld Quandt we test for heteroscedasticity.
- H0 : Error terms are homoscedastic
- Ha : Error terms are not heteroscedastic.

```
name = ['F statistic', 'p-value']
test = sms.het_goldfeldquandt(y=residuals, x=Xc_scaled.astype(int))
lzip(name, test)
```

```
[('F statistic', 0.9883683581249076), ('p-value', 0.870594011099635)]
```

- Since p value is more than 0.05 in Goldfeld Quandt Test, we fail to reject the null hypothesis.
- We can conclude that the error terms are homoscedastic.

NO AUTOCORRELATION BETWEEN THE RESIDUALS.

- The Durbin Watson Test is a measure of autocorrelation in residuals from the regression analysis.
- It's value ranges from 0-4. If the value of Durbin- Watson is Between 0-2, it's known as Positive Autocorrelation.
- If the value ranges from 2-4, it is known as Negative autocorrelation.
- If the value is exactly 2, it means No Autocorrelation.
- For a good linear model, it should have low or no autocorrelation.

Omnibus:	17569.989	Durbin-Watson:	1.976
Prob(Omnibus):	0.000	Jarque-Bera (JB):	89934.337
Skew:	1.048	Prob(JB):	0.00
Kurtosis:	7.956	Cond. No.	2.54

- Durbin-Watson Statistic=1.976
- Since this value is very close to 2. Hence we can conclude that there is almost no autocorrelation

LINEARITY OF RELATIONSHIP

- **Rainbow Test**
- H0: fit of model using full sample = fit of model using a central subset (linear relationship)
- H1: fit of model using full sample is worse compared to fit of model using a central subset

```
name = ['F statistic', 'p-value']
test_1=sm.stats.diagnostic.linear_rainbow(model)
lzip(name, test_1)
```

```
[('F statistic', 0.9709890534186474), ('p-value', 0.9977555189763275)]
```

- As we can see the p-value is more than the 0.05 so we fail to reject the null hypothesis.
- We can conclude the linear relationship.

ML MODELS
MODEL EVALUATION

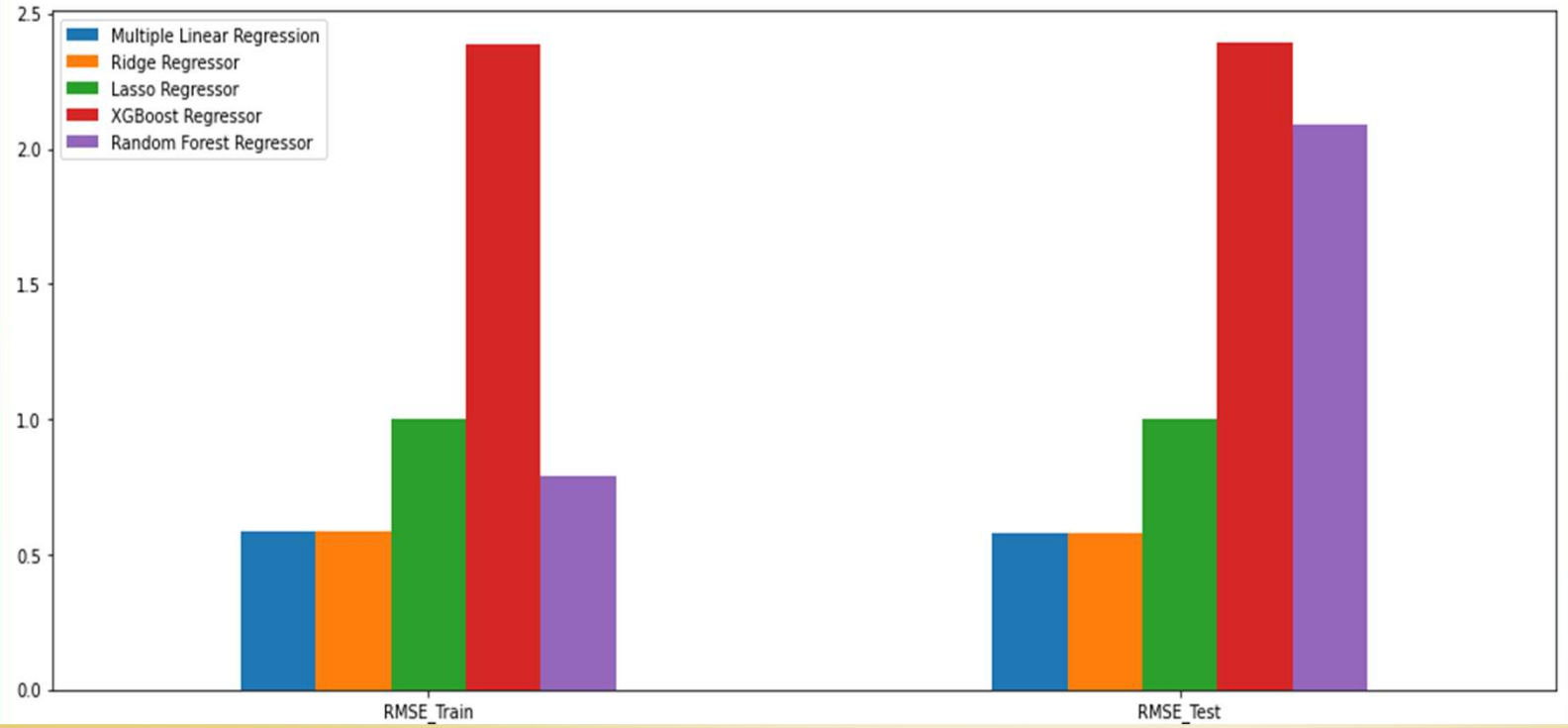
- We have built Machine Learning models like,

i. Linear Regression
ii. Ridge Regression
iii. Lasso Regression
iv. XGBoost Regressor
v. Random Forest Regressor

COMPARING THE EVALUATION OF THE MODEL WITH VISUALIZATION (BEFORE HYPER PARAMETER TUNING)

	Multiple Linear Regression	Ridge Regressor	Lasso Regressor	XGBoost Regressor	Random Forest Regressor
RMSE_Train	0.587	0.587	0.999	2.383	0.789
RMSE_Test	0.582	0.582	1.003	2.389	2.089

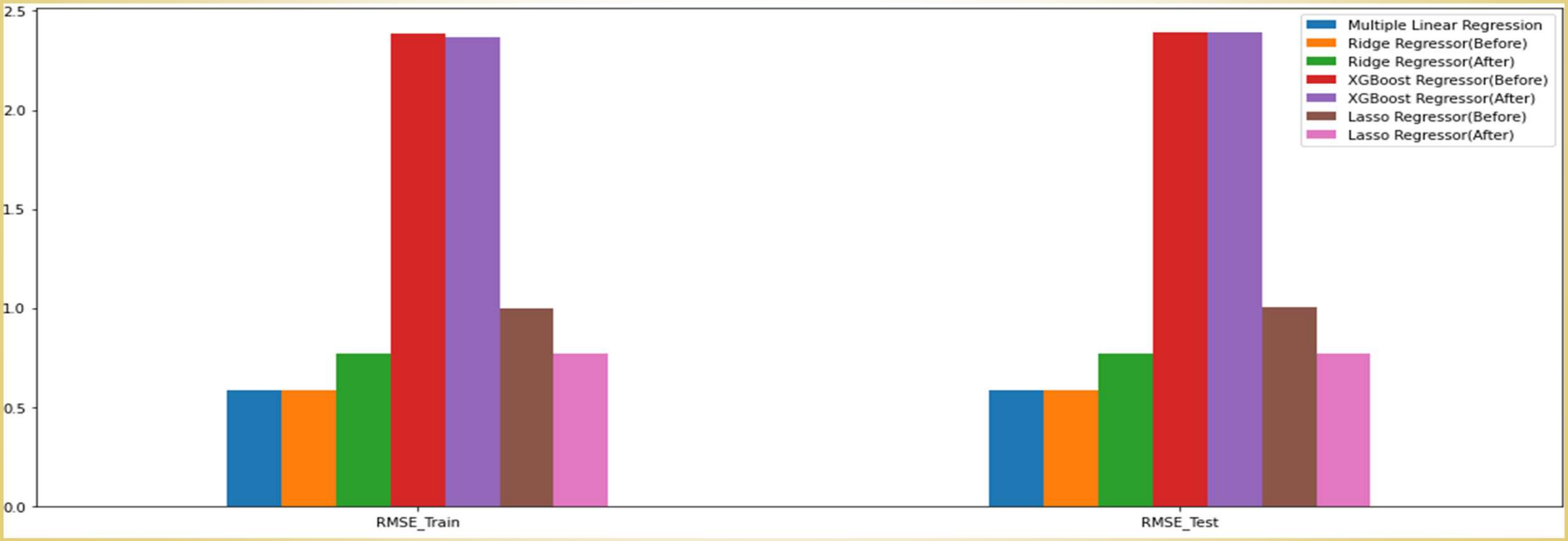
```
ax = EM.plot.bar(rot=0)
```



- According to the bar plot, we can conclude that:
- Random Forest regressor is not a good choice as it is clearly overfitting.
- XGBoost Regressor shows very high RMSE scores which we don't want but we can try using hyper parameter tuning and check for the reduced scores.
- Lasso Regressor is showing slightly higher RMSE score which is also not desirable but we can still try using hyper parameter tuning on it.
- Multiple Linear Regression and Ridge Regressor show stable scores and much less RMSE scores than the other Regressors. But we can still try using hyper parameter tuning on Ridge Regressor for the optimal results.

COMPARING THE EVALUATION OF THE MODEL WITH VISUALIZATION (POST HYPER PARAMETER TUNING)

	Multiple Linear Regression	Ridge Regressor(Before)	Ridge Regressor(After)	XGBoost Regressor(Before)	XGBoost Regressor(After)	Lasso Regressor(Before)	Lasso Regressor(After)
RMSE_Train	0.587	0.587	0.771	2.383	2.367	0.999	0.771
RMSE_Test	0.582	0.582	0.771	2.389	2.394	1.003	0.771



CONCLUSIONS

- Visualizing the distribution of data & their relationships, helped us to get some insights on the feature-set.
- Tested the Assumptions of the Linear Regression and have observed that the features show moderate linear relationship w.r.t the target variables. It was also observed that the residuals were not normally distributed and the transformations did not have any significant effect.
- **Model Conclusion**
- Lower values RMSE scores are expected.
- XGBoost Regressor & Lasso Regressor (Before & After the hyper parameter tuning) cannot be considered due to higher RMSE scores.
- Ridge Regressor (After hyper parameter tuning) gives slightly higher RMSE scores. So we can think of Ridge Regressor (Before hyper parameter tuning)
- Therefore, Multiple Linear Regressor & Ridge Regressor with the default parameter settings can be considered as a better fit model because of lesser RMSE scores as compared to other models.

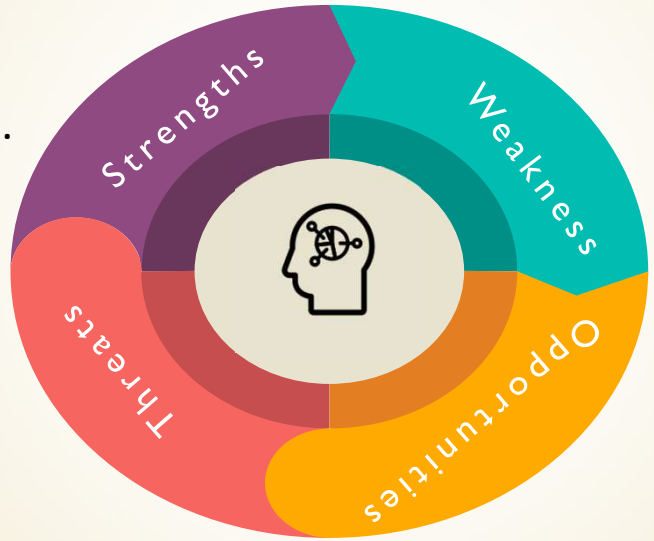
SWOT ANALYSIS

STRENGTHS

- Can understand the demographics of the customers.
- No missing values.
- No duplicate values

THREATS

- The fare amounts predicted can fluctuate due to economical issues like rise in fuel prices, etc.



WEAKNESS

- Having outliers in the data set.
- Invalid record entries.

OPPORTUNITIES

- Use the data to create more new features.
- Study the hidden patterns in demographics and make relevant recommendation to the business

THANK YOU