FLIP ROBO

HOUSING: PRICE PREDICTION

Submitted by:

Deepam Purkayastha

ACKNOWLEDGMENT

All the required information & the dataset were scrapped from olx & cardekho websites. Also, I have used a few below external resources that helped me to complete the project.

External Resources:

1) Google
2) https://scikit-learn.org
3) kaggle & github

INTRODUCTION

- ## Business Problem Framing

  With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model.

- ## Conceptual Background of the Domain Problem

  The client works with small traders, who sell used cars. Hence, by using the websites such as olx, & cardekho, we need to scrap the data & using this data, we need to build the machine learning model, which would help the client to understand the used car market & accordingly they would be able to sell the used car in the market.

- ## Review of Literature

  Based on the sample data we have scrapped from the websites like olx & cardekho where we have understood the real time data & the valuation of used car in the market. The data set explains it is a regression problem as we need to build a model using Machine Learning in order to

predict the price of used car. Also, we have other independent features that would help to decide which all variables are important to predict the price of the variable and how do these variables describe the price of the used car.

- Motivation for the Problem Undertaken

Based on the problem statement & the real time data scrapped from the olx & cardekho websites, I have understood how each independent features helped me to understand the data as each features provides a different kind of information. It is so interesting to work with different types of real time data in a single data set and perform root cause analysis to predict the price of the used car. Based on the analysis of the brand, model of the car, Km driven, & transmission, etc. I would be able to model the price of used car as this model will then be used by the client to understand how exactly the prices vary with the variables. They can accordingly work on it & make some strategies to sell the used car and get some high returns. Furthermore, the model will be a good way for the client to understand the pricing dynamics of a used car.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

The statistical analysis by using data. Describe().

```
pd.DataFrame(X_scaled).describe()
```

	0	1	2	3	4	5	6	7	8	9	...	34	35	
count	938.000000	938.000000	938.000000	938.000000	938.000000	938.000000	938.000000	938.000000	938.000000	938.000000	...	938.000000	938.000000	9
mean	0.515404	0.847548	0.521435	0.326837	0.002132	0.331557	0.658849	0.005330	0.127932	0.007463	...	0.019190	0.452026	
std	0.146453	0.359650	0.351444	0.089332	0.046151	0.471024	0.474349	0.072854	0.334192	0.086110	...	0.137265	0.497959	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	
25%	0.388312	1.000000	0.150442	0.269864	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	
50%	0.519094	1.000000	0.513274	0.331187	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	
75%	0.617152	1.000000	0.849558	0.387988	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	...	0.000000	1.000000	
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000	

Here, I got to know statistics of the data set where it explains the mean, standard deviation, minimum & maximum value, and how the % of the data is distributed. Then by the help of .info(), I get to know the data type & if there are any missing values. By using correlation function &

heat map, I have understood if there is any multi-collinearity issue or which feature is negative/positive correlated with the target variable. I have used few visualization techniques to understand more about the data and which helped me to decide the importance of independent features. Also, checked the outliers/skewness of integer data type by using z-score &. skew() method. I have used minmaxscaler method to scale the data before building the model.

- ## Data Sources and their formats

  Data set has been scrapped from olx & cardekho websites having format of CSV (Comma Separated Values). The dimension of data is 1017 rows & 10 columns.

- ## Data Pre-processing Done

1) Checking for null values: Null values found in a 'Variant' column, & removed the column as it is having noisy data & it will cost more when it would require to encoded.
2) Dropped a few unwanted columns as it is having no relation with the target variable.
3) Some of the columns are having '- '. Hence removed it.
4) We have details of brand & model together. Hence I have split the details of model & brand respectively.
5) For Km driven column, I have removed Kms & make it as Integer data type. Also, for price column removed lakh & *, which makes it integer column.
6) For No of owner's column, I have combined & make as first, second, third & fourth owner. Also, for price column, we have changed the format as integer data type by multiplying with 100000.
7) For Year column, took the difference from current year (2021). Also, for fuel column, I have combined the same category together as it gives the same meaning.
8) Checked for the correlation to visualize the feature importance & accordingly dropped a few features if required, which are highly correlated to each other.
9) Performed encoding for the categorical features.

10) Checked for the outliers, however not applied z-score as minimum data loss was 23.56%, which is huge to remove.
11) Checked for the skewness & transformed few continuous features by using power transformation method.
12) Scaling has been done using minimax scaler method.
13) Checked for the final dimension of dataset: 938 rows & 45 columns.
14) Created train test split: We have split the train & test data in 0.2 test size with finding the best random state, which is 46.

- ## Data Inputs- Logic- Output Relationships

I have all the data format as Integer & float in the dataset. To visualize the inputs-output relationship, I have used strip plot, boxplot, scatterplot & boxen plot, which helped me to understand that each categorical features has some relation with the price. Hence kept all the feature for model building. Most of the numerical columns are not having a linear relationship, only few of them have as most of the data is scattered. So, have used subplot & found most of the numerical columns are skewed & having outliers.

- ## Hardware and Software Requirements and Tools Used

Hardware: Laptop (OS: Windows, RAM: 8GB)

Software: Anaconda jupyter notebook

Libraries:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
import statsmodels.api as sm
import warnings
warnings.filterwarnings('ignore')
```

1) I have used numpy pandas to load the dataset, perform the data cleaning part & perform EDA.

2) To visualize the dataset & to check correlation/multi-collinearity, I have used seaborn & matplotlib.
3) For Scaling the data set, I have used StandardScaler/MinmaxScaler.
4) For outliers removal, I have used scipy.stats to import z-score.
5) For encoding the categorical feature, I have used Label encoding method.
6) Sklearn has been used for pre- processing the data and for model building.
7) Joblib has used to save the model.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  I have used both statistical and analytical approaches to solve the problem which mainly includes the pre-processing of the data and EDA to check the correlation of independent & dependent features. Also, before building the model, I make sure that the data is cleaned & scaled.

- Testing of Identified Approaches (Algorithms)

1) Linear Regression
2) DecisionTreeRegressor
3) RandomForestRegressor
4) Support Vector Regressor
5) AdaBoostRegressor
6) GradientBoostingRegressor
7) BaggingRegressor
8) Ridge & Lasso.

- Run and Evaluate selected models

1) Linear Regression

```
LR=LinearRegression()
LR.fit(x_train,y_train)
pred_test=LR.predict(x_test)

print(r2_score(y_test,pred_test))
```

```
0.6389322405073128
```

2) DecisionTreeRegressor

```
from sklearn.tree import DecisionTreeRegressor

DTR=DecisionTreeRegressor()
DTR.fit(x_train,y_train)
pred_test=DTR.predict(x_test)

print(r2_score(y_test,pred_test))
```

```
0.51133261395706
```

3) RandomForestRegressor

```
## RandomForestRegressor
RFR1= RandomForestRegressor()
RFR1.fit(x_train, y_train)
y_pred = RFR1.predict(x_test)
RFR1.score(x_train, y_train)
```

```
0.8926541837125526
```

4) Support Vector Regressor

```
from sklearn.svm import SVR
SV= SVR()
SV.fit(x_train,y_train)
pred_test=SV.predict(x_test)

print(r2_score(y_test,pred_test))
```

```
-0.024482065734022562
```

5) AdaBoostRegressor

```
from sklearn.ensemble import AdaBoostRegressor
ADR= AdaBoostRegressor()
ADR.fit(x_train,y_train)
pred_test=ADR.predict(x_test)

print(r2_score(y_test,pred_test))
```

```
0.4314500487901629
```

6) GradientBoostingRegressor

```
from sklearn.ensemble import GradientBoostingRegressor
GBR= GradientBoostingRegressor()
GBR.fit(x_train,y_train)
pred_test=GBR.predict(x_test)

print(r2_score(y_test,pred_test))
```

```
0.7600034277602261
```

7) Bagging Regressor

```
from sklearn.ensemble import BaggingRegressor
BR= BaggingRegressor()
BR.fit(x_train,y_train)
pred_test=BR.predict(x_test)

print(r2_score(y_test,pred_test))
```

```
0.6659319139423701
```

8) Lasso & Ridge

```
ls=Lasso(alpha=0.001)
ls.fit(x_train,y_train)
ls.score(x_train,y_train)
predls=ls.predict(x_test)
print(r2_score(y_test,predls))
print('mean_squared_error:',mean_squared_error(y_test,predls))
print('mean_absolute_error:',mean_absolute_error(y_test,predls))
print('root_mean_squared_error',np.sqrt(mean_squared_error(y_test,predls)))
```

```
0.6389366442366785
mean_squared_error: 31412882441.402943
mean_absolute_error: 116270.95481869382
root_mean_squared_error 177236.7976504962
```

```
rd=Ridge(alpha=0.001)
rd.fit(x_train,y_train)
rd.score(x_train,y_train)
pedi=rd.predict(x_test)
print(r2_score(y_test,pedi))
print('mean_squared_error:',mean_squared_error(y_test,pedi))
print('mean_absolute_error:',mean_absolute_error(y_test,pedi))
print('root_mean_squared_error',np.sqrt(mean_squared_error(y_test,pedi)))
```

```
0.6391372331703962
mean_squared_error: 31395430998.343594
mean_absolute_error: 116248.76985232758
root_mean_squared_error 177187.55881365822
```

Cross Validation

```
scr = cross_val_score(LR, X_scaled, y, cv=5, scoring= 'r2')
print("Cross validation score for LinearRegression model:" , scr.mean())
```

```
Cross validation score for LinearRegression model: -1.866810585826352e+26
```

```
scr = cross_val_score(DTR, X_scaled, y, cv=5, scoring= 'r2')
print("Cross validation score for Decision Tree Regression:" , scr.mean())
```

```
Cross validation score for Decision Tree Regression: 0.12819678429212306
```

```
scr = cross_val_score(RFR, X_scaled, y, cv=5, scoring= 'r2')
print("Cross validation score for RandomForestRegressor:" , scr.mean())
```

```
Cross validation score for RandomForestRegressor: 0.4512364401833187
```

```
scr = cross_val_score(SV, X_scaled, y, cv=5, scoring= 'r2')
print("Cross validation score for SupportVectorRegressor:" , scr.mean())
```

```
Cross validation score for SupportVectorRegressor: -0.023820166542287226
```

```
scr = cross_val_score(ADR, X_scaled, y, cv=5, scoring= 'r2')
print("Cross validation score for AdaBoostRegressor:" , scr.mean())
```

```
Cross validation score for AdaBoostRegressor: -0.23131379706491506
```

```
scr = cross_val_score(GBR, X_scaled, y, cv=5, scoring= 'r2')
print("Cross validation score for GradientBoostingRegressor:" , scr.mean())
```

```
Cross validation score for GradientBoostingRegressor: 0.4457105644122473
```

```
scr = cross_val_score(BR, X_scaled, y, cv=5, scoring= 'r2')
print("Cross validation score for BaggingRegressor:" , scr.mean())
```

```
Cross validation score for BaggingRegressor: 0.42983869073386566
```

```
scr = cross_val_score(ls, X_scaled, y, cv=5)
print("Cross validation score for Lasso model:" , scr.mean())
```

```
Cross validation score for Lasso model: 0.39709038824088666
```

```
scr = cross_val_score(rd, X_scaled, y, cv=5)
print("Cross validation score for Ridge model:" , scr.mean())
```

```
Cross validation score for Ridge model: 0.39707566793857163
```

Evaluation Metrics

```
#RandomForestRegressor
RFR= RandomForestRegressor()
RFR.fit(x_train,y_train)
pred_test=RFR.predict(x_test)
print("MSE:",mean_squared_error(y_test,pred_test))
print("RMSE:",math.sqrt(mean_squared_error(y_test,pred_test)))
print("MAE:",mean_absolute_error(y_test,pred_test))
```

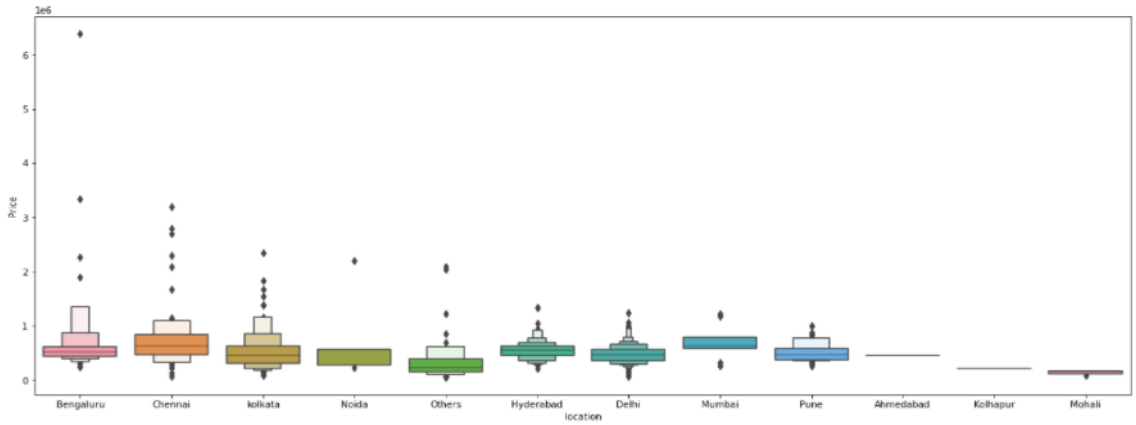
```
MSE: 22203205739.806637
RMSE: 149007.40162759242
MAE: 98691.46718085108
```

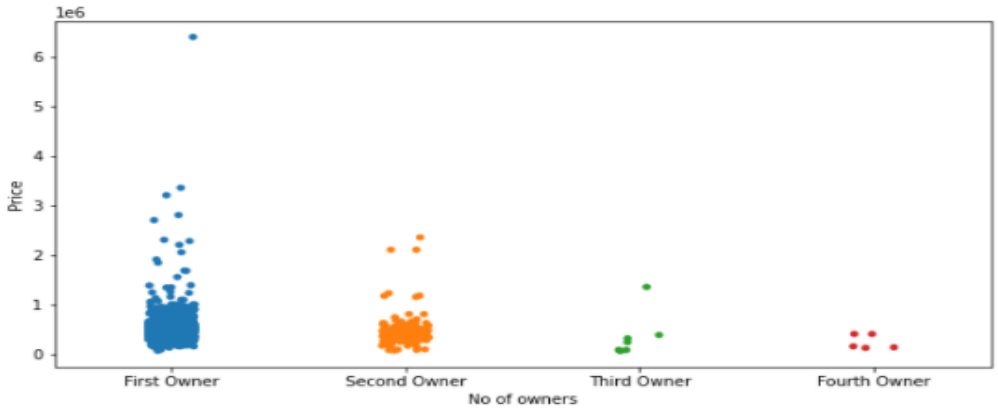
- Key Metrics for success in solving problem under consideration

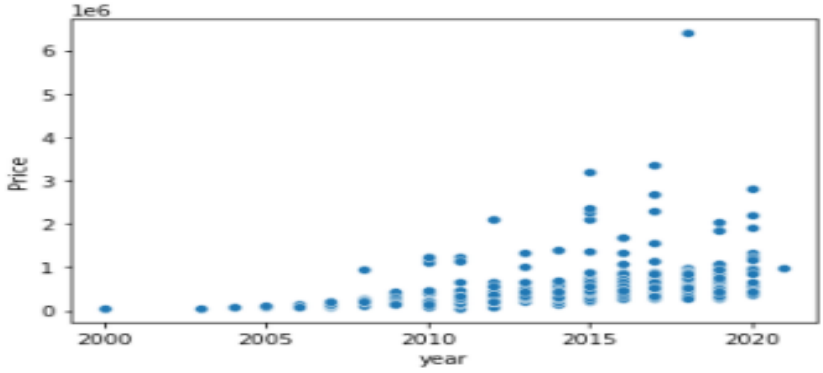
  I have used above shown screenshot of key metrics for model building & it helped me to understand why out of 8 models, I choose the best model based on the metrics such as R2 score, mean_squared_error, mean_absolute_error, root_mean_squared_error, Cross validation & hyper parameter tuning to increase the accuracy.

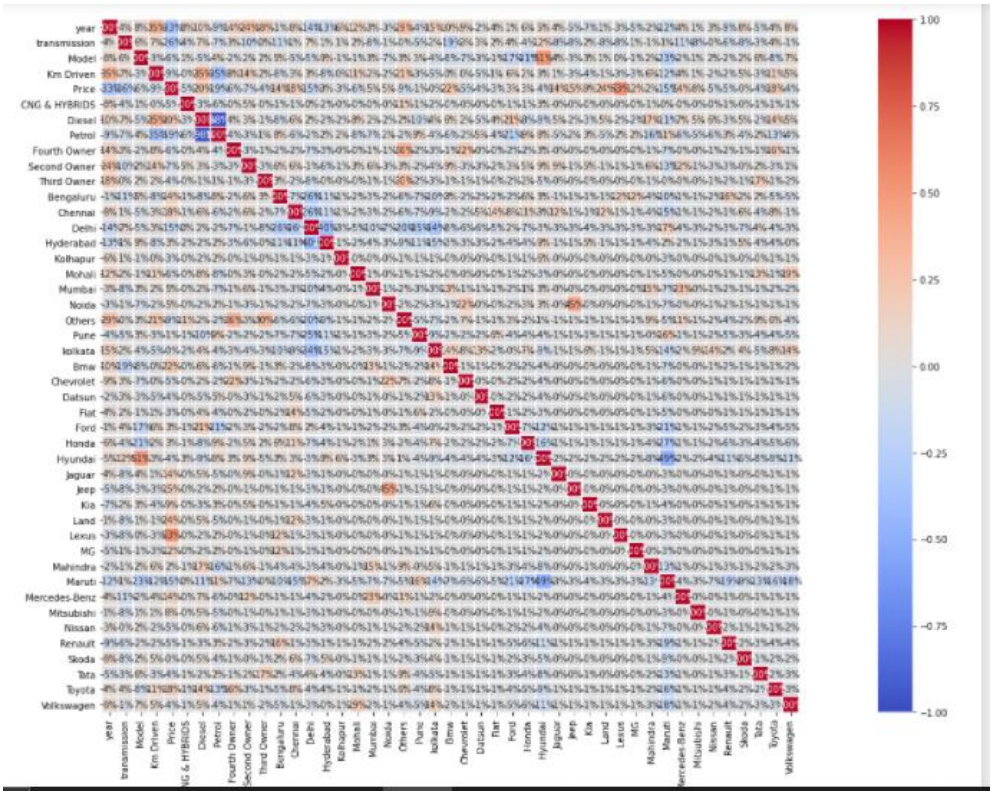
- Visualizations:

  Boxen plot, Strip plot, heat map & Scatter plot helped me to understand the correlation of the feature & with the label.

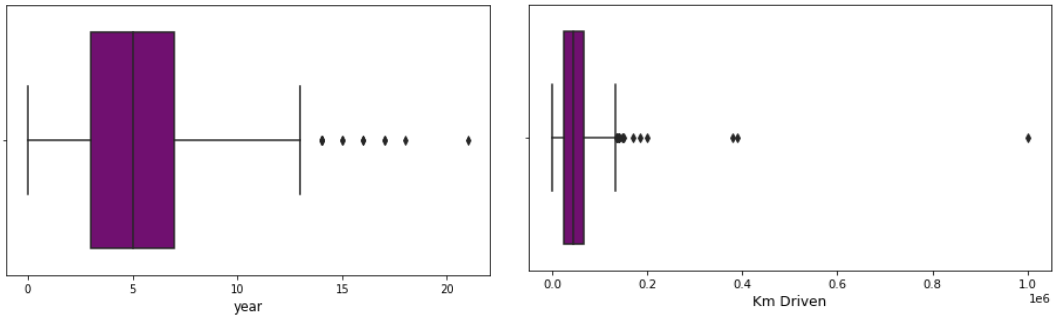




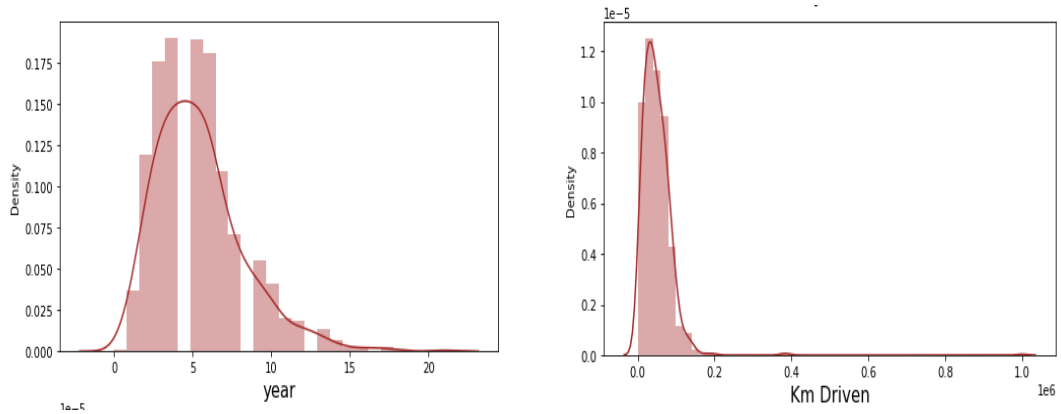




For outlier’s detection

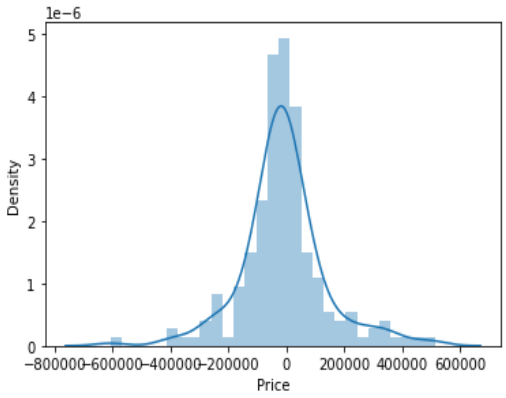


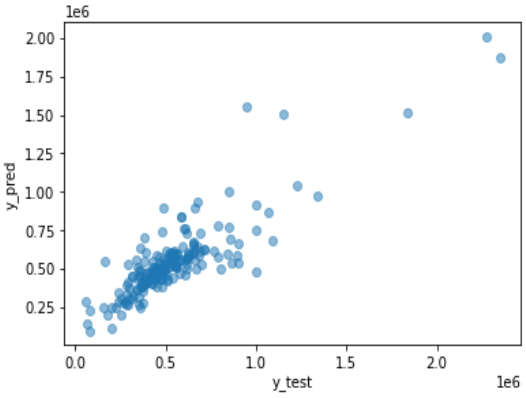
For skewness detection



- Interpretation of the Results

Visualizations: It helped me to understand the correlation between independent & dependent features. Also, helped me with feature selection & to check for multi collinearity issues. Detected outliers/skewness with the help of boxplot & distribution plot. I got to know the count of a particular category for each feature by using count plot & most importantly with predicted target value distribution & Scatter plot helped me to select the best model.





Pre-processing: Basically before building the model the dataset should be cleaned & scaled by performing few steps, which I mentioned above in the Pre-processing steps where all the important features are present in the data set and ready for model building.

Model Creation: Now, after performing the train test split, I have x_train, x_test, y_train & y_test, which are required to build Machine learning models. I have built multiple regression models to get the best R2 score, MSE, RMSE & MAE out of all the models.

CONCLUSION

- Key Findings and Conclusions of the Study

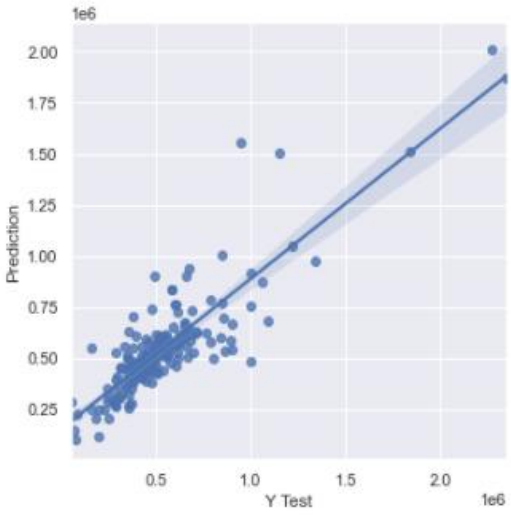
After performing the model building, I have got the highest score for RandomForestRegressor with 89% R2 score having less MSE, RMSE & MAE error as compared to other models but it could be due to overfitting, so I have checked for the cross validation scores & found 46% score, which gives very less difference between R2 score & CV

score. Hence, based on Cross Validation Score, R2 score & having less MSE, RMSE & MAE error, I have got the best fit model is RandomForestRegressor. Now let's perform hyper parameter tuning & check if I could increase the accuracy.

Findings: After using hyper parameter tuning, I observe accuracy could not increase. So let's keep original accuracy of Random Forest Regressor, which gives 89% score with less MSE & MAE error out of other models.

Let's predict & compare the results:

	Y Test	Prediction
183	622203	573886.47
210	515000	580685.60
714	617500	497961.16
467	490928	538967.00
656	378500	401164.64



Concluding Remarks

1) Saving the model: The model is ready & we have saved the model in 'pkl' format by using "joblib".

Final conclusion:

As we have seen, the prediction is showing almost similar relationship with the actual price from the train data set, which means the model predicted correctly & this could help the client to predict the price of the used cars & prospective to sell accordingly.

- Learning Outcomes of the Study in respect of Data Science

1) Visualization helped me to understand the data as it provides graphical representation of huge data, which helped me to understand the feature importance, outlier's/skewness detection & to compare the independent & dependent features.

2) Data cleaning is the most important part of model building as before model building, I make sure the data is cleaned & scaled.
3) I have performed multiple algorithms to get the best model, and found Random Forest regressor is the best fit model out of all the algorithms based on the metrics I have performed.

4) The challenges I faced while working on this project is when I was scrapping the real time data from olx & cardekho websites, it took days to gather the data because my target was to gather 5k data, which is huge & because of that my system failed. To overcome this, I scrap less data & start working on the project. The data I scrapped was noisy & cleaning part was challenging for me but to fix this, I did few manual cleaning by using excel functions, which took less time & it worked well to clean the data. Also, to tune the model, I used randomizedsearchCV instead of gridsearchCV, which took less execution time, however tuning the model did not help as score was not increased. So, I kept the original score & save the model.