FLIP ROBO

CAR PRICE PREDICTION

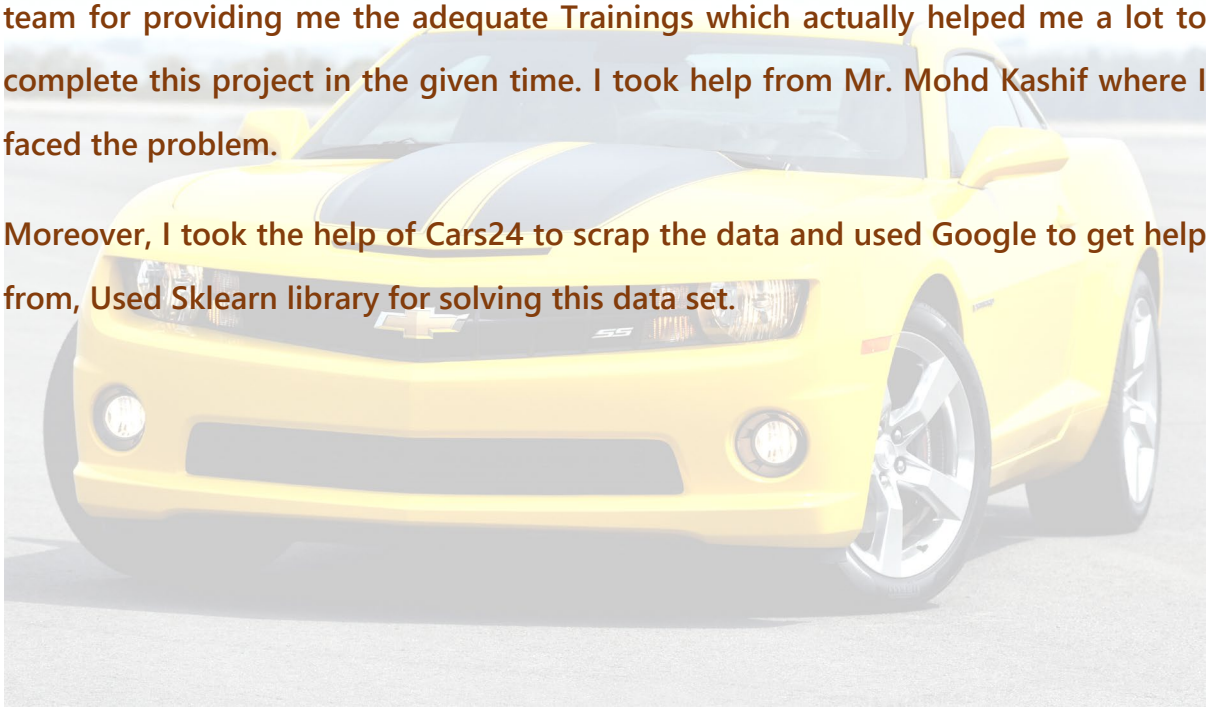
SUBMITTED BY

Abhijit Sarkar

ACKNOWLEDGMENT

I would like to express my gratitude for the opportunity and would like to thank Flip Robo Technologies for giving me an opportunity to work on this Given project. While going through this project I could find the new modes of facts and a conceptual thinking's of the second-hand cars market behind the scene and what can be a good consumption of the customers. I am very grateful to DATA TRAINED team for providing me the adequate Trainings which actually helped me a lot to complete this project in the given time. I took help from Mr. Mohd Kashif where I faced the problem.

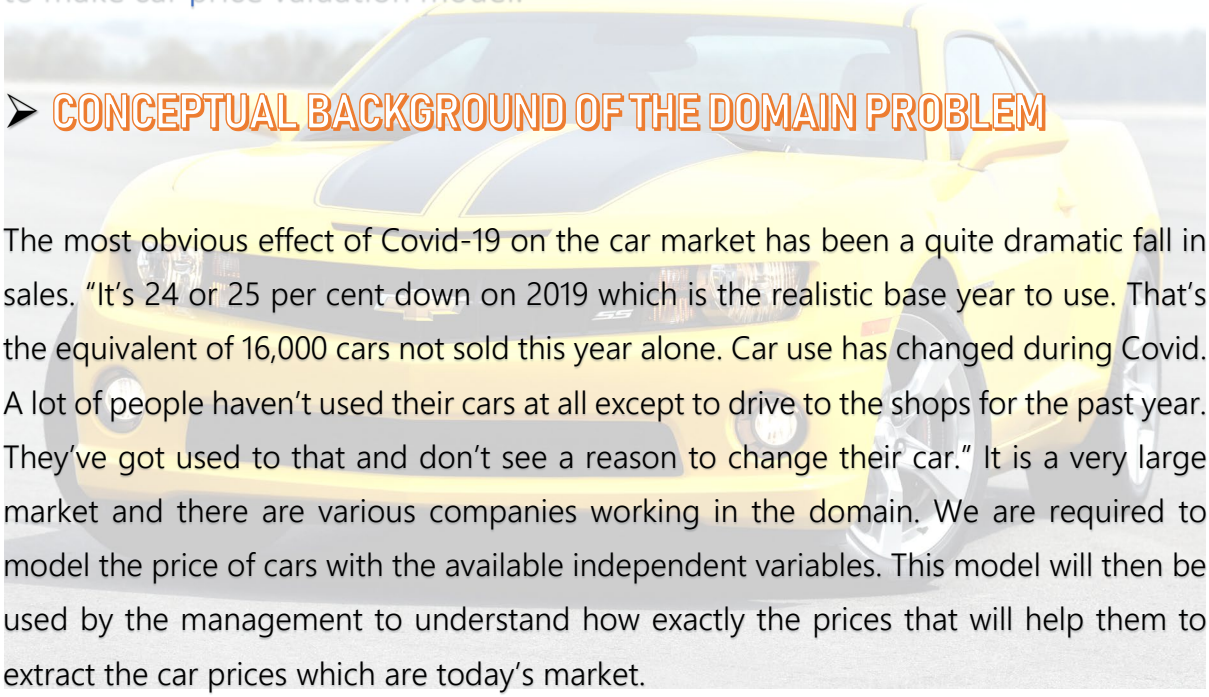
Moreover, I took the help of Cars24 to scrap the data and used Google to get help from, Used Sklearn library for solving this data set.



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 most obvious effect of Covid-19 on the car market has been a quite dramatic fall in sales. "It's 24 or 25 per cent down on 2019 which is the realistic base year to use. That's the equivalent of 16,000 cars not sold this year alone. Car use has changed during Covid. A lot of people haven't used their cars at all except to drive to the shops for the past year. They've got used to that and don't see a reason to change their car." It is a very large market and there are various companies working in the domain. We are required to model the price of cars with the available independent variables. This model will then be used by the management to understand how exactly the prices that will help them to extract the car prices which are today's market.

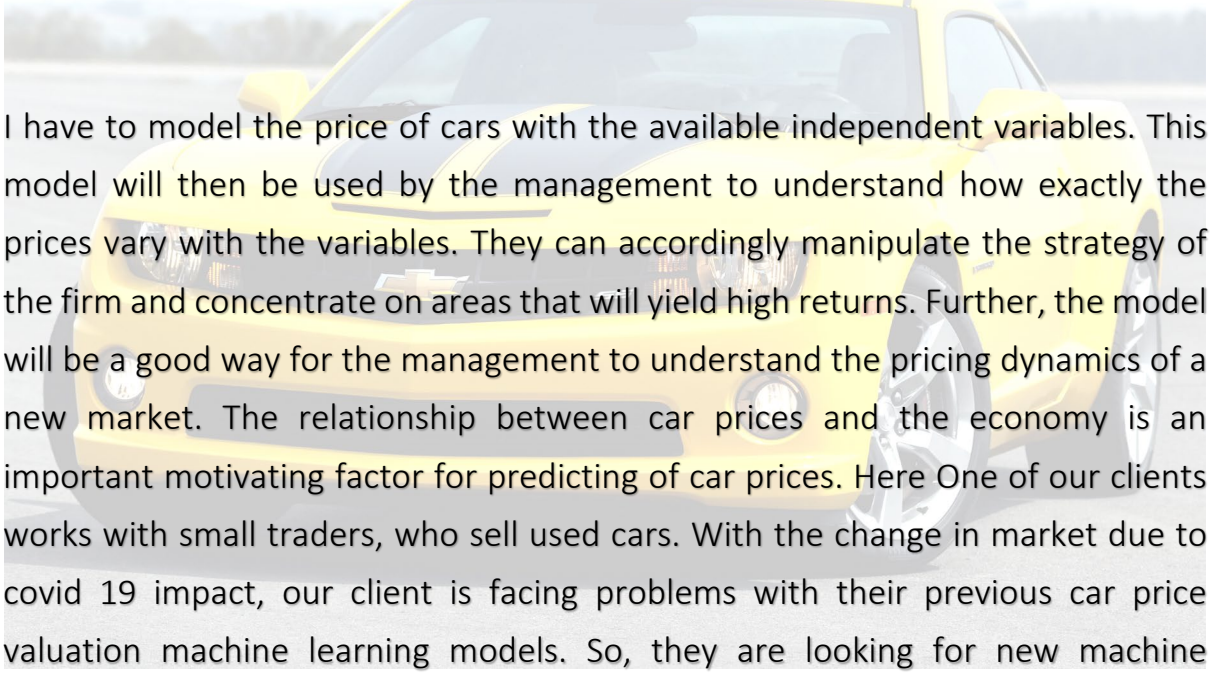
➢ REVIEW OF LITERATURE.

The factors that affect the Car price have to be studied and their impact on price has also to be modelled. An analysis of the present data is to be considered. Some cars are in demand hence making them costly whereas some are not in demand hence cheaper. Here I have scrapped more than 5000 data from cars24 and this are all saved in csv format, and that same data is used for further

analysed. We have used regression models, using various features to have lower Residual Sum of Squares error. While using features in a regression model some feature engineering is required for better prediction.

The primary aim of this report is to use these Machine Learning Techniques and accurate them into ML models which can then serve the users. The main objective for making this model is for one of 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.

➢ MOTIVATION FOR THE PROBLEM UNDERTAKEN



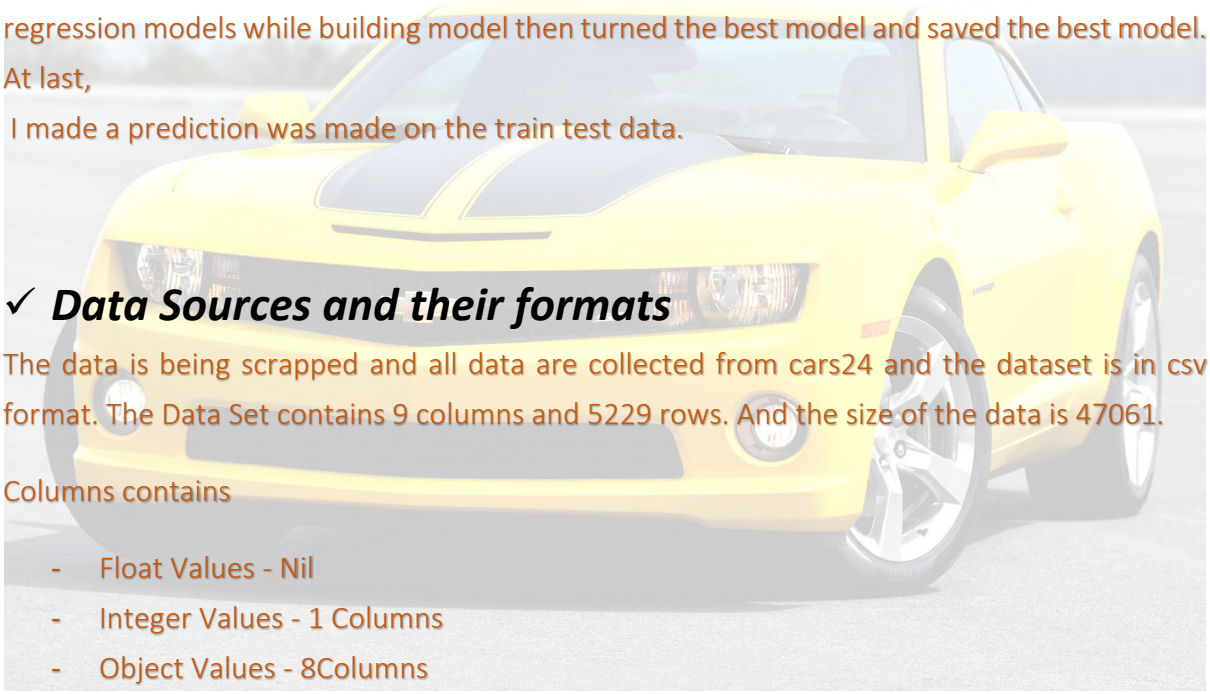
I have to model the price of cars with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market. The relationship between car prices and the economy is an important motivating factor for predicting of car prices. Here 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.

ANALYTICAL PROBLEM FRAMING

✓ *Mathematical/ Analytical Modelling of the Problem*

This dataset was scrapped by using selenium web driver and website scraped was car24, the dataset was saved in csv format and it consist of only one dataset, Target Variable is price so Linear Model was used to work with this dataset. Here Price which is a Target variable, kilometre data type was on string type, those were converted to integer by removing all coma. To get better insight on the features I have used plotting like count plot, hist plot, scatter plot and cat plot. With these plotting I was able to understand the relation between the features in better manner. Also, I used box plot to find outliers and Finally I used Distribution plot to check the skewness in the dataset, in data cleansing, I removed outliers using zscore method and I removed skewness using power transform method or yeo-johnson method. I have used all the regression models while building model then turned the best model and saved the best model. At last,

I made a prediction was made on the train test data.



✓ *Data Sources and their formats*

The data is being scrapped and all data are collected from cars24 and the dataset is in csv format. The Data Set contains 9 columns and 5229 rows. And the size of the data is 47061.

Columns contains

- Float Values - Nil
- Integer Values - 1 Columns
- Object Values - 8Columns
- Memory Used: 388KB

✓ *Data Pre-processing*

These Steps:

1. Loading Data Set, Locating Data types
2. Finding the unique values in categorical and numerical columns.
3. Finding Data Missing percentage
4. Finding nan values

5. Finding duplicated values in columns rows.

6. Use encoding was not required.

7. Feature Extraction was done here.

✓ Data Inputs- Logic- Output Relationships

X variables plays a very import role in machine learning for the Prediction of Target variable. Here 'Price 'is the target variable on which the predictions are being made.

I used the following to determine the relationship between variable:

- I have used Catplot for each pair of categorical features and hist plot to find the relation with the Target Variable.
- Used Box plot to check outliers.
- I used Distplot to determine the skewness.
- For continuous numerical variables, I have used scatterplot to check the relationship between a continuous numerical variable and target variable.
- Used univariate, Bivariate, Multi-variate Graph to check for relations.

By the Use of these Graph, I Uncovered the is a relationship between continuous numerical variable and The Target Variable.

✓ Hardware and Software Requirements and Tools Used

Hardware:

SOFTWARE USED:

- I Used Jupiter Note Book.
- Microsoft Office 2020
- Windows 11 OS

Library used: To run the program and to build the model we need some basic libraries as follows:

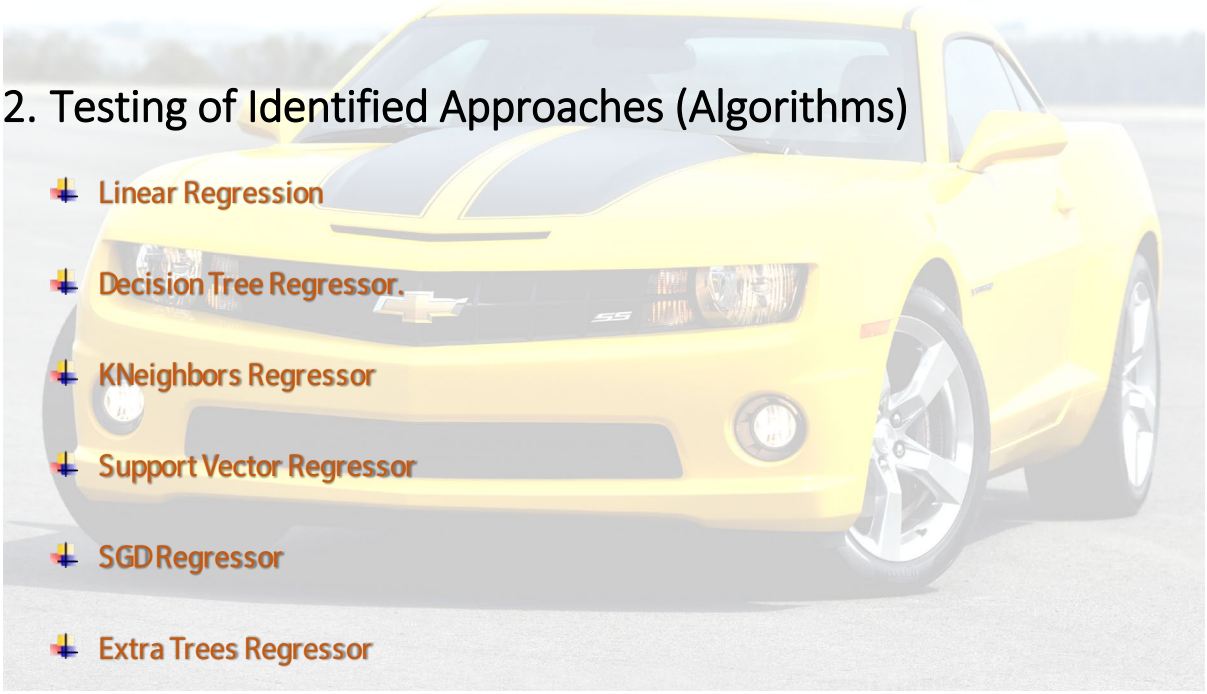
- NumPy
- Pandas
- Seaborn
- Matplotlib
- SciPy
- Sklearn
- Pickle
- Imbalance Learn

1) import pandas as pd:
   Pandas are a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a NumPy matrix array. This makes pandas a trusted ally in data science and machine learning.
2) import NumPy as np:
   NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.
3) import seaborn as sns:
   Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas' data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.
4) Import matplotlib.pyplot as plt:
   matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

Model/s Development and Evaluation.

1. Identification of possible problem-solving approaches (methods)

- To check outliers, I have used Boxplot and to remove outliers with the help of Zscore.
- To check skewness, I used distplot. And removed skewness with the help of have yeo-johnson Methods.
- Use of Pearson's correlation coefficient to check the correlation between dependent and independent features.
- Also, I have used standard Scaler for Standardization.
- Then followed by model building with all Classifiers and Linear Regression algorithms.



2. Testing of Identified Approaches (Algorithms)

- Linear Regression
- Decision Tree Regressor.
- KNeighbors Regressor
- Support Vector Regressor
- SGD Regressor
- Extra Trees Regressor
- Random Forest Regressor
- Ada Boost Regressor
- Gradient Boosting Regressor
- MLP Regressor
- Bagging Regressor

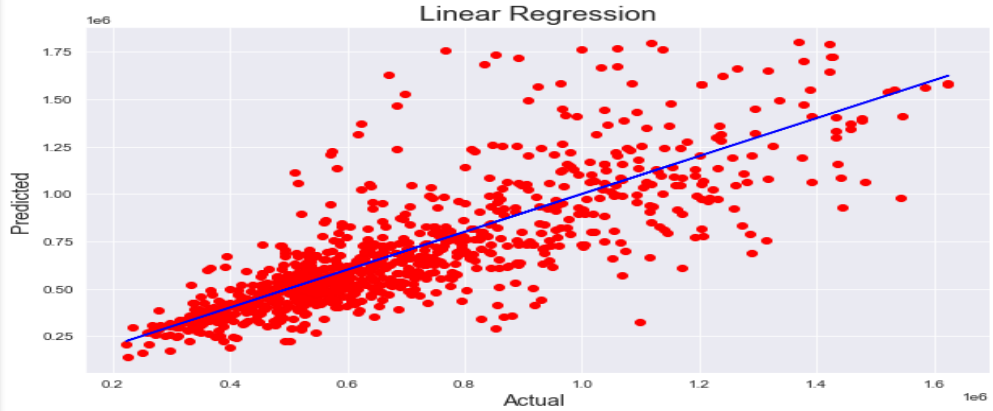
3. Run and evaluate selected models.

Linear Model

```
In [77]: # Train Test
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=98,test_size=0.3)

#training
lr.fit(x_train,y_train)
lr_score=lr.score(x_train,y_train)

#predict
pred_train=lr.predict(x_train)
pred_test=lr.predict(x_test)

#result
print("Training Accuracy r2_score ",r2_score(y_train,pred_train)*100,"Testing Accuracy R2_score ",r2_score(y_test,pred_test)*100)
print("Training Mean_squared_Error ",mean_squared_error(y_train,pred_train),"Testing Mean_squared_error ",mean_squared_error(y_te
print("Training Absolute_Error ",mean_absolute_error(y_train,pred_train),"Testing Absolute Error",mean_absolute_error(y_test,pred

Training Accuracy r2_score  38.22199831106512 Testing Accuracy R2_score  38.48161907368587
Training Mean_squared_Error  69712783172.61488 Testing Mean_squared_error  66416503738.617905
Training Absolute_Error  195491.8374957711 Testing Absolute Error 189567.41704512225
```

2.KNeighbors Classifier:

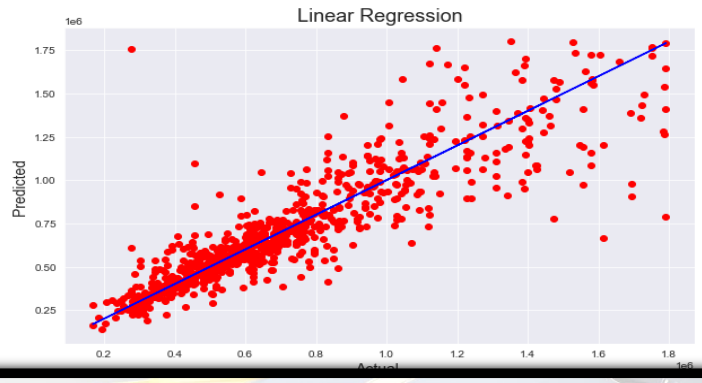
```
Model KNeighborsRegressor()
 ******************************************************
Training Accuracy r2_score  0.7487929814107104 Testing Accuracy R2_score  0.6415280642695963
Training Mean_squared_Error  28347210883.46701 Testing Mean_squared_error  38701364212.01557
Training Absolute_Error  109229.39489051094 Testing Absolute Error 129971.28425531913

 KNeighborsRegressor() Model Cross Validation

CROSS_FOLD 2, CV_Score0.53 Train Score0.75 Testing Score0.64
CROSS_FOLD 3, CV_Score0.52 Train Score0.75 Testing Score0.64
CROSS_FOLD 4, CV_Score0.59 Train Score0.75 Testing Score0.64
CROSS_FOLD 5, CV_Score0.59 Train Score0.75 Testing Score0.64
CROSS_FOLD 6, CV_Score0.6 Train Score0.75 Testing Score0.64
CROSS_FOLD 7, CV_Score0.6 Train Score0.75 Testing Score0.64
CROSS_FOLD 8, CV_Score0.61 Train Score0.75 Testing Score0.64
CROSS_FOLD 9, CV_Score0.59 Train Score0.75 Testing Score0.64

Model Graph
```




```
******************************************************
```

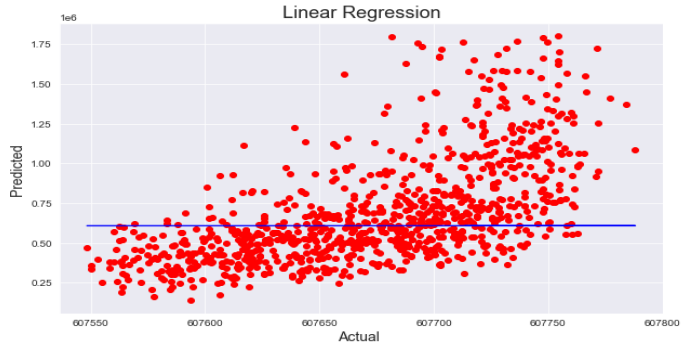
Model-Decision Tree:

```
Model DecisionTreeRegressor()
 ****************************************************
click to scroll output; double click to hide
                                    09069185 Testing Accuracy R2_score  0.7779222400593963
                                    9306569345 Testing Mean_squared_error  23975969704.121277
Training Absolute_Error  52.18978102189781 Testing Absolute Error 91347.56170212766

 DecisionTreeRegressor() Model Cross Validation

CROSS_FOLD 2, CV_Score0.68 Train Score1.0 Testing Score0.78
CROSS_FOLD 3, CV_Score0.67 Train Score1.0 Testing Score0.78
CROSS_FOLD 4, CV_Score0.73 Train Score1.0 Testing Score0.78
CROSS_FOLD 5, CV_Score0.75 Train Score1.0 Testing Score0.78
CROSS_FOLD 6, CV_Score0.78 Train Score1.0 Testing Score0.78
CROSS_FOLD 7, CV_Score0.75 Train Score1.0 Testing Score0.78
CROSS_FOLD 8, CV_Score0.77 Train Score1.0 Testing Score0.78
CROSS_FOLD 9, CV_Score0.76 Train Score1.0 Testing Score0.78

Model Graph
```



Model-SVC Regressor

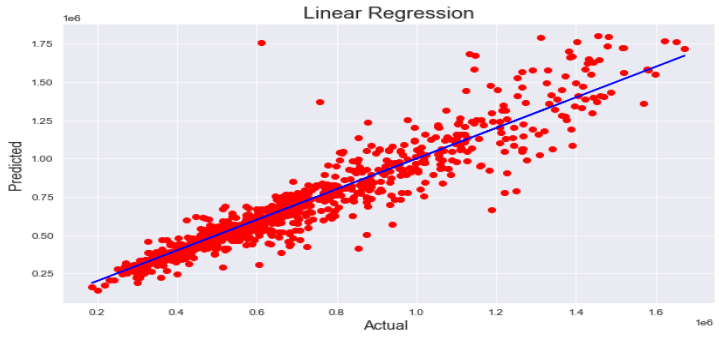
```
Model SVR()
 ****************************************************
Training Accuracy r2_score  -0.06073879999711118 Testing Accuracy R2_score  -0.06324682907555834
Training Mean_squared_Error  119698034810.70358 Testing Mean_squared_error  114790304840.68604
Training Absolute_Error  248567.40308464487 Testing Absolute Error 242519.29237371086

 SVR() Model Cross Validation

CROSS_FOLD 2, CV_Score-0.08 Train Score-0.06 Testing Score-0.06
CROSS_FOLD 3, CV_Score-0.09 Train Score-0.06 Testing Score-0.06
CROSS_FOLD 4, CV_Score-0.09 Train Score-0.06 Testing Score-0.06
CROSS_FOLD 5, CV_Score-0.08 Train Score-0.06 Testing Score-0.06
CROSS_FOLD 6, CV_Score-0.08 Train Score-0.06 Testing Score-0.06
CROSS_FOLD 7, CV_Score-0.09 Train Score-0.06 Testing Score-0.06
CROSS_FOLD 8, CV_Score-0.09 Train Score-0.06 Testing Score-0.06
CROSS_FOLD 9, CV_Score-0.09 Train Score-0.06 Testing Score-0.06

Model Graph
```




```
****************************************************
```

Model-Random Forest Regressor:

```
Model RandomForestRegressor()
 ******************************************************
Training Accuracy r2_score  0.9833552822521074 Testing Accuracy R2_score  0.8860445343773081
Training Mean_squared_Error  1878256932.2504337 Testing Mean_squared_error  12302865411.284037
Training Absolute_Error  26519.064984358705 Testing Absolute Error 68184.23780420466

 RandomForestRegressor() Model Cross Validation

CROSS_FOLD 2, CV_Score0.84 Train Score0.98 Testing Score0.89
CROSS_FOLD 3, CV_Score0.84 Train Score0.98 Testing Score0.89
CROSS_FOLD 4, CV_Score0.86 Train Score0.98 Testing Score0.89
CROSS_FOLD 5, CV_Score0.87 Train Score0.98 Testing Score0.89
CROSS_FOLD 6, CV_Score0.88 Train Score0.98 Testing Score0.89
CROSS_FOLD 7, CV_Score0.87 Train Score0.98 Testing Score0.89
CROSS_FOLD 8, CV_Score0.87 Train Score0.98 Testing Score0.89
CROSS_FOLD 9, CV_Score0.87 Train Score0.98 Testing Score0.89

Model Graph
```



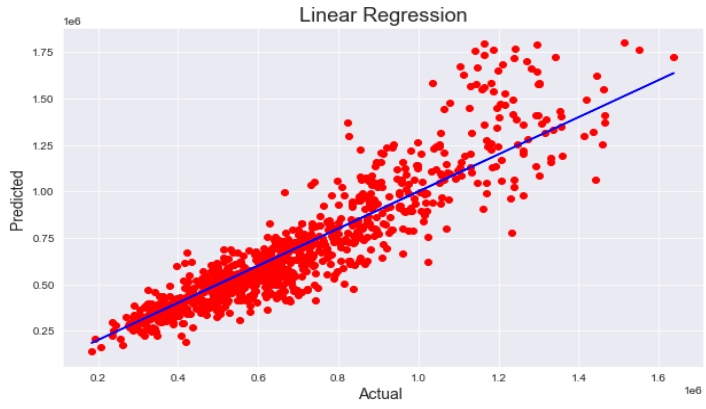

```
******************************************************
```

Model- Gradient Boosting Regressor

```
Model GradientBoostingRegressor()
 ******************************************************
Training Accuracy r2_score  0.8532139694449646 Testing Accuracy R2_score  0.8215983326929646
Training Mean_squared_Error  16563926383.337227 Testing Mean_squared_error  19260609309.379803
Training Absolute_Error  90869.21160261428 Testing Absolute Error 97513.06353135273
```

click to scroll output; double click to hide Cross Validation

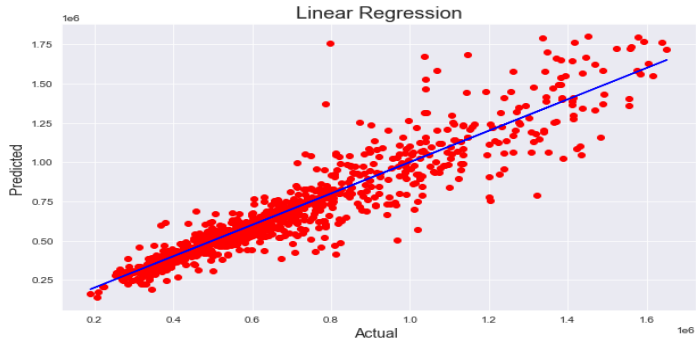
```
CROSS_FOLD 2, CV_Score0.8 Train Score0.85 Testing Score0.82
CROSS_FOLD 3, CV_Score0.78 Train Score0.85 Testing Score0.82
CROSS_FOLD 4, CV_Score0.81 Train Score0.85 Testing Score0.82
CROSS_FOLD 5, CV_Score0.8 Train Score0.85 Testing Score0.82
CROSS_FOLD 6, CV_Score0.81 Train Score0.85 Testing Score0.82
CROSS_FOLD 7, CV_Score0.8 Train Score0.85 Testing Score0.82
CROSS_FOLD 8, CV_Score0.8 Train Score0.85 Testing Score0.82
CROSS_FOLD 9, CV_Score0.8 Train Score0.85 Testing Score0.82

Model Graph
```




```
******************************************************
```

Model- Bagging Regressor

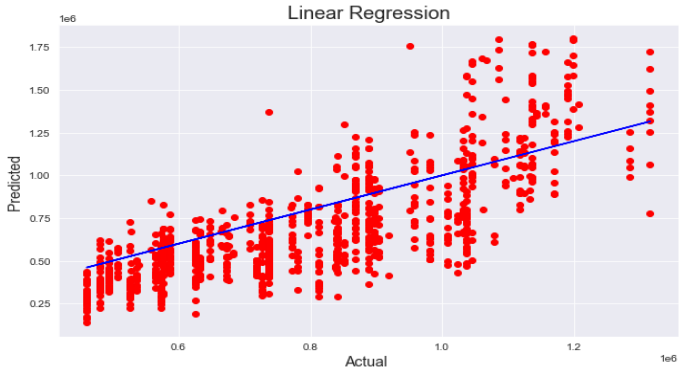
```
Model BaggingRegressor()
 ****************************************************
Training Accuracy r2_score  0.9734121367006785 Testing Accuracy R2_score  0.8673469259584816
Training Mean_squared_Error  3000281489.4233 Testing Mean_squared_error  14321497502.625408
Training Absolute_Error  31462.53020072993 Testing Absolute Error 75118.28964539008

 BaggingRegressor() Model Cross Validation

CROSS_FOLD 2, CV_Score0.82 Train Score0.97 Testing Score0.87
CROSS_FOLD 3, CV_Score0.82 Train Score0.97 Testing Score0.87
CROSS_FOLD 4, CV_Score0.84 Train Score0.97 Testing Score0.87
CROSS_FOLD 5, CV_Score0.85 Train Score0.97 Testing Score0.87
CROSS_FOLD 6, CV_Score0.86 Train Score0.97 Testing Score0.87
CROSS_FOLD 7, CV_Score0.86 Train Score0.97 Testing Score0.87
CROSS_FOLD 8, CV_Score0.86 Train Score0.97 Testing Score0.87
CROSS_FOLD 9, CV_Score0.86 Train Score0.97 Testing Score0.87

Model Graph
```




```
****************************************************
```

Model- ADA Boost Regressor

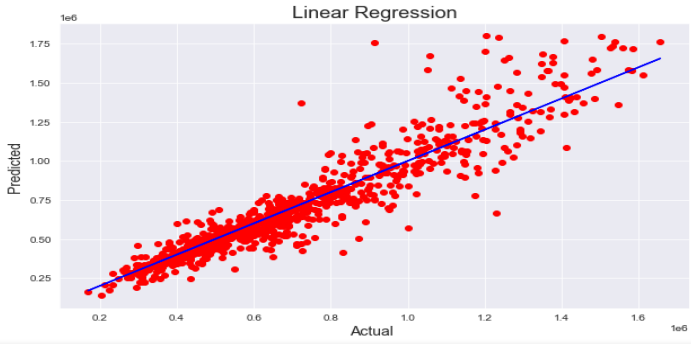
```
Model AdaBoostRegressor()
 ****************************************************
Training Accuracy r2_score  0.5102857445509394 Testing Accuracy R2_score  0.49506236259806846
Training Mean_squared_Error  55261327290.18592 Testing Mean_squared_error  54514101277.215775
Training Absolute_Error  191082.71967505268 Testing Absolute Error 187358.34572951964

 AdaBoostRegressor() Model Cross Validation

CROSS_FOLD 2, CV_Score0.47 Train Score0.51 Testing Score0.5
CROSS_FOLD 3, CV_Score0.39 Train Score0.51 Testing Score0.5
CROSS_FOLD 4, CV_Score0.4 Train Score0.51 Testing Score0.5
CROSS_FOLD 5, CV_Score0.44 Train Score0.51 Testing Score0.5
CROSS_FOLD 6, CV_Score0.42 Train Score0.51 Testing Score0.5
CROSS_FOLD 7, CV_Score0.43 Train Score0.51 Testing Score0.5
CROSS_FOLD 8, CV_Score0.42 Train Score0.51 Testing Score0.5
CROSS_FOLD 9, CV_Score0.41 Train Score0.51 Testing Score0.5

Model Graph
```




```
****************************************************
```

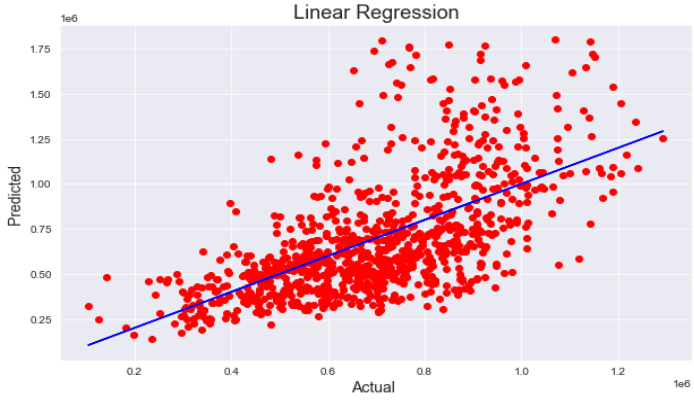
Model- Extra Trees Regressor

```
Model ExtraTreesRegressor()
 ****************************************************
Training Accuracy r2_score  0.9999961409069185 Testing Accuracy R2_score  0.8823781776887821
Training Mean_squared_Error  435475.59306569345 Testing Mean_squared_error  12698692786.848852
Training Absolute_Error  52.18978102189781 Testing Absolute Error 69789.10258510639

 ExtraTreesRegressor() Model Cross Validation

CROSS_FOLD 2, CV_Score0.82 Train Score1.0 Testing Score0.88
CROSS_FOLD 3, CV_Score0.82 Train Score1.0 Testing Score0.88
CROSS_FOLD 4, CV_Score0.86 Train Score1.0 Testing Score0.88
CROSS_FOLD 5, CV_Score0.86 Train Score1.0 Testing Score0.88
CROSS_FOLD 6, CV_Score0.87 Train Score1.0 Testing Score0.88
CROSS_FOLD 7, CV_Score0.87 Train Score1.0 Testing Score0.88
CROSS_FOLD 8, CV_Score0.87 Train Score1.0 Testing Score0.88
CROSS_FOLD 9, CV_Score0.87 Train Score1.0 Testing Score0.88

Model Graph
```



Model- SGD Regressor

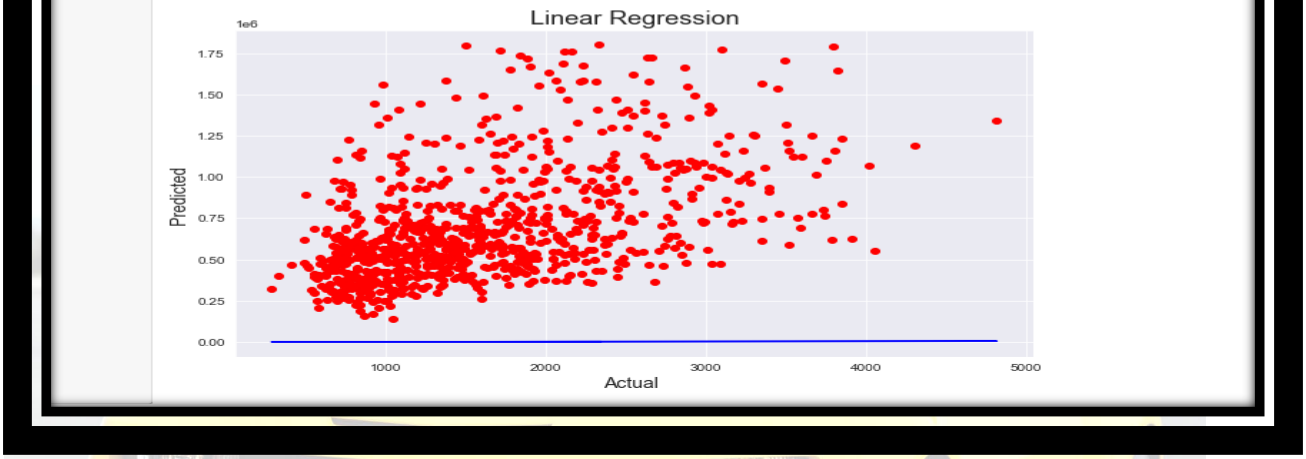
```
Model SGDRegressor()
 ****************************************************
Training Accuracy r2_score  0.3818770385686614 Testing Accuracy R2_score  0.38469358063600945
Training Mean_squared_Error  69751482414.80849 Testing Mean_squared_error  66429740976.170044
Training Absolute_Error  195969.59366886131 Testing Absolute Error 190123.03563702392

 SGDRegressor() Model Cross Validation

CROSS_FOLD 2, CV_Score0.38 Train Score0.38 Testing Score0.38
CROSS_FOLD 3, CV_Score0.33 Train Score0.38 Testing Score0.38
CROSS_FOLD 4, CV_Score0.36 Train Score0.38 Testing Score0.38
CROSS_FOLD 5, CV_Score0.36 Train Score0.38 Testing Score0.38
CROSS_FOLD 6, CV_Score0.35 Train Score0.38 Testing Score0.38
CROSS_FOLD 7, CV_Score0.35 Train Score0.38 Testing Score0.38
CROSS_FOLD 8, CV_Score0.36 Train Score0.38 Testing Score0.38
CROSS_FOLD 9, CV_Score0.36 Train Score0.38 Testing Score0.38

Model Graph
```

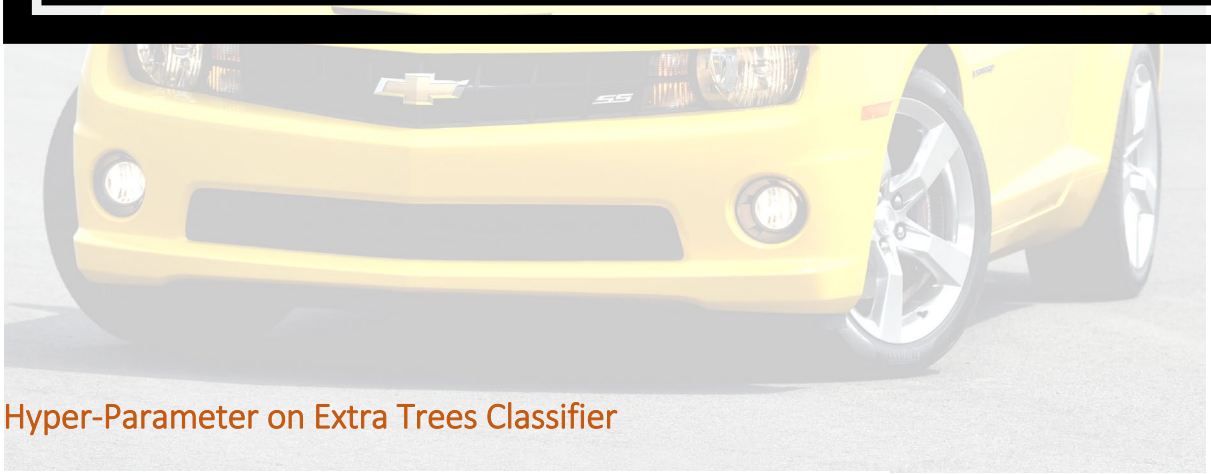



```
 ****************************************************
```

Model- MLP Regressor

```
Model MLPRegressor()
 ************************************************
Training Accuracy r2_score  -4.204740686493271 Testing Accuracy R2_score  -4.392662037637452
Training Mean_squared_Error  587323883951.6886 Testing Mean_squared_error  582202835950.5298
Training Absolute_Error  689019.9418788457 Testing Absolute Error 688842.042816311

 MLPRegressor() Model Cross Validation

CROSS_FOLD 2, CV_Score-4.28 Train Score-4.2 Testing Score-4.39
CROSS_FOLD 3, CV_Score-4.31 Train Score-4.2 Testing Score-4.39
CROSS_FOLD 4, CV_Score-4.32 Train Score-4.2 Testing Score-4.39
click to scroll output; double click to hide
CROSS_FOLD 7, CV_Score-4.41 Train Score-4.2 Testing Score-4.39
CROSS_FOLD 8, CV_Score-4.4 Train Score-4.2 Testing Score-4.39
CROSS_FOLD 9, CV_Score-4.45 Train Score-4.2 Testing Score-4.39

Model Graph
```





Hyper-Parameter on Extra Trees Classifier

Hyper-Parameter With Models

Extra Trees Regressor and Random Forest, Bagging Regressor are working very close

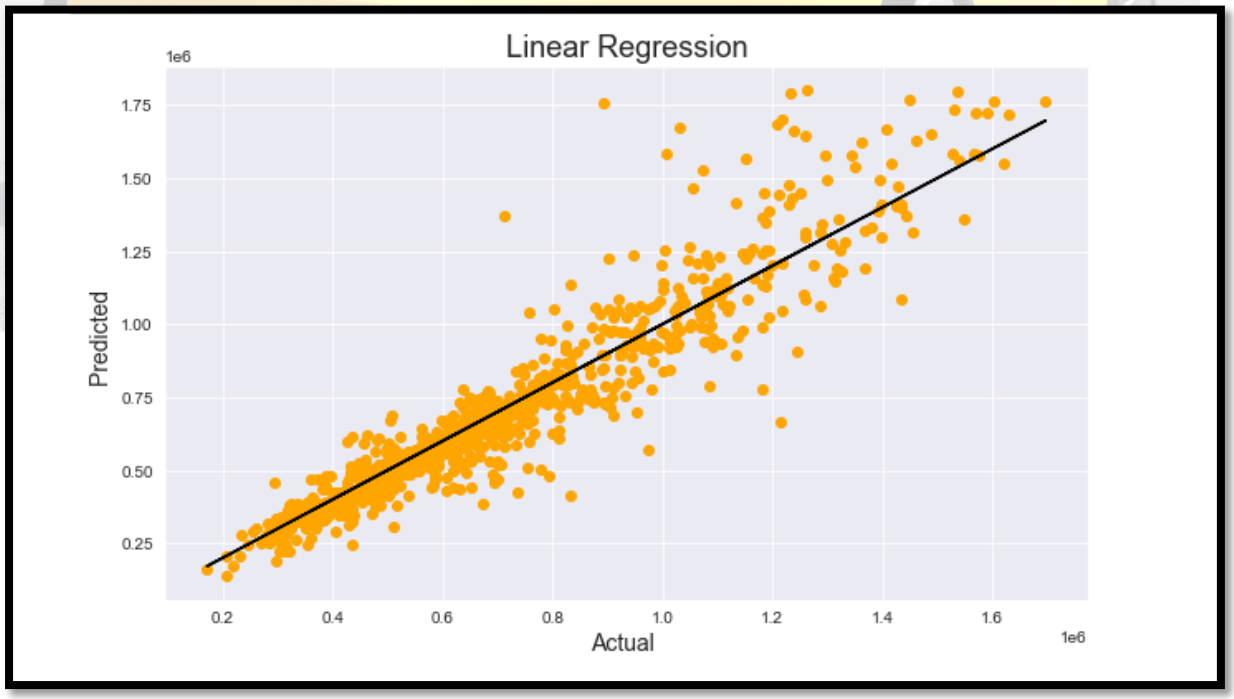
we selected them due the following reasons:

- R2 Score is better for the model.
- CV Score better for these model they are nearer to training score.
- "MSE" and "MAE" are less on compare to other models.(They are close to the mean)

Extra Trees Regressor

```
In [94]: from sklearn.ensemble import ExtraTreesRegressor
etr=ExtraTreesRegressor(criterion= 'absolute_error',n_estimators= 200,max_features= 1.0, n_jobs=2,
                        random_state=98,max_depth=2500,min_samples_split=2,min_impurity_decrease=0.00001)

#training score
etr.fit(x_train,y_train)
etr_score=etr.score(x_train,y_train)

#predict
pred_train1=etr.predict(x_train)
pred_test1=etr.predict(x_test)

#result
print("Training Score",etr_score)
print("Training Accuracy r2_score ",r2_score(y_train,pred_train1),"Testing Accuracy R2_score ",r2_score(y_test,pred_test1))
print("Training Mean_squared_Error ",mean_squared_error(y_train,pred_train1),"Testing Mean_squared_error ",mean_squared_error(y_t
print("Training Absolute_Error ",mean_absolute_error(y_train,pred_train1),"Testing Absolute Error",mean_absolute_error(y_test,pre

Training Score 0.9998415543291546
Training Accuracy r2_score  0.9998415543291546 Testing Accuracy R2_score  0.8859330307410067
Training Mean_squared_Error  17879647.11436941 Testing Mean_squared_error  12314903572.181269
Training Absolute_Error  1481.5264781021897 Testing Absolute Error 67509.89688829788
```

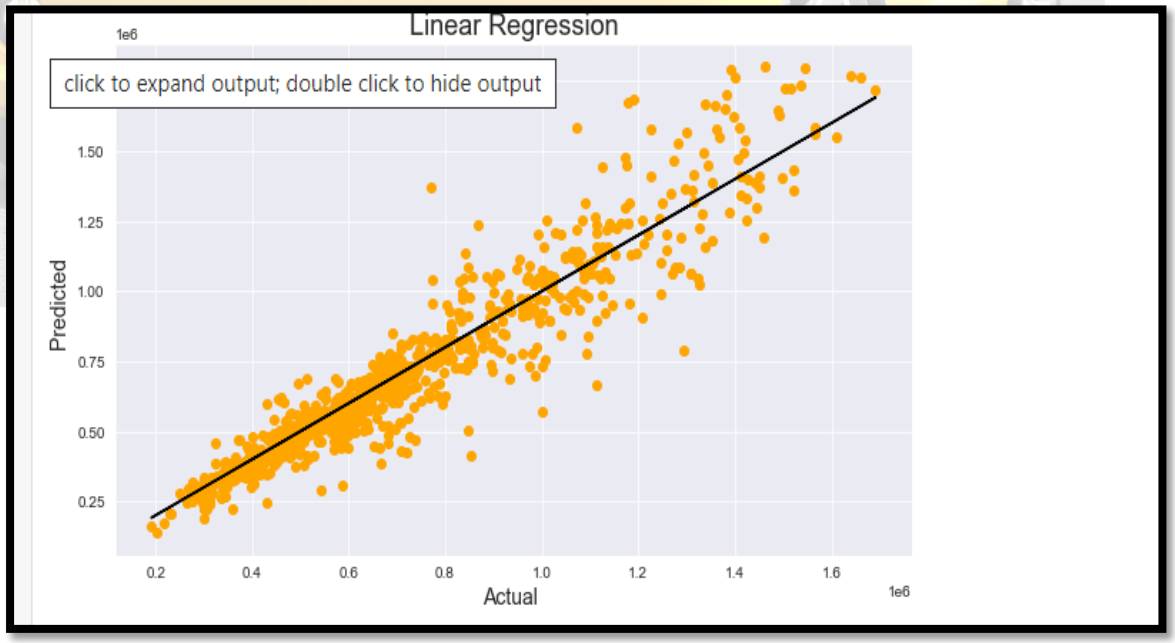
Graph:



Model: Random Forest

```
[140]: from sklearn.ensemble import RandomForestRegressor

rfr=RandomForestRegressor(criterion= 'squared_error',n_estimators= 100,max_features= 1.0,max_depth=200,
                          random_state=0,min_samples_split=2,min_impurity_decrease=0.000001,max_leaf_nodes=None)

#training score
rfr.fit(x_train,y_train)
rfr.score(x_train,y_train)

#predict
pred_train2=rfr.predict(x_train)
pred_test2=rfr.predict(x_test)

#result

print("Training Accuracy r2_score ",r2_score(y_train,pred_train2),"Testing Accuracy R2_score ",r2_score(y_test,pred_test2))
print("Training Mean_squared_Error ",mean_squared_error(y_train,pred_train2),"Testing Mean_squared_error ",mean_squared_error(y_t
print("Training Absolute_Error ",mean_absolute_error(y_train,pred_train2),"Testing Absolute Error",mean_absolute_error(y_test,pre

Training Accuracy r2_score  0.9823138575616585 Testing Accuracy R2_score  0.8874794820137715
Training Mean_squared_Error  1995775485.2159984 Testing Mean_squared_error  12147945526.158905
Training Absolute_Error  26930.725616853495 Testing Absolute Error 68380.40454964539
```

Graph:



Saving Model and Loading Model:

Saving Best Model

In [152]:
```
import pickle
filename='cars.pkl'
pickle.dump(etr,open(filename,'wb'))
```

Loading Back Model:

```
# Opening back
import pickle
model =pickle.load(open('cars.pkl','rb'))
result=model.score(x_test,y_test)
print(result*100)
```

```
88.59330307410067
```

```
Final_Result=pd.DataFrame([model.predict(x_test)[:],y_test[:]],index=["Predicted","Original"])

Final_Result
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
	i4.25	797704.75	388308.5	1014087.375	994614.5	802780.0	997324.0	540257.0	534513.25	779505.5	508721.25	1454861.5	974811.25	275016.0	918495.5	810
	)9.00	820899.00	390399.0	915999.000	1082499.0	826699.0	971799.0	545149.0	488399.00	504099.0	486999.00	1316099.0	1053599.00	253799.0	800599.0	610

Key Metrics for success in solving problem under consideration.

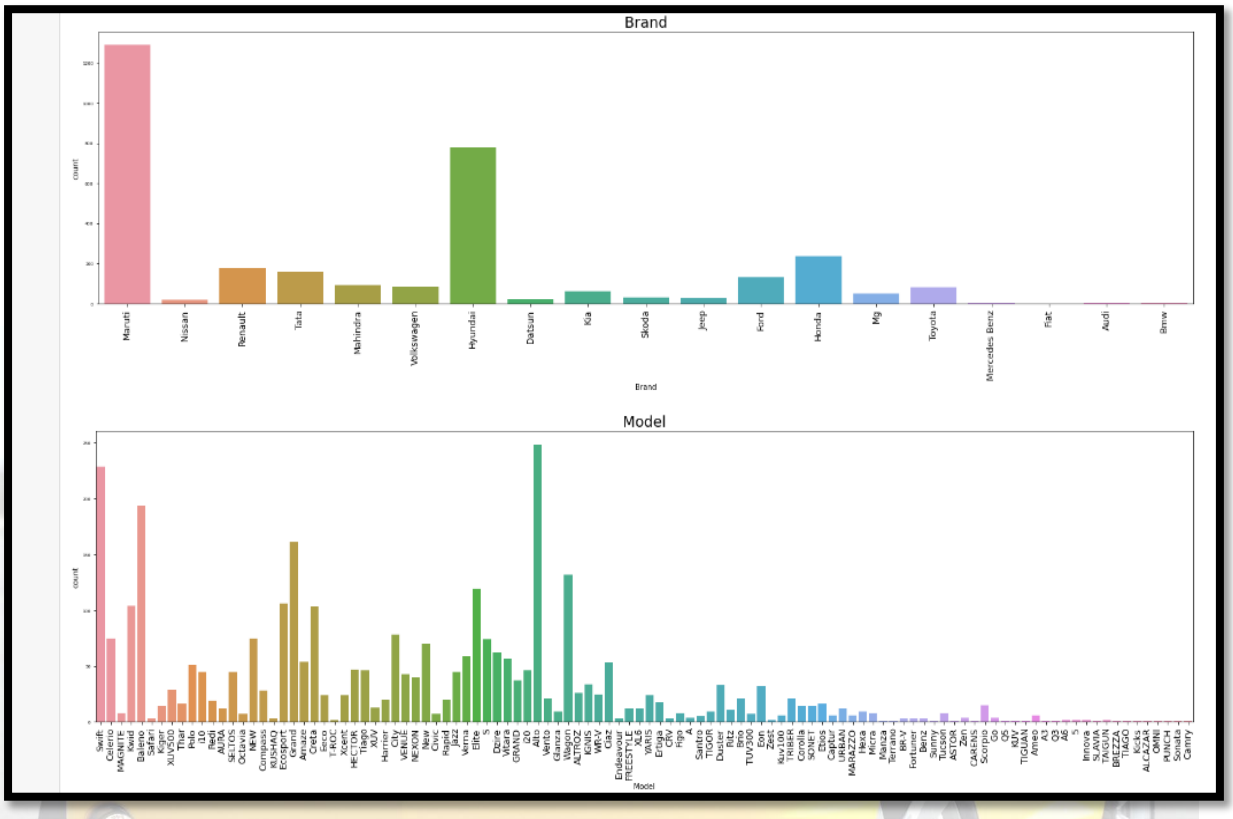
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score,mean_squared_error, mean_absolute_error
```

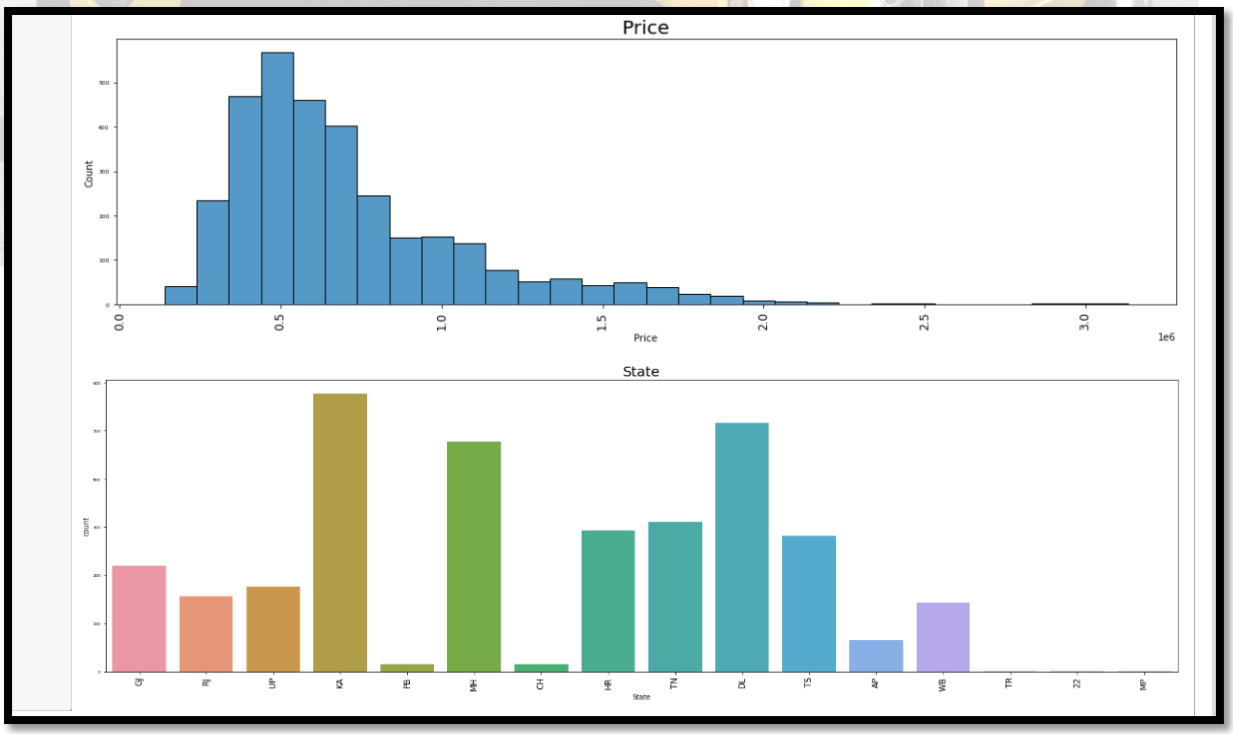
The Following Metrics were Used:

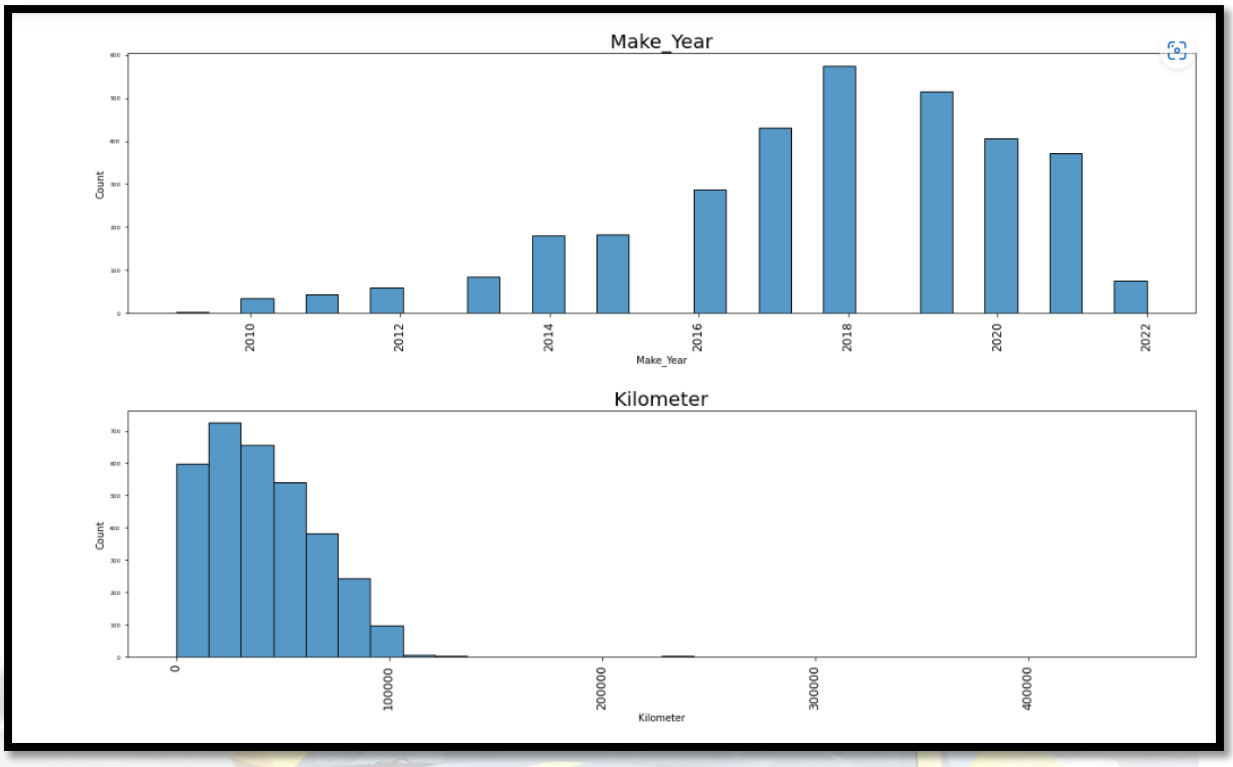
1. **R2_Score**
2. **Mean Squared Error**
3. **Mean Absolute Error**

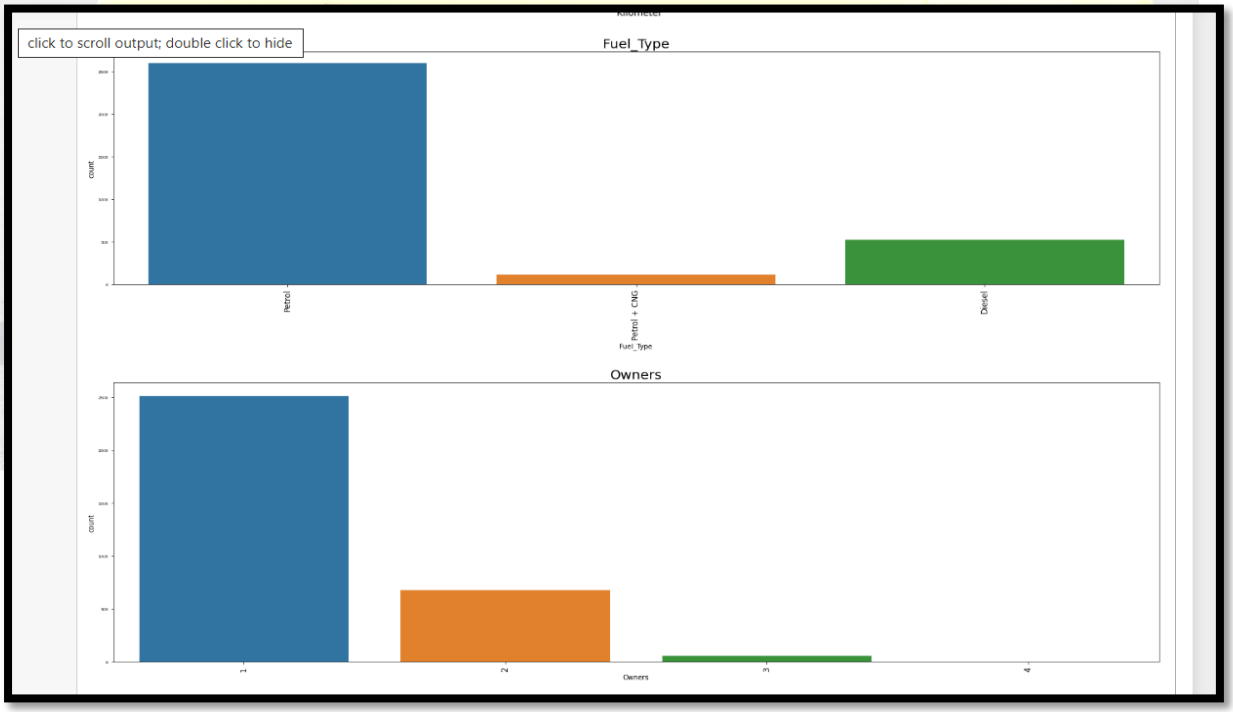
Visualizations:

1.Uni-Variate Analysis:





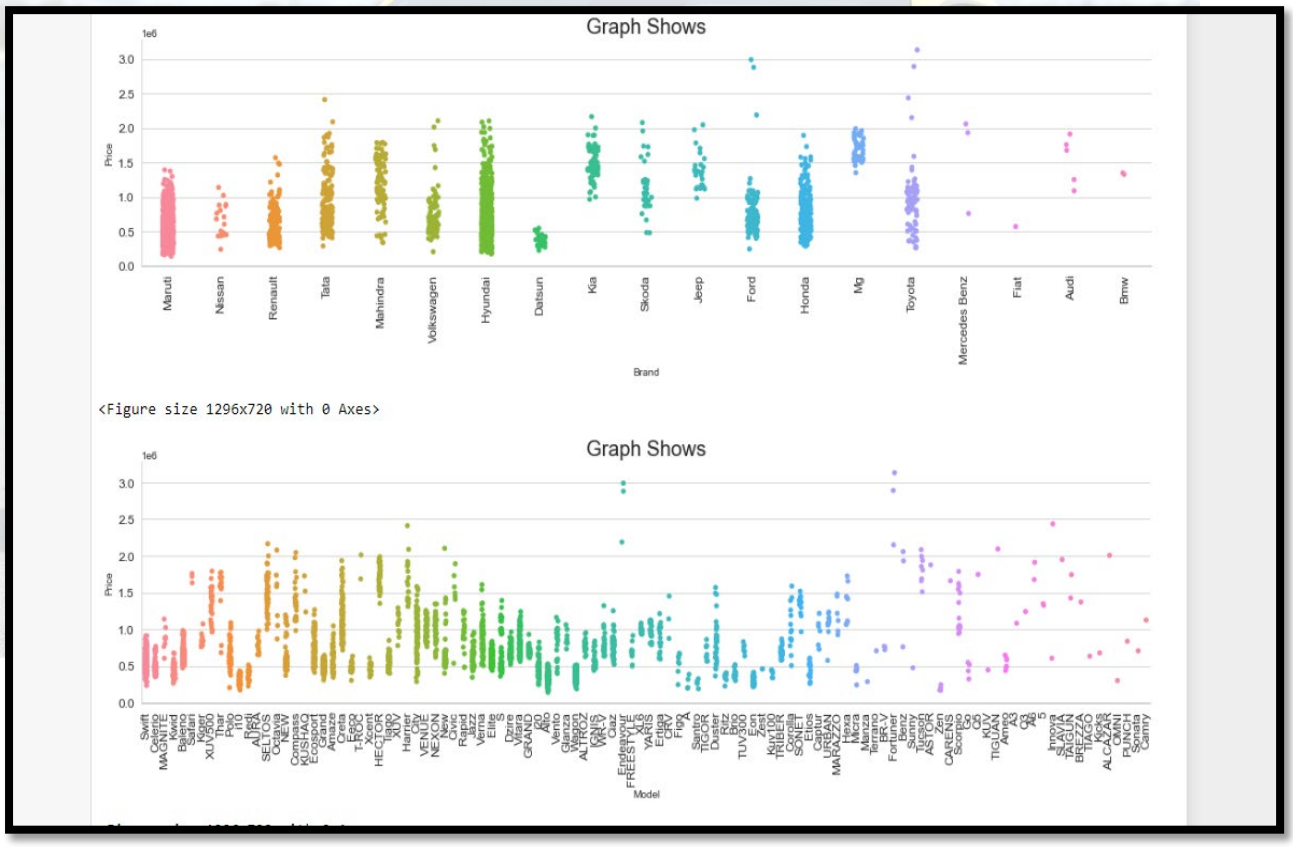

Make_Year
Count
2010
2012
2014
2016
2018
2020
2022
Make_Year
Kilometer
Count
0
100000
200000
300000
400000
Kilometer


click to scroll output; double click to hide
Fuel_Type
count
Petrol
Petrol + CNG
Diesel
Fuel_Type
Owners
count
1
2
3
4
Owners

Observations:

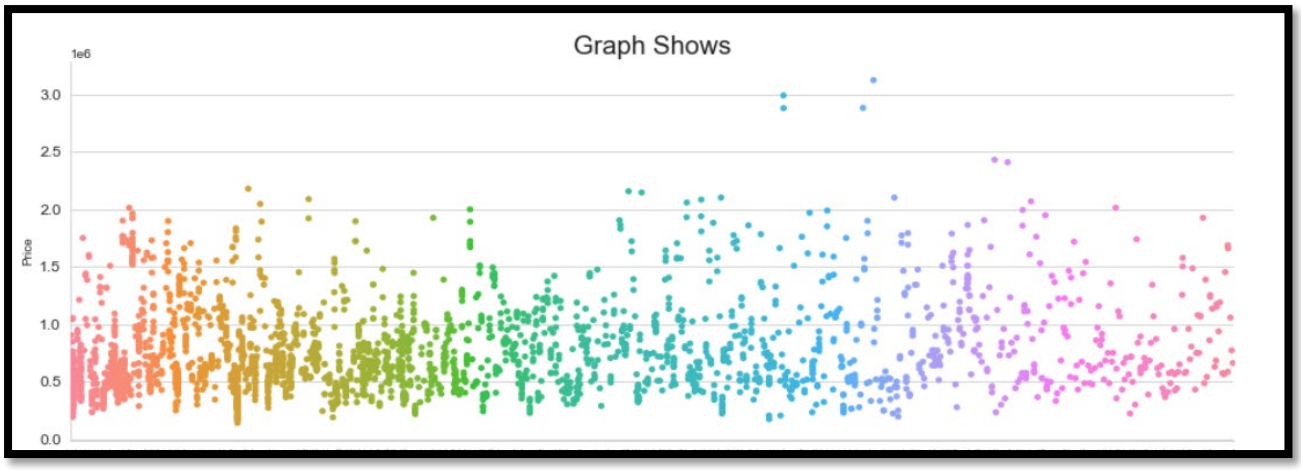
- Maruti Car (Brand) are for sale.
- Alto and Swift model's cars are in maximum for sale.
- VXI and LXI variants are maximum for sale.
- Cars that are maximum for sale are in Make year between 2018 to 2019
- Maximum car is for sale lies between kilometres 40,000 to 50,000 km.
- Petrol cars are maximum for sale, Petrol + CNG cars are very less for sale.
- Maximum Sales are 1st Owners, followed by 2nd Owners.
- Delhi and Karnataka cars are maximum for sale.
- Maximum sellers lie between 3 lakhs to 5 lakhs

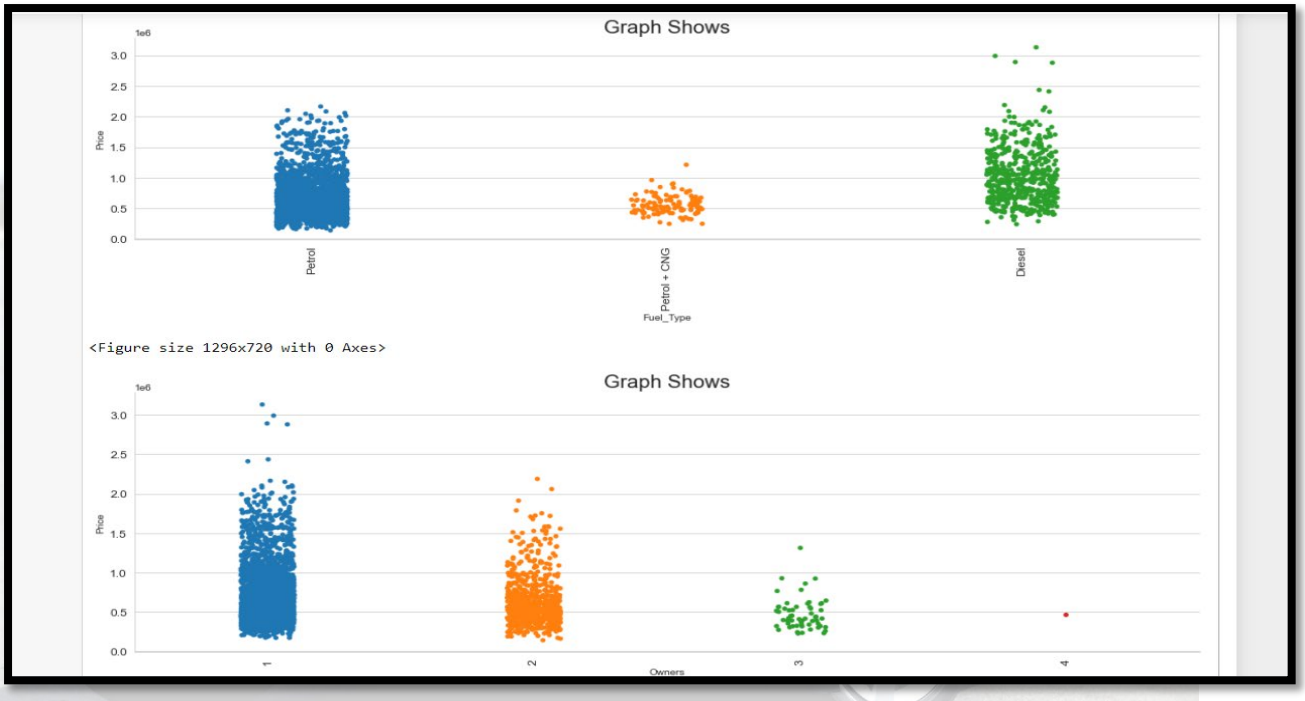
Bivariate Analysis



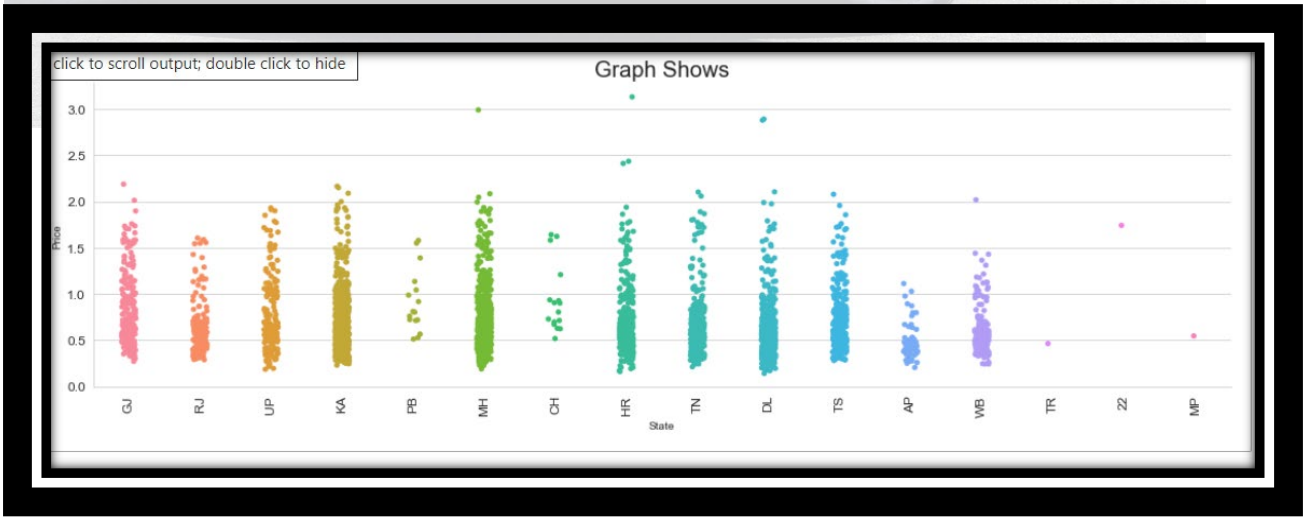
Observation:

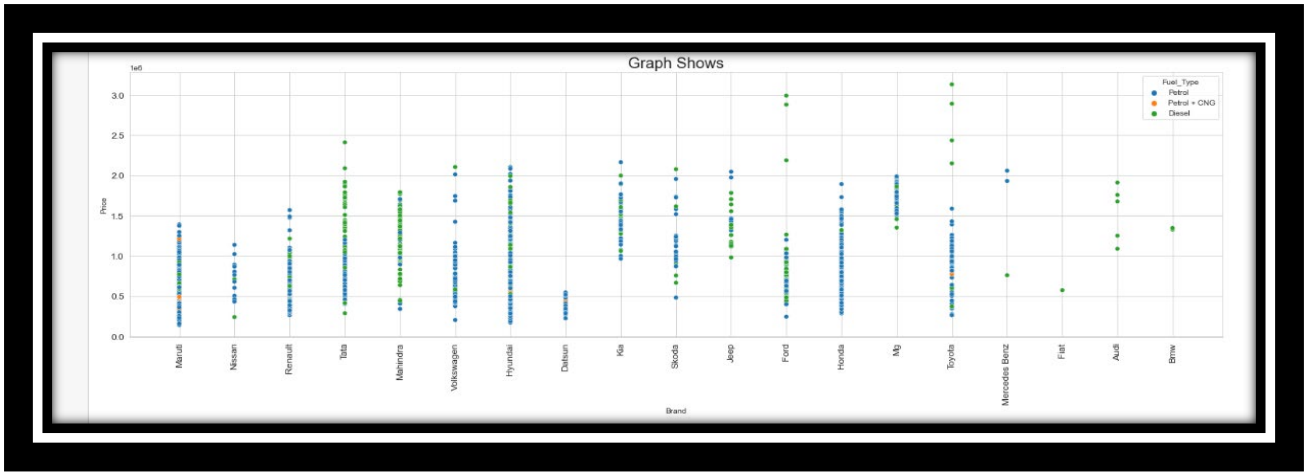
- Hyundai, Tata cars having high range and price as compare to other brands are high, Datsun likes under 0.0 to 0.5 range along with Fiat.

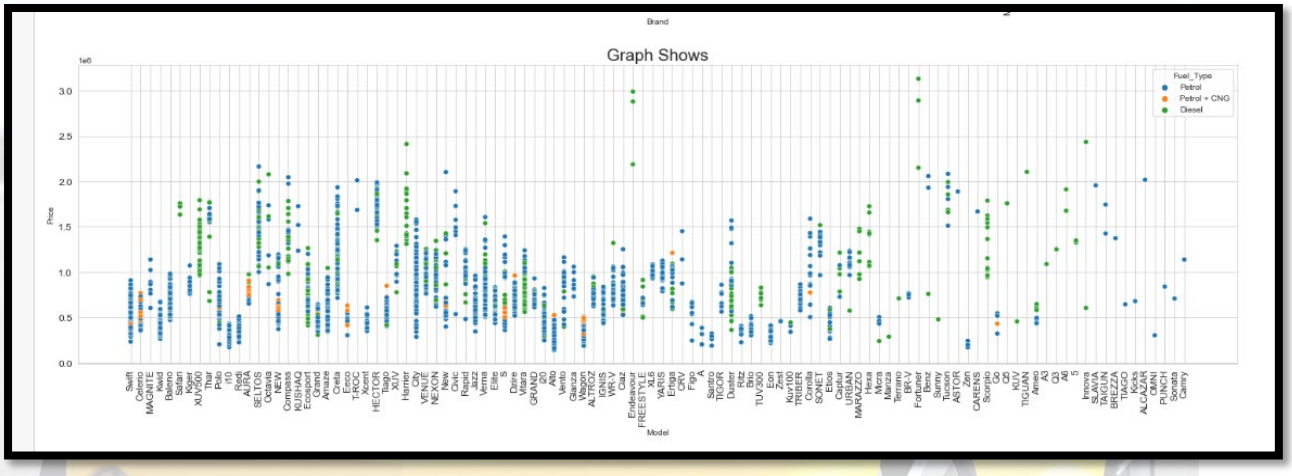


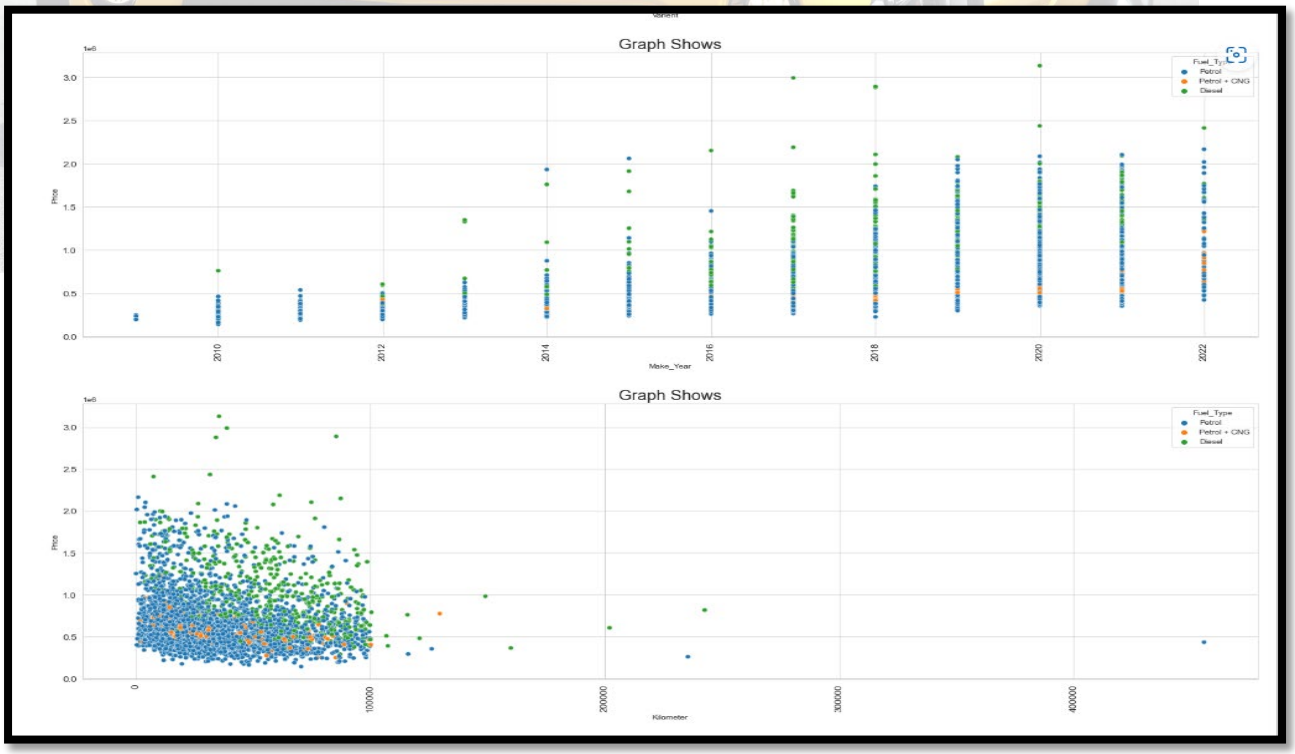


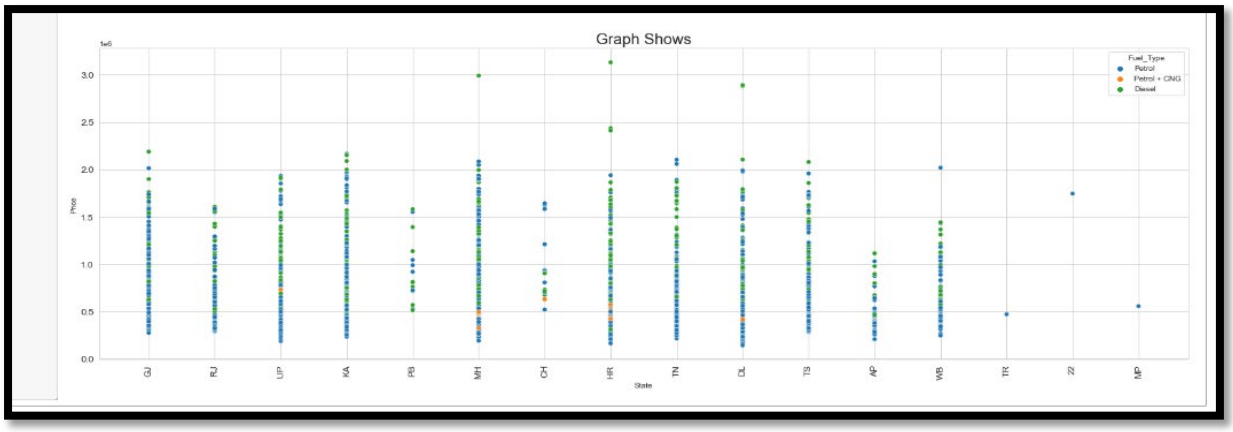

```
<Figure size 1296x720 with 0 Axes>
```


Graph Shows
Price
Brand
Fuel_Type
Petrol
Petrol + CNG
Diesel
Maruti
Nissan
Renault
Tata
Mahindra
Volkswagen
Hyundai
Datsun
Kia
Skoda
Jeep
Ford
Honda
Mg
Toyota
Mercedes Benz
Fiat
Audi
Bmw


Brand
Graph Shows
Price
Model
Fuel_Type
Petrol
Petrol + CNG
Diesel

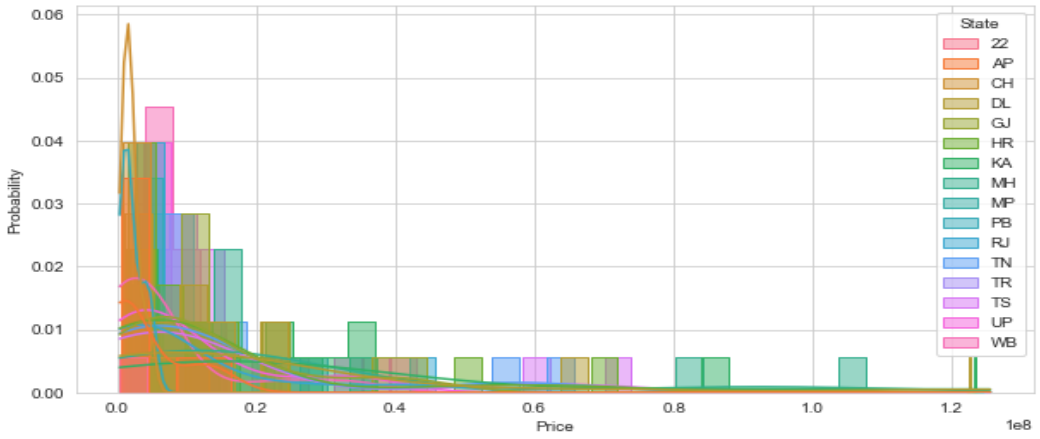

Graph Shows
Price
Make_Year
Fuel_Type
Petrol
Petrol + CNG
Diesel
Graph Shows
Price
Kilometer
Fuel_Type
Petrol
Petrol + CNG
Diesel



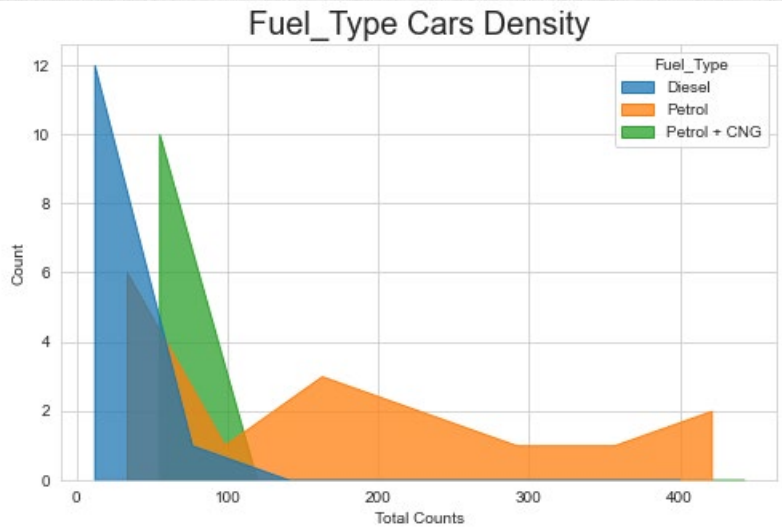
Observation price per state:

```
plt.figure(figsize=(10,5))
sns.histplot(x='Price',data=B_state,hue='State',element='step',multiple='dodge',stat='proba
```

```
<AxesSubplot:xlabel='Price', ylabel='Probability'>
```



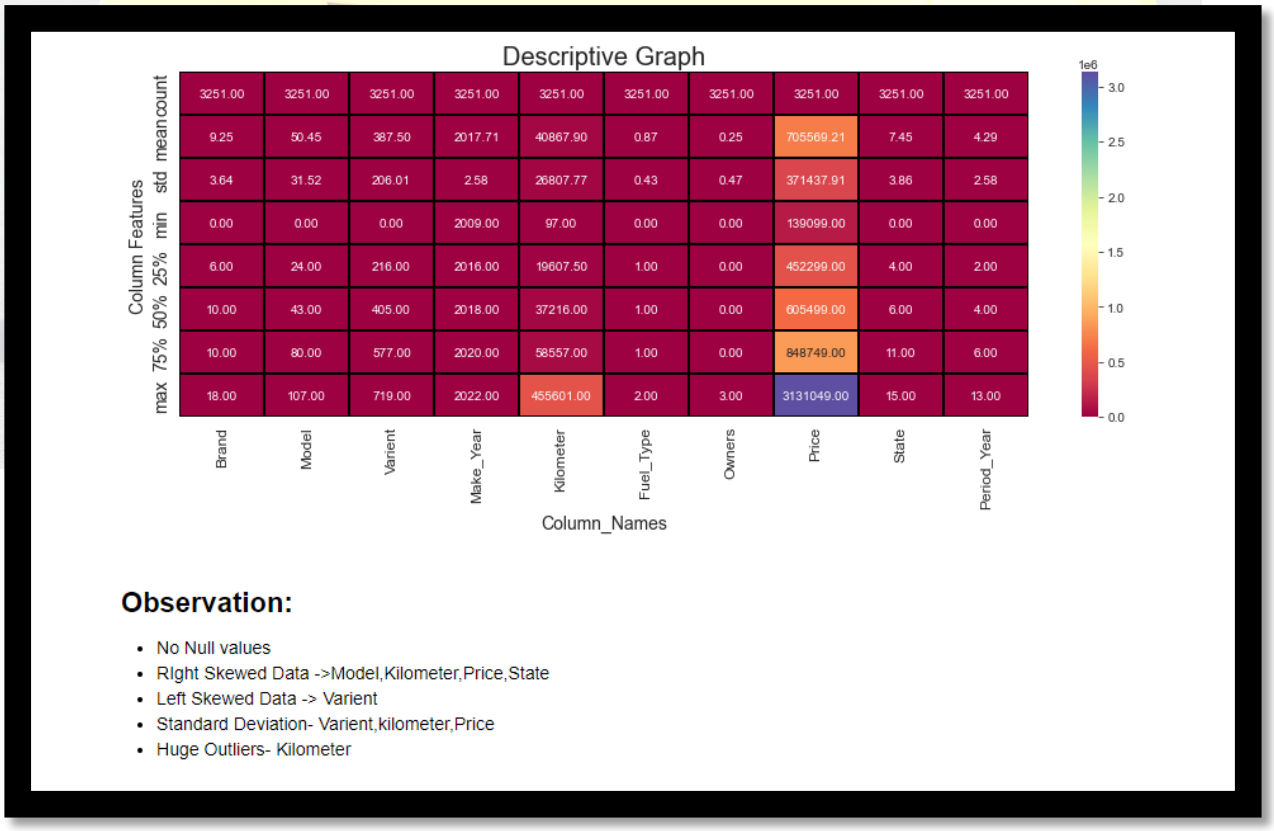
Density of Petrol cars are more for sale



Observation:

- Diesel car prices are high as compare to petrol or model with petrol and CNG.
- 1st Owner sells their cars in high price.
- Datsun Price are low; Toyota Ford shows maximum prices for sale.
- Maruti and Tata and Hyundai prices are low deviation.
- Endeavour, Fortuner prices are high compare to others, Harrier, compass and XUV 500 shows high price cars
- Maximum sale of petrol cars in 2020 followed 2022.
- Price of Petrol Cars are high if the Kilometre are less.
- Gujrat has maximum sellers of petrol cars.
- UP, KA, MH, HR, TN, DL, TS, AP, WB has maximum Diesel cars.
- HR, KA, GJ having prices for diesel cars
- Maximum price can be noticed for HR, MH, DL for diesel
- Endeavour and Fortuner have high selling values.
- High price value for first owner car.

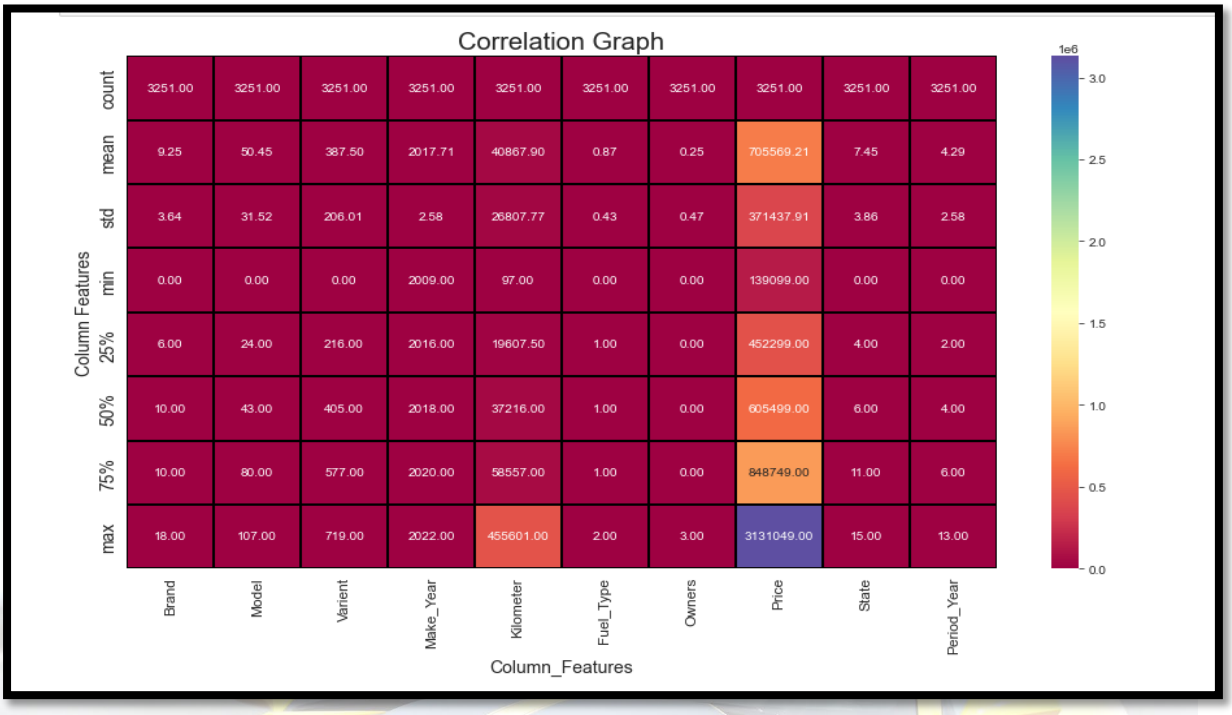
Descriptive Graph:

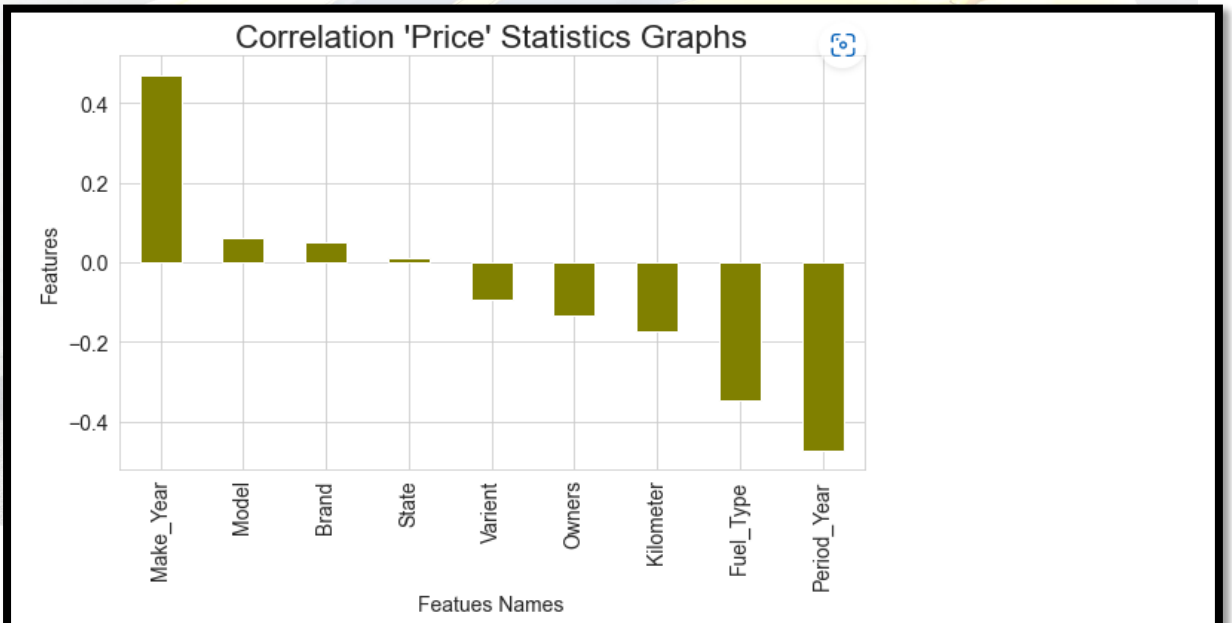


Observation:

- No Null values
- RIght Skewed Data ->Model,Kilometer,Price,State
- Left Skewed Data -> Varient
- Standard Deviation- Varient,kilometer,Price
- Huge Outliers- Kilometer

Correlation:

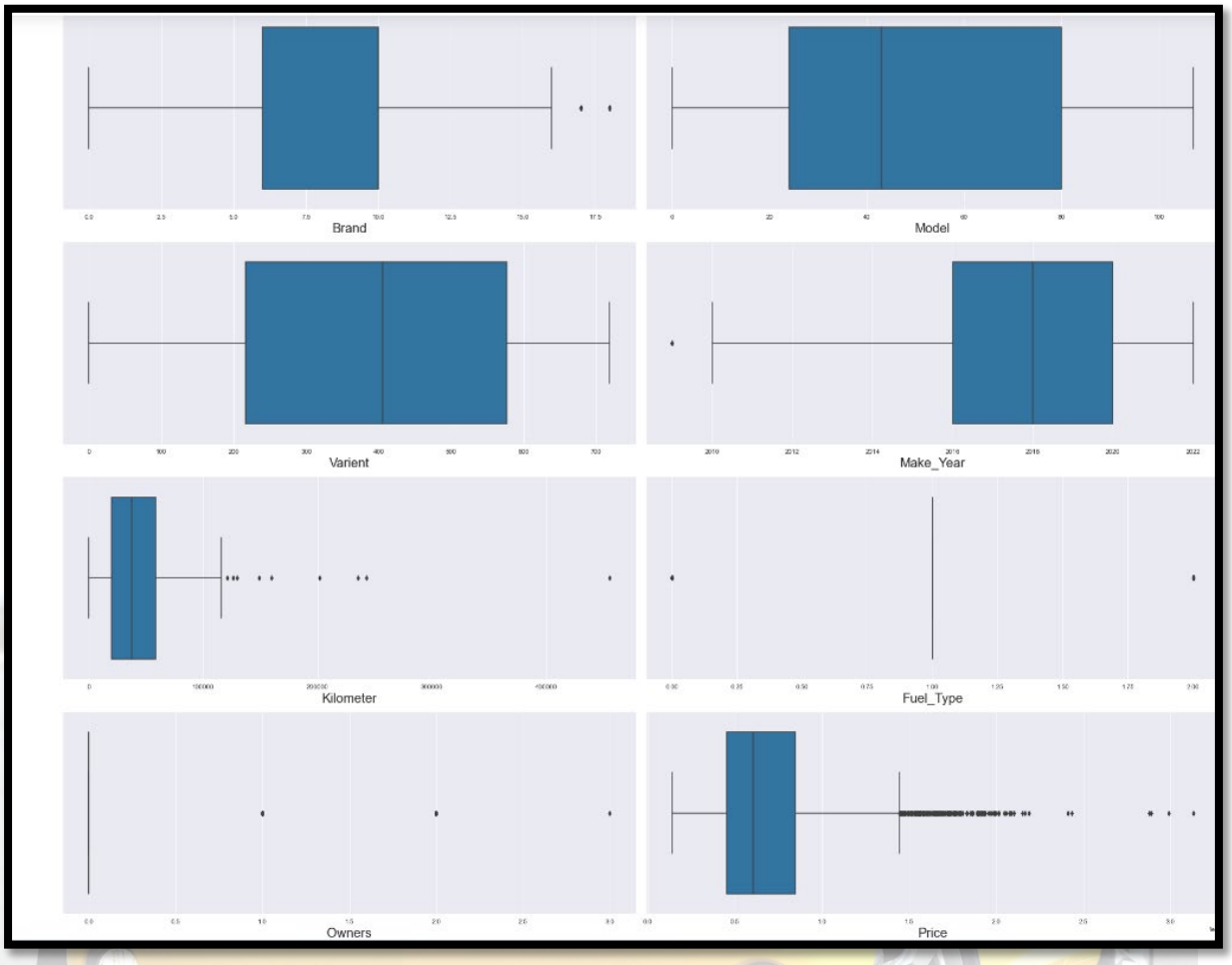


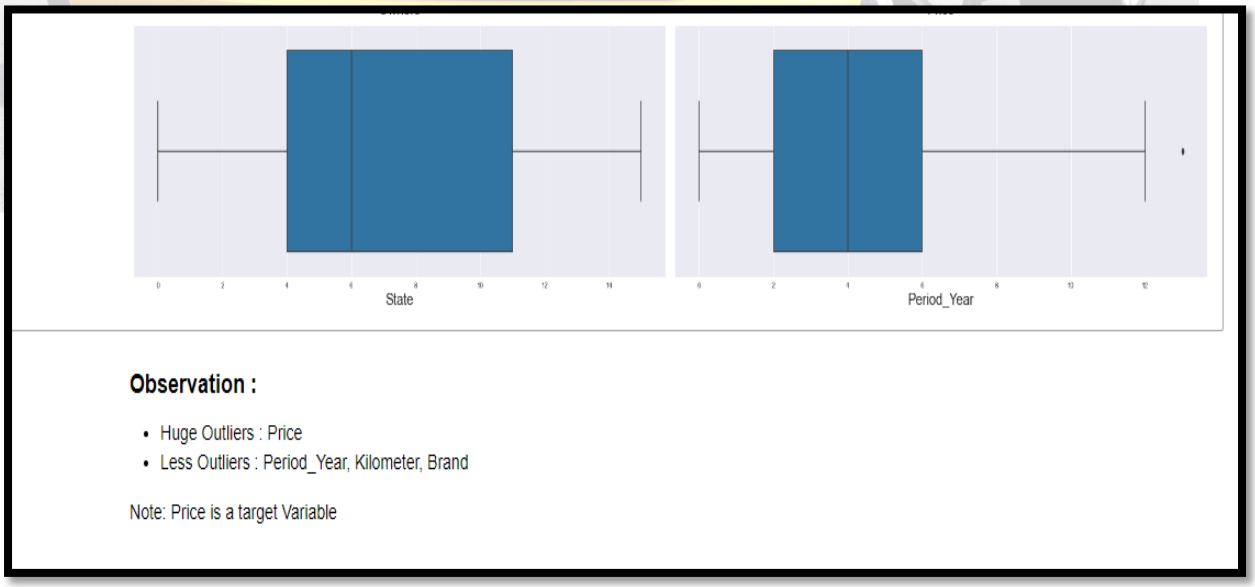


Observation Shows:

- Period_Year is 47 percentage Negatively Correlated with the Target Variable
- Fuel_Type is 35 percentage Negatively Correlated with the Target Variable
- Kilometer is 17 percentage Negatively Correlated with the Target Variable.
- Owners is 13 percentage Negatively Correlated with the Target Variable.
- Variable is 10 percentage Negatively Correlated with the Target Variable.
- State is 1 percentage Positively correlated with the Target Variable.
- Brand is 5 percentage Positively correlated with the Target Variable
- Model is 6 percentage Positively Correlated with the Target Variable
- Price is 100 percentage correlated to OWn.

Outliers:



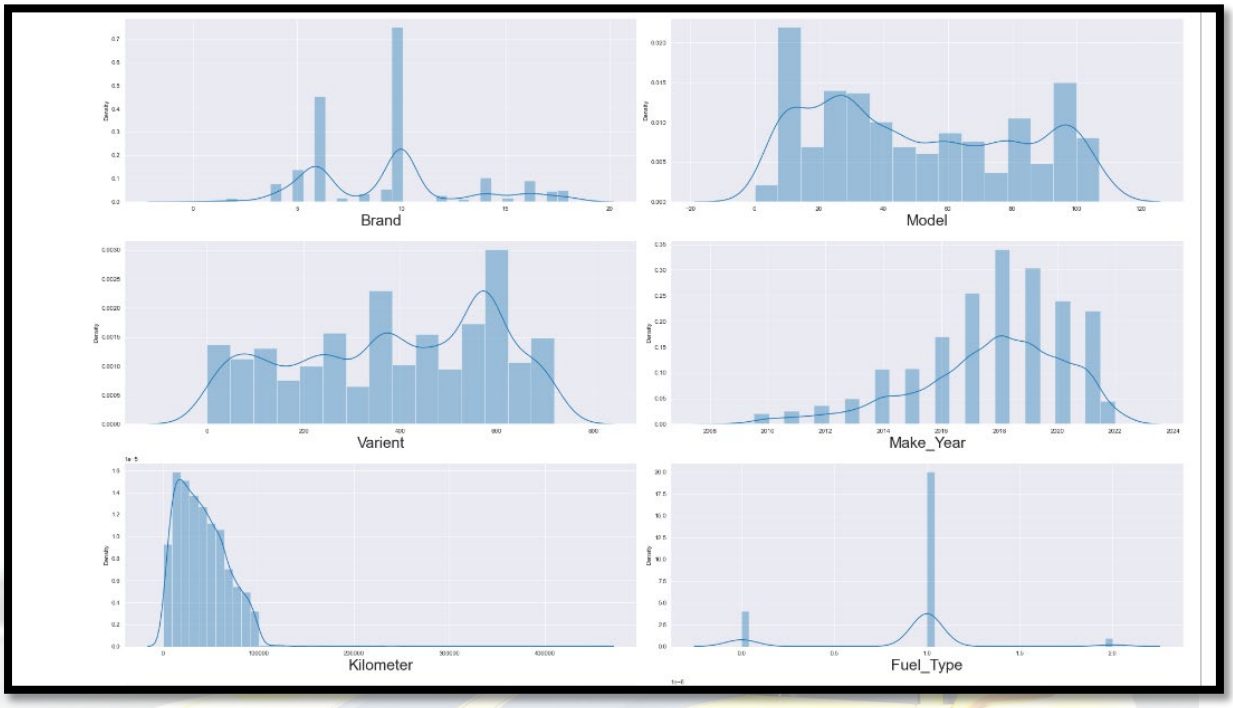


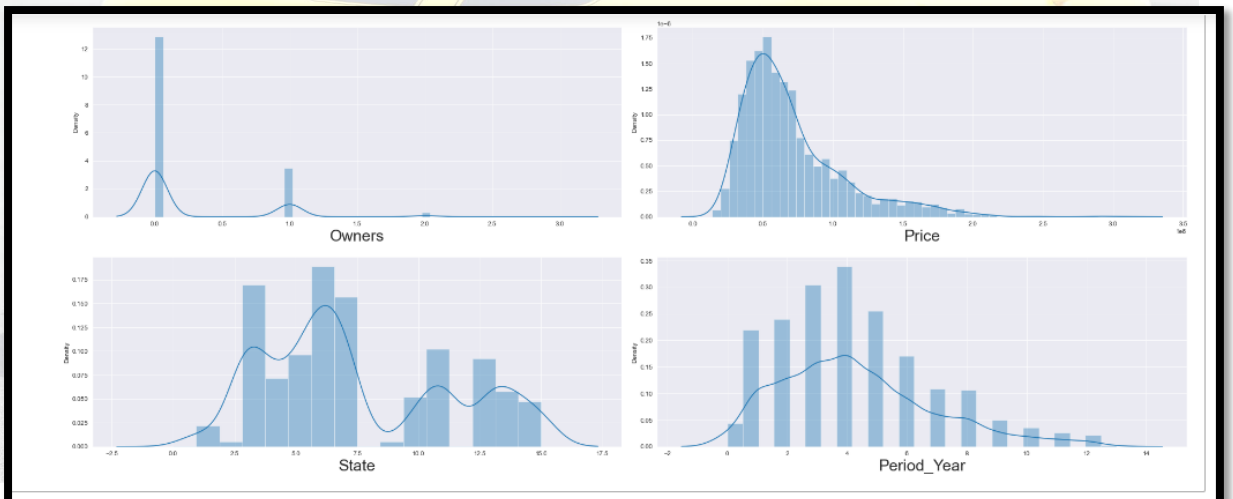
Observation :

- Huge Outliers : Price
- Less Outliers : Period_Year, Kilometer, Brand

Note: Price is a target Variable

Skewness in these columns





Graph Observation

- Brand graph shows its is not norlised as the graph lies beyond the Normal curve (Multi-Modal)
- Model graph shows is not normalised. as the building blocks are beyond the normal curve.
- Varient graph Shows that it is not normalised as the building blocks are beyond the normal curve.
- Kilometer graph Shows that it is not normalised as the building blocks are beyond the normal graph.
- Fuel_Type Shows Graph data has three modes.
- Owners Shows Graph data has 3 modes.
- Price graph Shows data not normalised.
- State and Period_Year shows data is not normalised.

➢ Interpretation of the Results

- This dataset was very special as it had a separate data type dataset for int columns.
- Firstly, the datasets were having no null values.
- The data set had 1948 duplicated rows. Checked and dropped.
- I found maximum continuous columns and less integer columns.
- Maximum cat Plot and count plot was used followed by hist plot, scatter plot to find the relationship with target variable.
- I notice a huge number of outliers and high skewness in the data so we have chosen proper methods to deal with the outliers and skewness. If we ignore this outlier and skewness, we may end up with a bad model which has less accuracy.
- Then scaling both train dataset has a good impact like it will help the model not to get biased.
- We have to use multiple models while building model using train dataset as to get the best model out of it.
- Extra Trees and Random Forest were the best among all the models. Result received with both model is approx. above 88 percent.
- However Extra Trees Shows accuracy and CV_Score both around 88%, Shows our model is performing extremely well....
- Finally selected Extra Tree Regressor and saved the model and finally printed the score and finally compared the data with the test data set.

CONCLUSION

➢ Key Findings and Conclusions of the Study

In this project report, I have found that Diesel cars have high price, First owners get maximum Price on compare to others. Rest all the observations have been updated in observation columns. Here we have used machine learning algorithms to predict the price of old cars prices for our client, we have mentioned the step-by-step procedure to analyse the dataset and finding the correlation Between the features. Thus, we can select the features which are not correlated to each other and are independent in nature.

These feature set were then given as an input to algorithms to predict output. Hence, we calculated the performance of each model using different performance metrics and compared them based on these metrics.

And saved the model as filename="cars.pkl"

➢ Learning Outcomes of the Study in respect of Data Science

I found that the dataset was quite interesting to handle as it contains all types of data in it. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analysed. New analytical techniques of machine learning can be used in car prediction research.

The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove missing value and to replace null value and zero values with their respective mean, median or mode.

This study is an exploratory attempt to use machine learning algorithms in finding the probability for each car price against each model. To conclude the application of Machine Learning in prediction is still at an early stage. I hope this study has moved a small step ahead in providing solution to the companies.

The changes I faced was when I was during scrapping the huge data from cars 24. other Issue was while using binning process taking the median values helped me to achieve a good accuracy. Finally, I had to run the test twice and thrice with different standardization process. I was actually thinking to get best out of those.

However, I finally achieved a good model out of this.

➢ Limitations of this work and Scope for Future Work

This model doesn't predict future probability. The future will be unpredictable at all times due to this, the risk in investment in an import factor. This can predict for a time period and needs to be updated on a basis. Machine can predict what would be the price as the data fended but it can't predict the intensity of the market and markers who are into the business.

The best way to be future ready to get the modal updated once or twice as per the market standard.

Thank-you