FLIP ROBO

PROJECT REPORT ON:
“CAR PRICE PREDICTION”

SUBMITTED BY

Safiya Firdose Khan

ACKNOWLEDGMENT

I would like to express my sincere thanks of gratitude to my mentors from Data Trained academy and Flip Robo Technologies Bangalore for letting me work on this project. Their suggestions and directions have helped me in the completion of this project successfully. All the required information & the dataset are provided by Flip Robo Technologies.

Finally, I would like to thank my family and friends who have helped me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

Contents:

1.INTRODUCTION

1.1 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. This project contains two phases: Data Collection Phase We need to scrape approx. 5000 used cars data. In this section we need to scrape the data of used cars from websites. We need webscraping for this. we have to fetch data for different locations. The number of columns for data doesn't have limit. Generally, these columns are Brand, model, variant, manufacturing year, driven kilometers, fuel, number of owners, location and at last target variable Price of the car. This data is to give you a hint about important variables in used car model. Model Building Phase After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model.

1.2 Conceptual Background of the Domain Problem

The factors affecting the Automotive Industry are:

1) Political factors

Political factors may also include goods and services which the government wants to provide or be provided and those that the government does not want to be provided.

Almost all of the regulations come from consumers increasing concerns for the environment and the concern for safer automotives. For instance trade barriers and incentives to the public to buy new cars.

2) Economic factors

Economic factors include interest rates, disposable income, unemployment rates, retail price index (inflation), gross domestic product(GDP), and exchange rates. These factors have major impacts on how businesses operate and make decisions.

The automotive industry has a huge impact on every country's economy. According to various studies this industry is the major user of computer chips, textiles, aluminum, copper, steel, iron, lead, plastics, vinyl, and rubber. These industries include anything from the aluminums to lead to vinyl.

3) Social factors

Social factors include the cultural aspects and include awareness, population growth rate, age distribution, career attitudes and emphasis on safety. Trends in social factors affect the demand for a company's products and how that company operates. Furthermore, companies may change various management strategies to adapt to these social trends.

Anyone who drives a nice vehicle is thought to be wealthy. No one wants to be seen driving an unattractive piece of junk because of what other people will think of him or her. Consumers also just feel better when they are driving a nice or new car, if makes them feel better about themselves.

4) Technological factors

Technological factors include ecological and environmental aspects, such as R&D activity, automation, technology incentives and the rate of technological change. Furthermore, technological shifts can affect costs, quality, and lead to innovation.

The internet has affected just about every industry in the world and has also had a huge impact on the automotive industry. Business-to-business marketplaces have given the industry many opportunities because of the internet, such as more efficiency and lower cost.

1.3 Review of Literature

As per the requirement of our client, by using Selenium, I have scrapped data from the website **cars24**, and used Pandas library to save the data in excel file and csv file.

Based on the data collected I have tried analyzing based on what factors the used car price is decided, and I have developed a model that will predict the price of the used cars.

1.4 Motivation for the Problem Undertaken

I have to build a model to predict the price of used cars with the available independent variables. This model will then be used by the client to understand how exactly the prices vary with the variables.

2.Analytical Problem Framing

2.1 Mathematical/ Analytical Modeling of the Problem

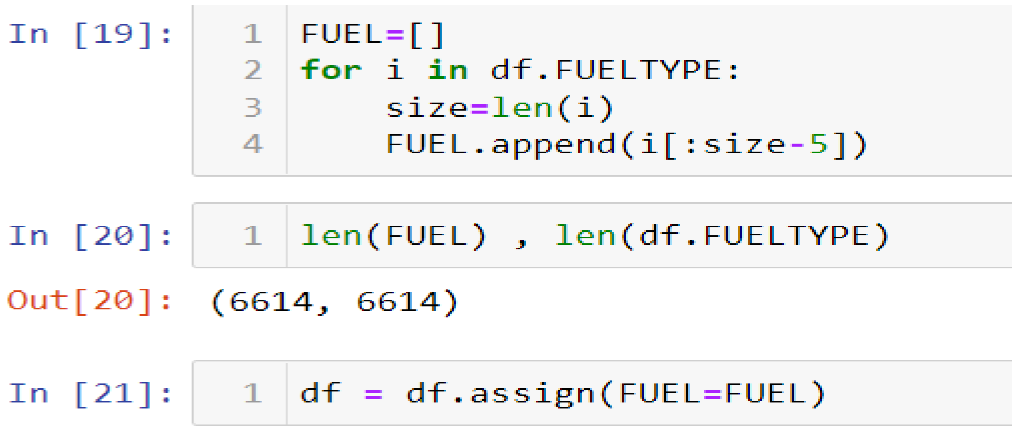
The dataset is collected by scrapping the data of used cars from the website **cars24.** I have built the model for predicting the price of used cars. By looking into the target column, I came to know that the entries of PRICE column were continuous and this was a Regression problem so I have to use all the regression algorithms while building the model. While checking the null values in the dataset, I found out that the column VARIANT contains 332 null values, So, I have replaced the null values with mode of that column. Next, I have removed the unnecessary details present in the column FUELTYPE, and renamed it as column FUEL. I have also removed the unnecessary details present in the columns DRIVEN_KM, PRICE and NOOF_OWNERS. To get better insight on the features I have used plotting like distribution plot, count plot, bar plot, line plot and scatter plot. With this plotting, I was able to understand the relation between the features in better manner. Also, I found outliers and skewness in the dataset, so I removed outliers using zscore method and I removed skewness using yeo-johnson method and log transformation technique. I have used all the regression models while building model, and then tuned the best model and saved it. At last, I have predicted the PRICE using the saved model.

2.2 Data Sources and their formats

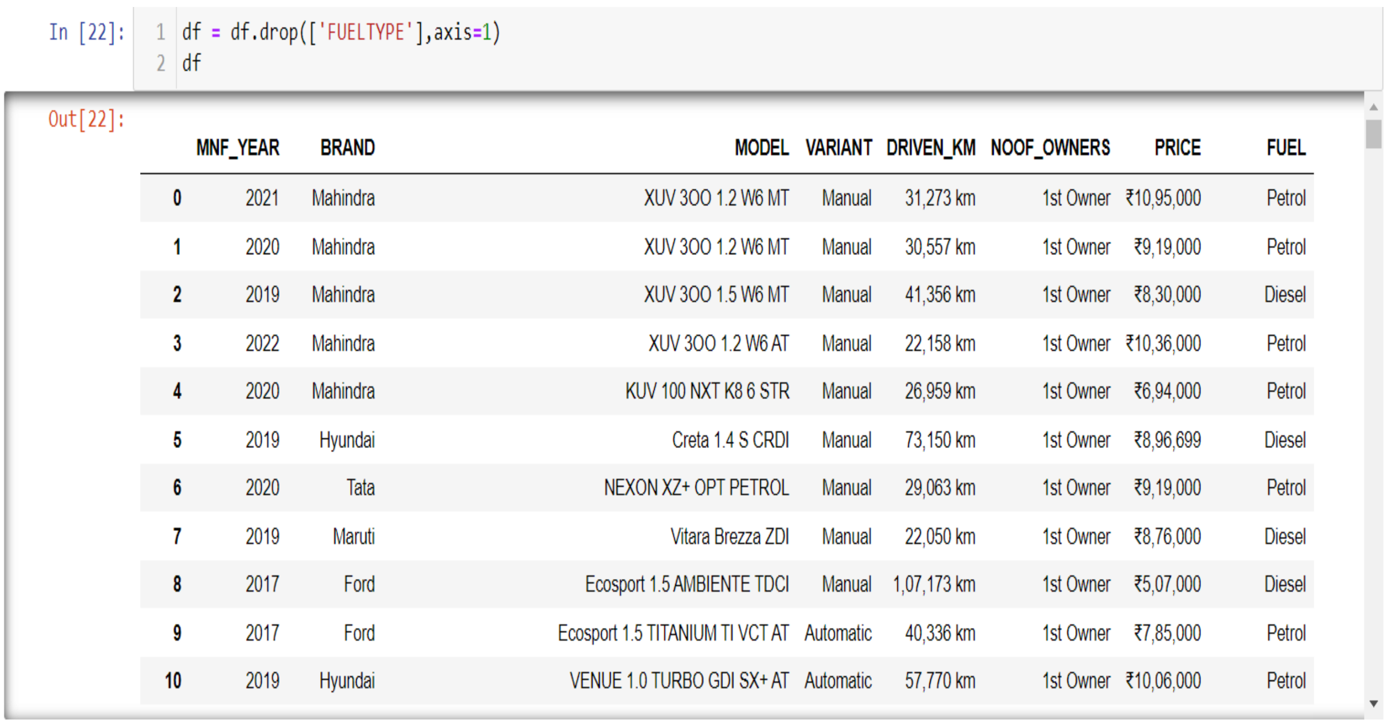
I have scrapped data from the website **cars24**, and used Pandas library to save th e data in excel file and csv file. The dataset consists of 6614 rows and 8 columns in cluding target. In this particular dataset, I have object and integer types of data.

2.3 Data Preprocessing Done

- ✓ As a first step I have imported required libraries and I have imported the dataset which is in csv format.
- ✓ Then, I performed all the statistical analysis, like checking shape, nunique, value counts, info, etc.
- ✓ While checking the null values in the dataset, I found out that the column VARIANT contains 332 null values, So, I have replace the null values with mode of that column.
- ✓ Next, I have removed the unnecessary details present in the column FUELTYPE, and renamed it as column FUEL.



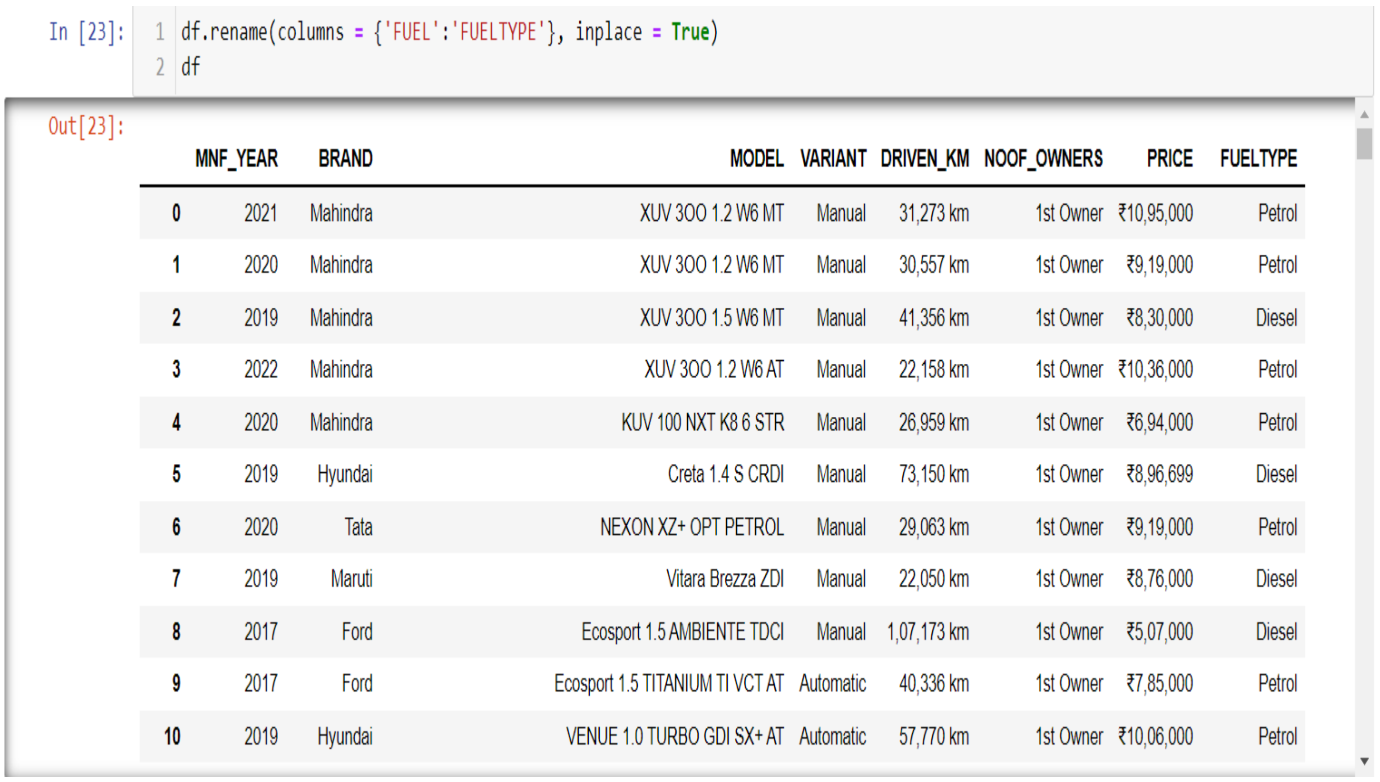
```
In [19]: FUEL=[]
         for i in df.FUELTYPE:
             size=len(i)
             FUEL.append(i[:size-5])

In [20]: len(FUEL) , len(df.FUELTYPE)

Out[20]: (6614, 6614)

In [21]: df = df.assign(FUEL=FUEL)
```



In [22]:
```python
df = df.drop(['FUELTYPE'],axis=1)
df
```

Out[22]:

	MNF_YEAR	BRAND	MODEL	VARIANT	DRIVEN_KM	NOOF_OWNERS	PRICE	FUEL
0	2021	Mahindra	XUV 3OO 1.2 W6 MT	Manual	31,273 km	1st Owner	₹10,95,000	Petrol
1	2020	Mahindra	XUV 3OO 1.2 W6 MT	Manual	30,557 km	1st Owner	₹9,19,000	Petrol
2	2019	Mahindra	XUV 3OO 1.5 W6 MT	Manual	41,356 km	1st Owner	₹8,30,000	Diesel
3	2022	Mahindra	XUV 3OO 1.2 W6 AT	Manual	22,158 km	1st Owner	₹10,36,000	Petrol
4	2020	Mahindra	KUV 100 NXT K8 6 STR	Manual	26,959 km	1st Owner	₹6,94,000	Petrol
5	2019	Hyundai	Creta 1.4 S CRDI	Manual	73,150 km	1st Owner	₹8,96,699	Diesel
6	2020	Tata	NEXON XZ+ OPT PETROL	Manual	29,063 km	1st Owner	₹9,19,000	Petrol
7	2019	Maruti	Vitara Brezza ZDI	Manual	22,050 km	1st Owner	₹8,76,000	Diesel
8	2017	Ford	Ecosport 1.5 AMBIENTE TDCI	Manual	1,07,173 km	1st Owner	₹5,07,000	Diesel
9	2017	Ford	Ecosport 1.5 TITANIUM TI VCT AT	Automatic	40,336 km	1st Owner	₹7,85,000	Petrol
10	2019	Hyundai	VENUE 1.0 TURBO GDI SX+ AT	Automatic	57,770 km	1st Owner	₹10,06,000	Petrol

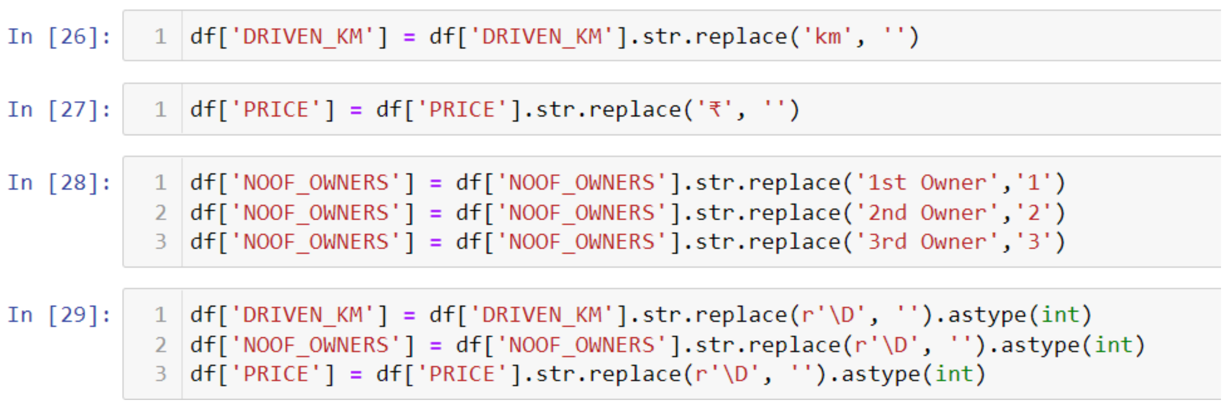


In [23]:
```python
df.rename(columns = {'FUEL':'FUELTYPE'}, inplace = True)
df
```

Out[23]:

	MNF_YEAR	BRAND	MODEL	VARIANT	DRIVEN_KM	NOOF_OWNERS	PRICE	FUELTYPE
0	2021	Mahindra	XUV 3OO 1.2 W6 MT	Manual	31,273 km	1st Owner	₹10,95,000	Petrol
1	2020	Mahindra	XUV 3OO 1.2 W6 MT	Manual	30,557 km	1st Owner	₹9,19,000	Petrol
2	2019	Mahindra	XUV 3OO 1.5 W6 MT	Manual	41,356 km	1st Owner	₹8,30,000	Diesel
3	2022	Mahindra	XUV 3OO 1.2 W6 AT	Manual	22,158 km	1st Owner	₹10,36,000	Petrol
4	2020	Mahindra	KUV 100 NXT K8 6 STR	Manual	26,959 km	1st Owner	₹6,94,000	Petrol
5	2019	Hyundai	Creta 1.4 S CRDI	Manual	73,150 km	1st Owner	₹8,96,699	Diesel
6	2020	Tata	NEXON XZ+ OPT PETROL	Manual	29,063 km	1st Owner	₹9,19,000	Petrol
7	2019	Maruti	Vitara Brezza ZDI	Manual	22,050 km	1st Owner	₹8,76,000	Diesel
8	2017	Ford	Ecosport 1.5 AMBIENTE TDCI	Manual	1,07,173 km	1st Owner	₹5,07,000	Diesel
9	2017	Ford	Ecosport 1.5 TITANIUM TI VCT AT	Automatic	40,336 km	1st Owner	₹7,85,000	Petrol
10	2019	Hyundai	VENUE 1.0 TURBO GDI SX+ AT	Automatic	57,770 km	1st Owner	₹10,06,000	Petrol

- ✓ I have also removed the unnecessary details present in the columns DRIVEN_KM, PRICE and NOOF_OWNERS.

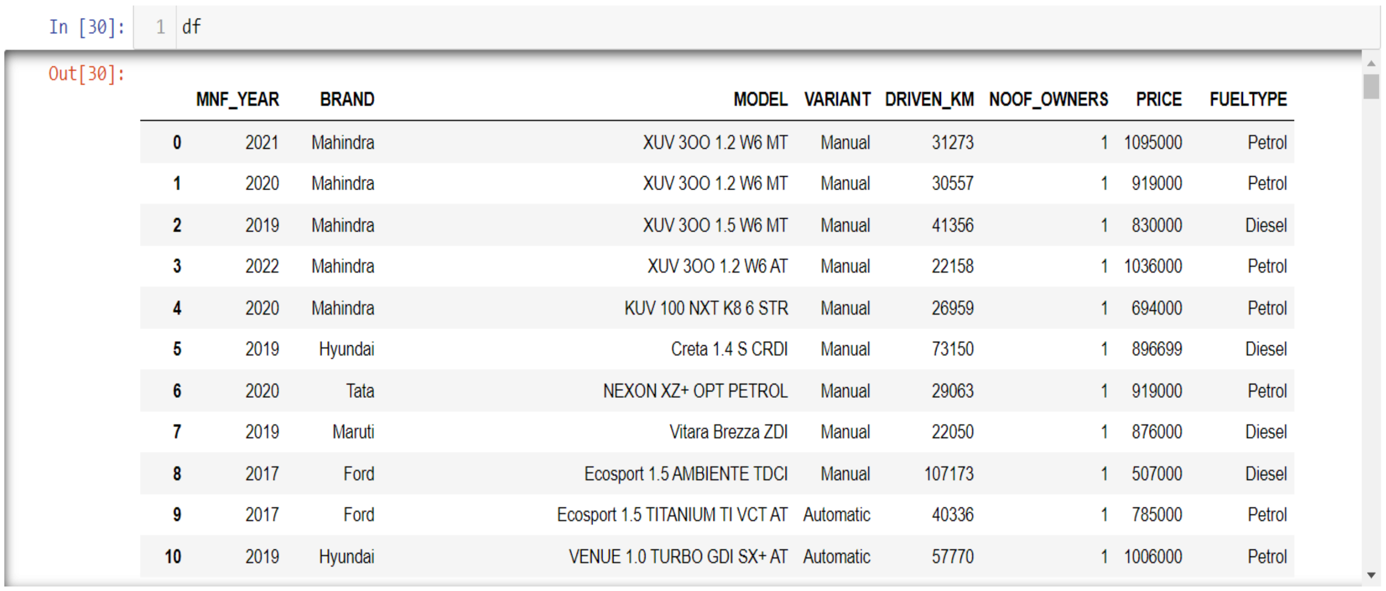


```
In [26]: df['DRIVEN_KM'] = df['DRIVEN_KM'].str.replace('km', '')
```

```
In [27]: df['PRICE'] = df['PRICE'].str.replace('₹', '')
```

```
In [28]: df['NOOF_OWNERS'] = df['NOOF_OWNERS'].str.replace('1st Owner','1')
         df['NOOF_OWNERS'] = df['NOOF_OWNERS'].str.replace('2nd Owner','2')
         df['NOOF_OWNERS'] = df['NOOF_OWNERS'].str.replace('3rd Owner','3')
```

```
In [29]: df['DRIVEN_KM'] = df['DRIVEN_KM'].str.replace(r'\D', '').astype(int)
         df['NOOF_OWNERS'] = df['NOOF_OWNERS'].str.replace(r'\D', '').astype(int)
         df['PRICE'] = df['PRICE'].str.replace(r'\D', '').astype(int)
```



```
In [30]: df
```

Out[30]:

	MNF_YEAR	BRAND	MODEL	VARIANT	DRIVEN_KM	NOOF_OWNERS	PRICE	FUELTYPE
0	2021	Mahindra	XUV 3OO 1.2 W6 MT	Manual	31273	1	1095000	Petrol
1	2020	Mahindra	XUV 3OO 1.2 W6 MT	Manual	30557	1	919000	Petrol
2	2019	Mahindra	XUV 3OO 1.5 W6 MT	Manual	41356	1	830000	Diesel
3	2022	Mahindra	XUV 3OO 1.2 W6 AT	Manual	22158	1	1036000	Petrol
4	2020	Mahindra	KUV 100 NXT K8 6 STR	Manual	26959	1	694000	Petrol
5	2019	Hyundai	Creta 1.4 S CRDI	Manual	73150	1	896699	Diesel
6	2020	Tata	NEXON XZ+ OPT PETROL	Manual	29063	1	919000	Petrol
7	2019	Maruti	Vitara Brezza ZDI	Manual	22050	1	876000	Diesel
8	2017	Ford	Ecosport 1.5 AMBIENTE TDCI	Manual	107173	1	507000	Diesel
9	2017	Ford	Ecosport 1.5 TITANIUM TI VCT AT	Automatic	40336	1	785000	Petrol
10	2019	Hyundai	VENUE 1.0 TURBO GDI SX+ AT	Automatic	57770	1	1006000	Petrol

2.4 Data Inputs- Logic- Output Relationships

- ✓ I have used bar plot, line plot and scatter plot for each pair of features that shows the relation with the PRICE.

2.5 Hardware and Software Requirements and Tools Used

While taking up the project we should be familiar with the Hardware and software required for the successful completion of the project. Here we need the following hardware and software.

Hardware required: -

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software/s required: -

1.Anaconda

Libraries required :-

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')
```

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

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

- ✓ **import pandas as pd**: **pandas** is 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.

- ✓ **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.

- ✓ **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.

- ✓ **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.
- ✓ from sklearn.preprocessing import OrdinalEncoder
- ✓ from sklearn.preprocessing import StandardScaler
- ✓ from statsmodels.stats.outliers_influence import variance_inflation_factor
- ✓ from sklearn.ensemble import RandomForestRegressor
- ✓ from xgboost import XGBRegressor
- ✓ from sklearn.ensemble import ExtraTreesRegressor
- ✓ from sklearn.ensemble import GradientBoostingRegressor
- ✓ from sklearn.tree import DecisionTreeRegressor
- ✓ from sklearn.ensemble import BaggingRegressor
- ✓ from sklearn.metrics import classification_report
- ✓ from sklearn.model_selection import cross_val_score

With these libraries, we can go ahead with our analytical skills.

3.Data Analysis and Visualization

3.1 Identification of possible problem-solving approaches (methods)

I have replaced the null values present in the column VARIANT, with mode of that column. To remove outliers, I have used zscore method, and to remove skewness I have used yeo-johnson method and log transformation technique. To encode the categorical columns, I have use Label Encoding. Use of Pearson's correlation coefficient to check the correlation between dependent and independent features. Also I have used standardization. Finally, I have built the model using all the regression algorithms, and finally saved the best model.

3.2 Testing of Identified Approaches (Algorithms)

Since PRICE is my target variable and it is continuous in nature, so this particular problem is regression problem. I have used all the regression algorithms to build my model. By looking into the difference of r2 score and cross validation score, I found XGBRegressor as the best model with least difference. Also to get the best model, we have to run through multiple models and to avoid the confusion of overfitting we have go through cross validation. Below are the list of regression algorithms I have used in my project.

- RandomForestRegressor
- XGBRegressor
- ExtraTreesRegressor
- GradientBoostingRegressor
- DecisionTreeRegressor
- BaggingRegressor

3.3 Key Metrics for success in solving problem under consideration

I have used the following metrics for evaluation:

I have used mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.

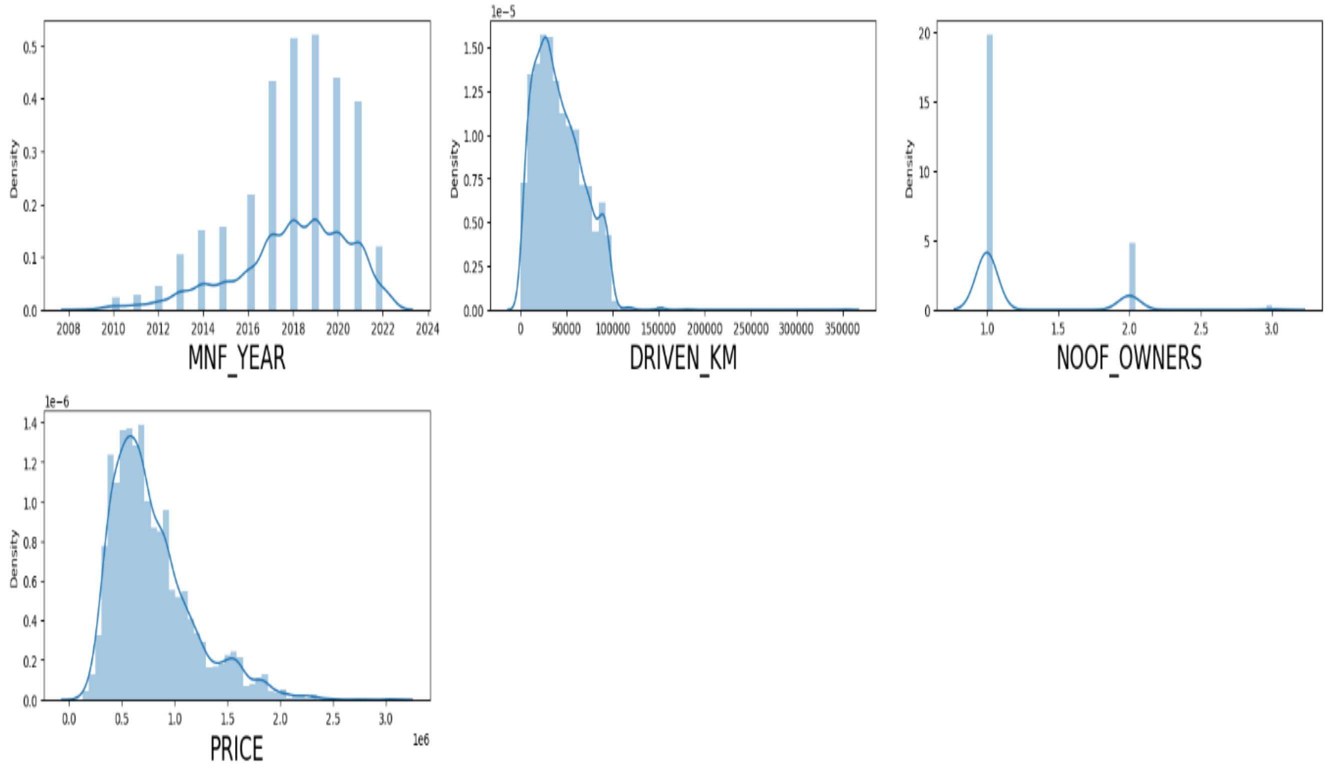
I have used root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions.

I have used r2 score which tells us how accurate our model is.

3.4 Visualizations

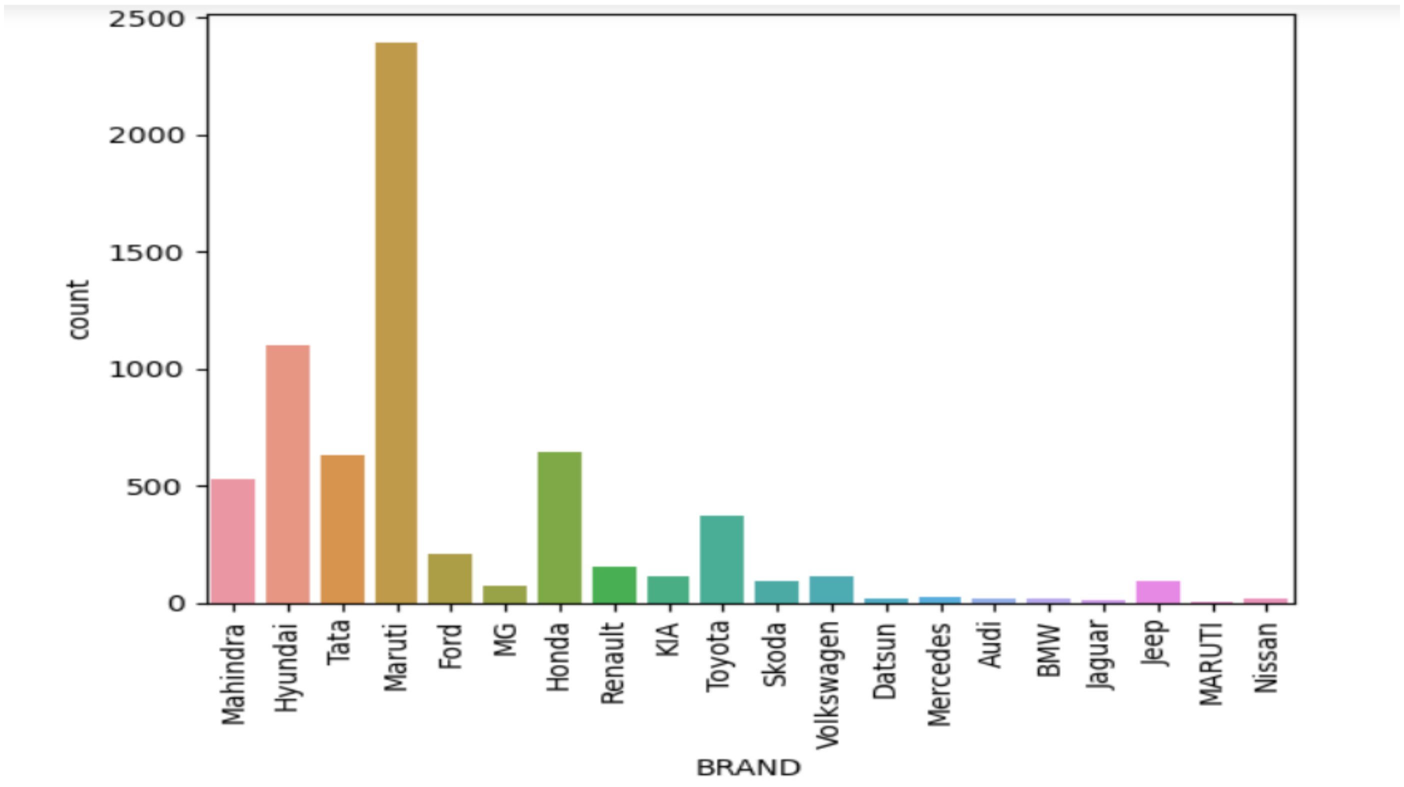
1. Univariate Analysis:

Univariate Analysis for numerical columns:

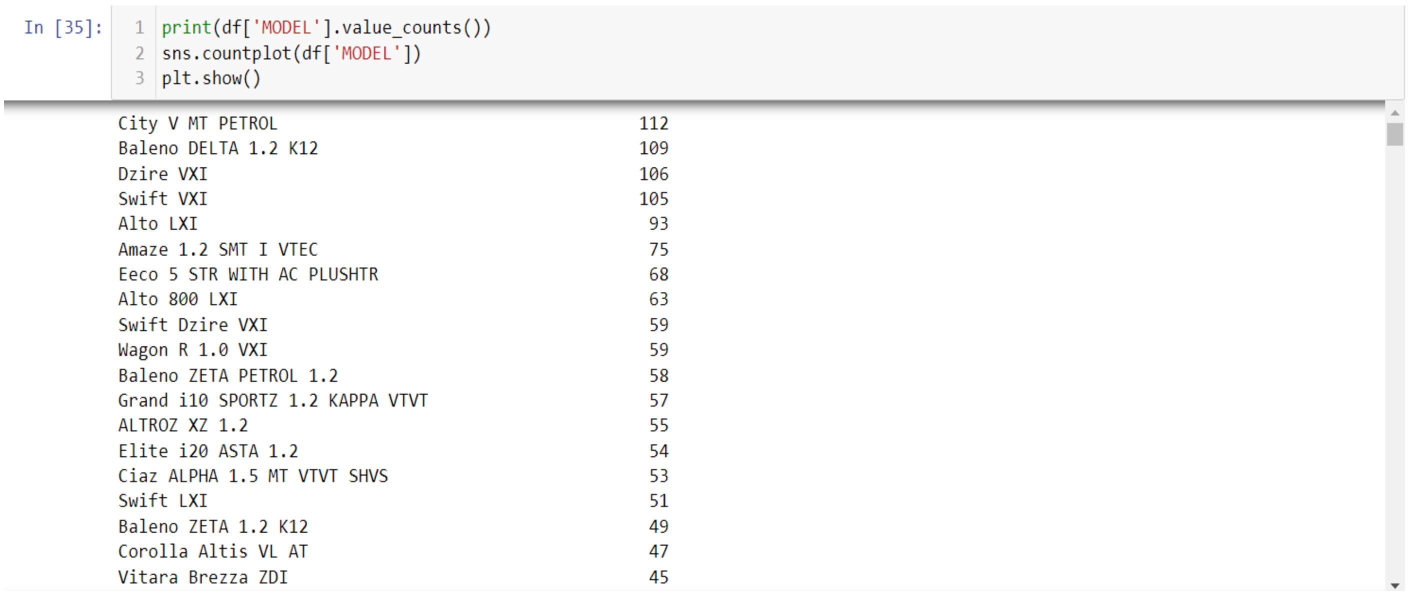


As we can see, skewness is present in the columns ‘MNF_YEAR’, ‘NOOF_OWNERS’ and ‘PRICE’. The column ‘DRIVEN_KM’ appears to be normal.

Univariate Analysis for categorical columns:

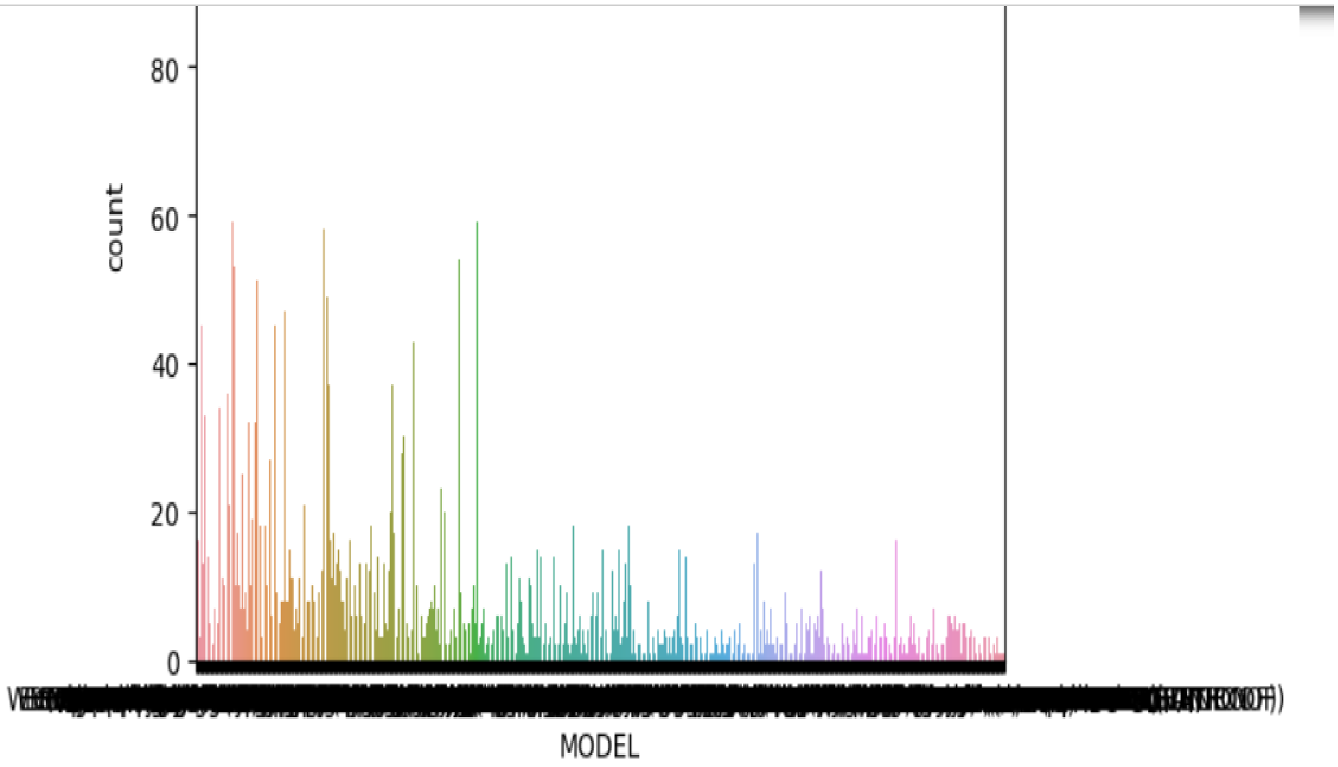


As we can see, the brand Maruti is largely available for sale, followed by Hyundai.

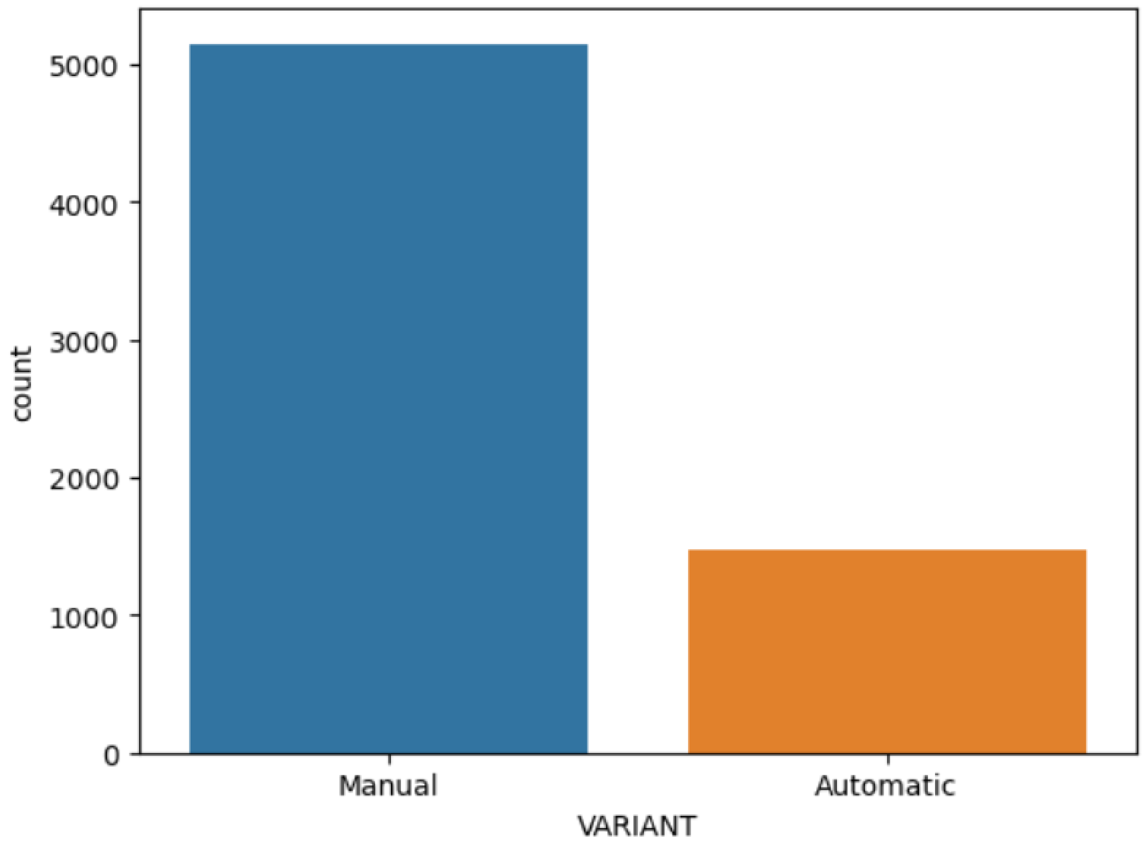


```
In [35]:
print(df['MODEL'].value_counts())
sns.countplot(df['MODEL'])
plt.show()
```

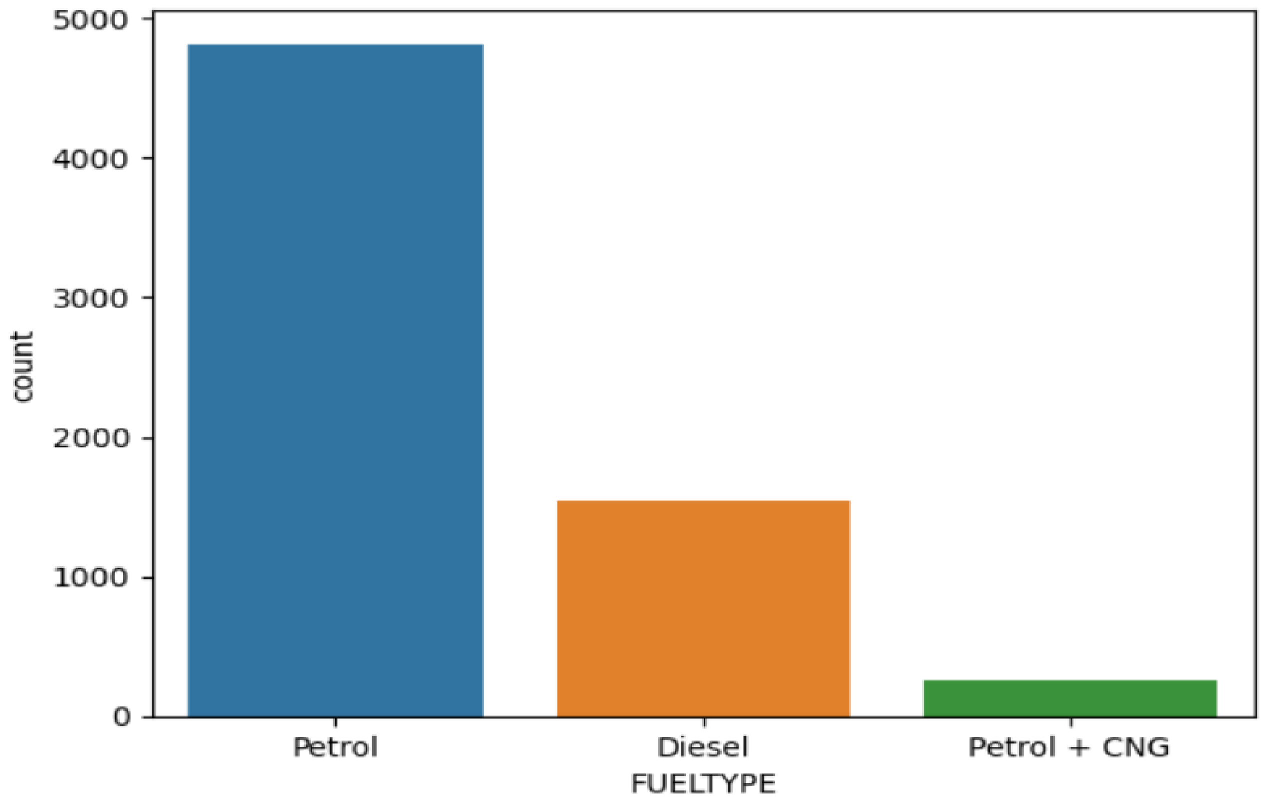
```
City V MT PETROL                        112
Baleno DELTA 1.2 K12                    109
Dzire VXI                               106
Swift VXI                               105
Alto LXI                                 93
Amaze 1.2 SMT I VTEC                     75
Eeco 5 STR WITH AC PLUSHTR               68
Alto 800 LXI                             63
Swift Dzire VXI                          59
Wagon R 1.0 VXI                          59
Baleno ZETA PETROL 1.2                   58
Grand i10 SPORTZ 1.2 KAPPA VTVT          57
ALTROZ XZ 1.2                            55
Elite i20 ASTA 1.2                       54
Ciaz ALPHA 1.5 MT VTVT SHVS              53
Swift LXI                                51
Baleno ZETA 1.2 K12                      49
Corolla Altis VL AT                      47
Vitara Brezza ZDI                        45
```



As we can see, the models largely available for sale are “City V MT PETROL”, followed by “Baleno DELTA 1.2 K12” and “Dzire VXI”.

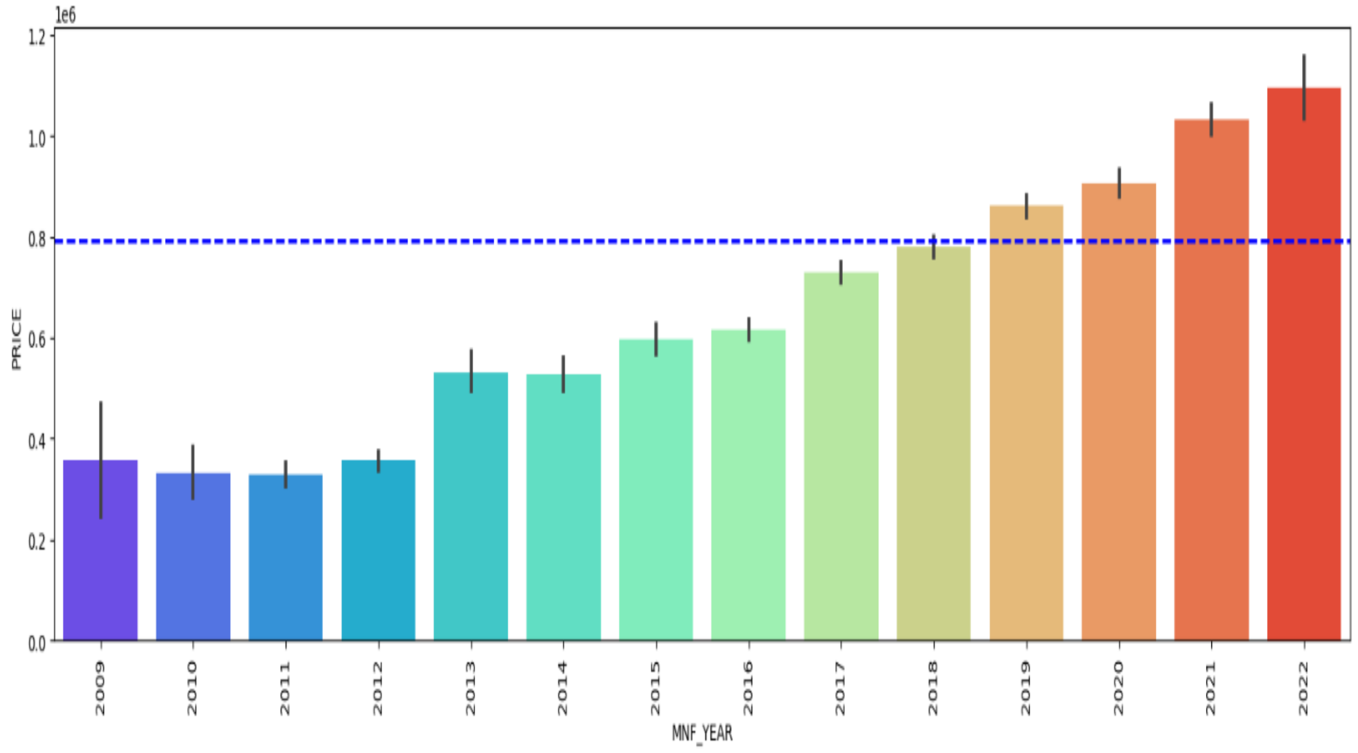


As we can see, the Manual Variants are largely available for sale, as compared to Automatic variants.

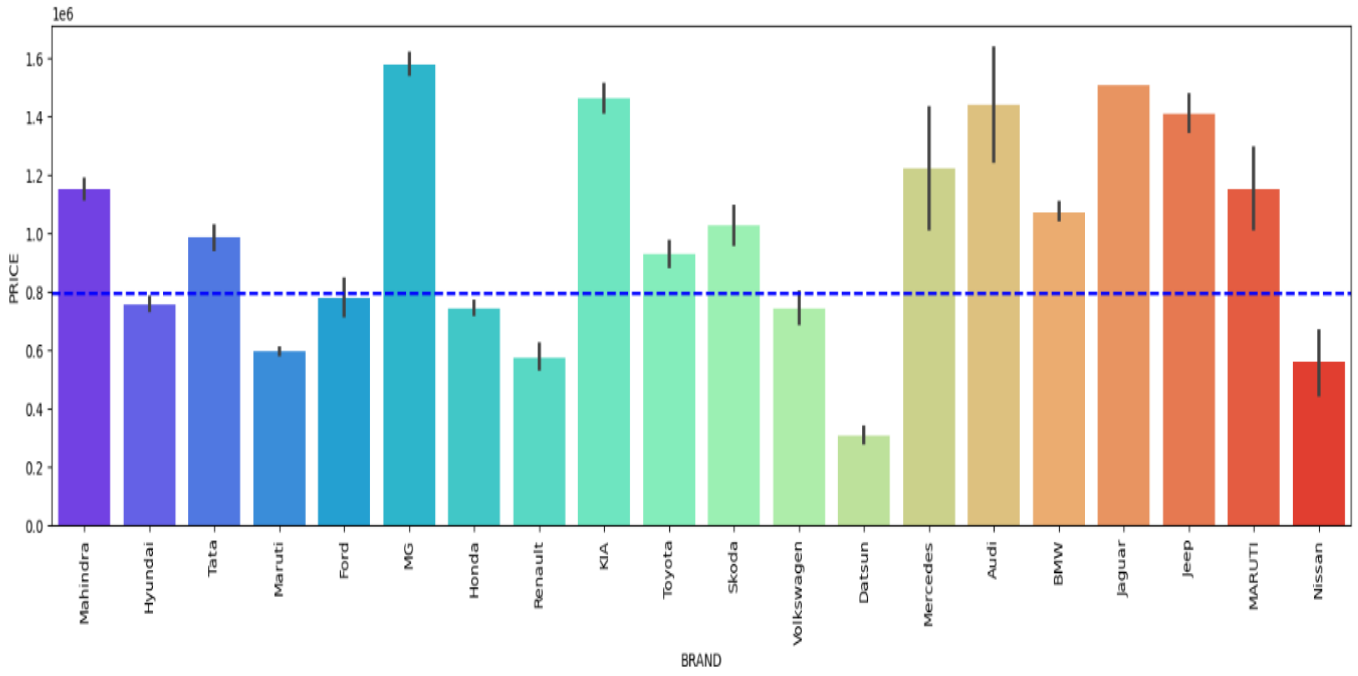


As we can see, the cars with fuel type Petrol are largely available for sale, followed by cars with fuel type Diesel and Petrol+CNG.

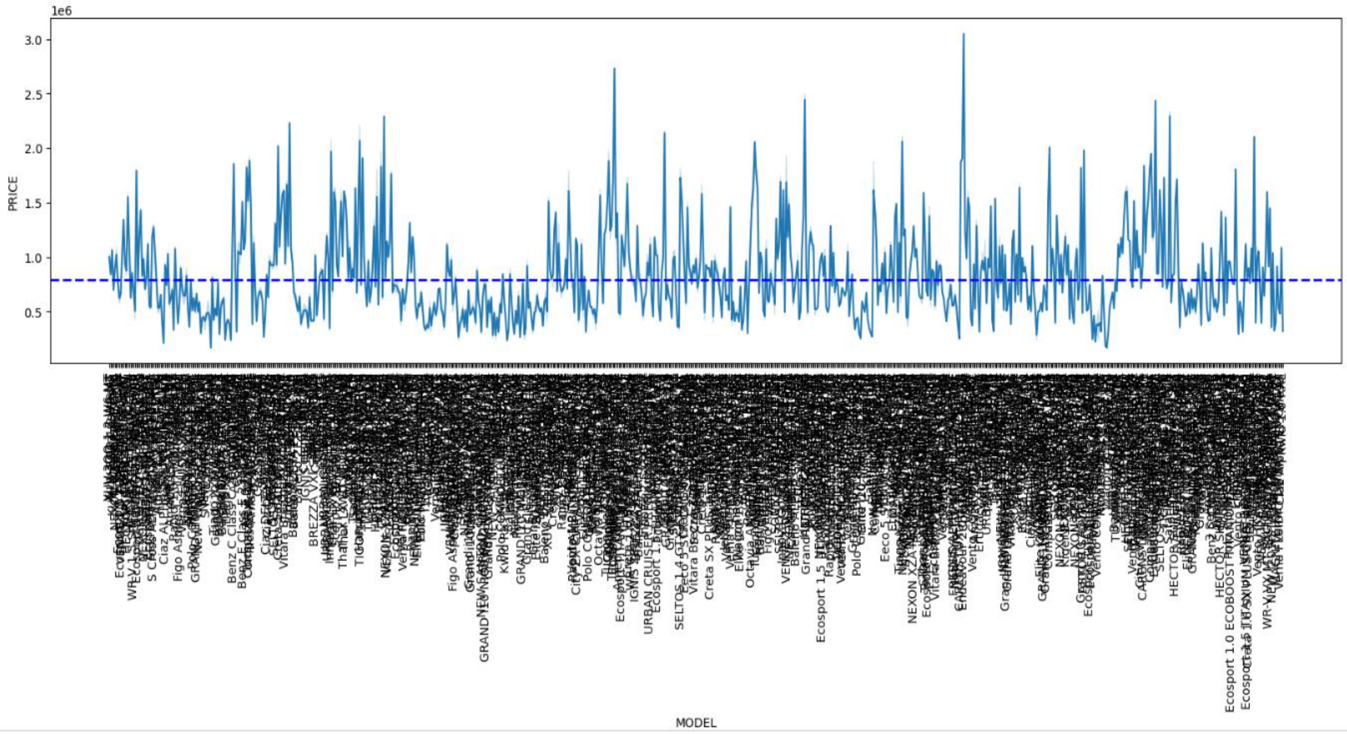
2. Bivariate Analysis:



As we can see, the cars manufactured in the year 2022 have a higher price, followed by cars manufactured in the year 2021 and 2020.

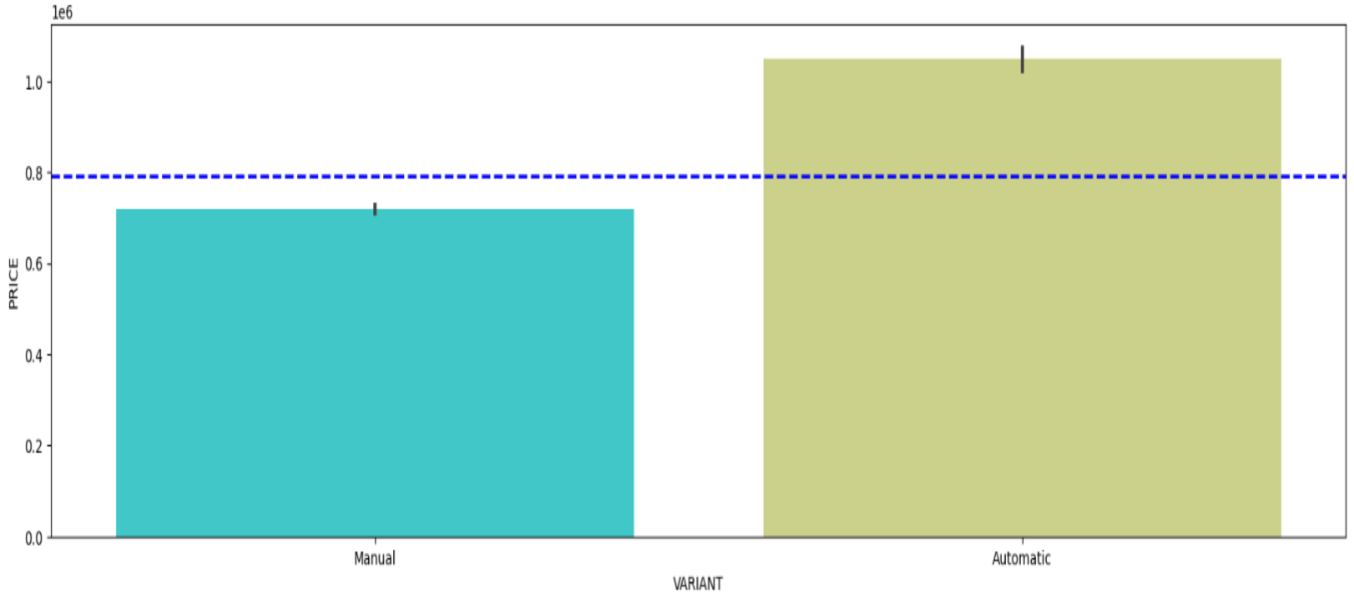


As we can see, the brand “MG” has the highest price, followed by “Jaguar” and “Audi”.

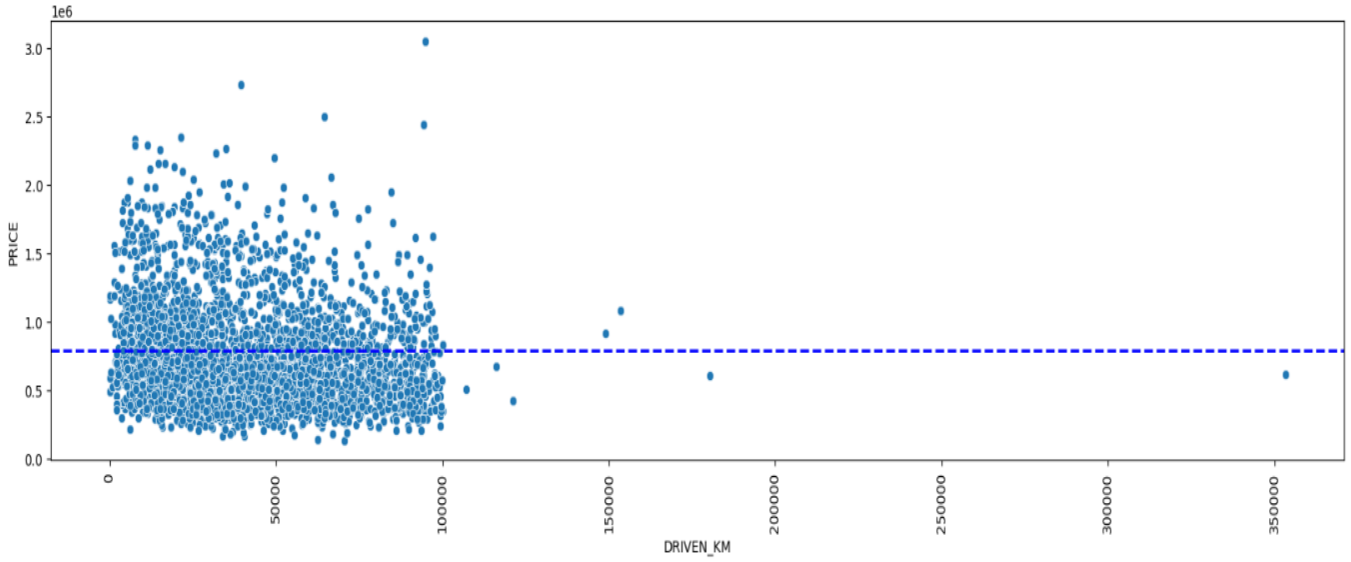


```
In [41]:   df['PRICE'].nlargest(5)

Out[41]: 3049    3048000
         3108    3048000
         3261    3048000
         1702    2731000
         1730    2731000
         Name: PRICE, dtype: int32
```

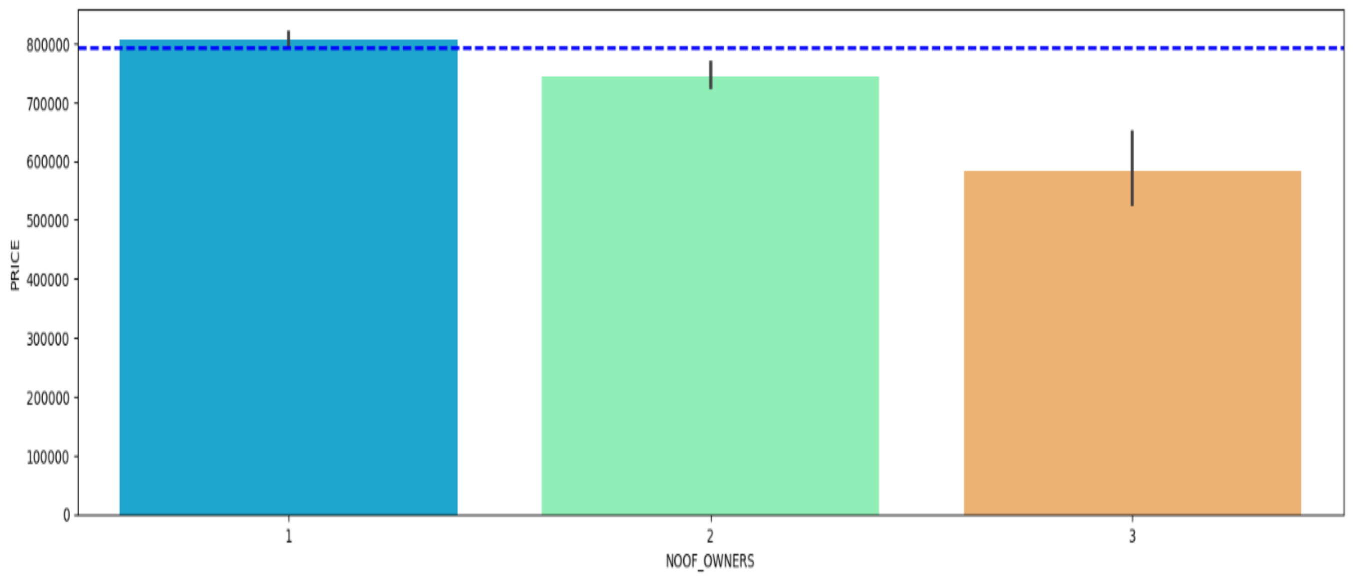
As we can see, the model “Endeavour 2.0 TITANIUM PLUS 4X2 AT” has the highest price, followed by “Endeavour 3.2l 4X4 AT Titanium”.



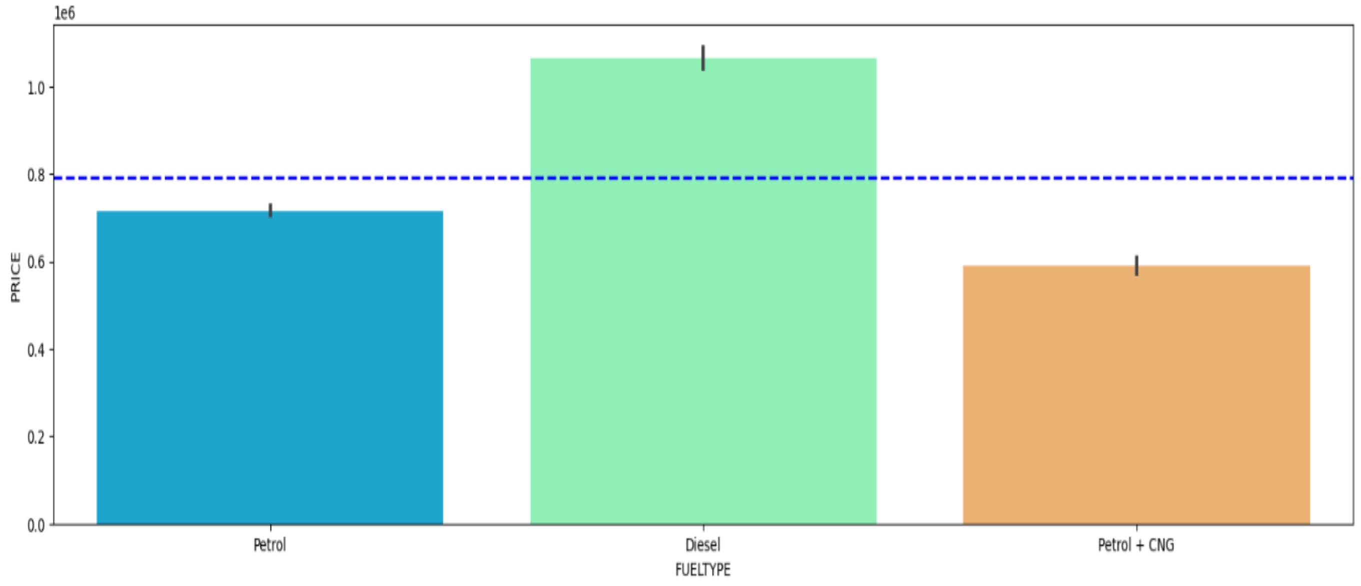
As we can see, the Automatic variant has higher price as compared to Manual variant.



As we can see, the cars which are driven less than 100000 kms have a higher price.

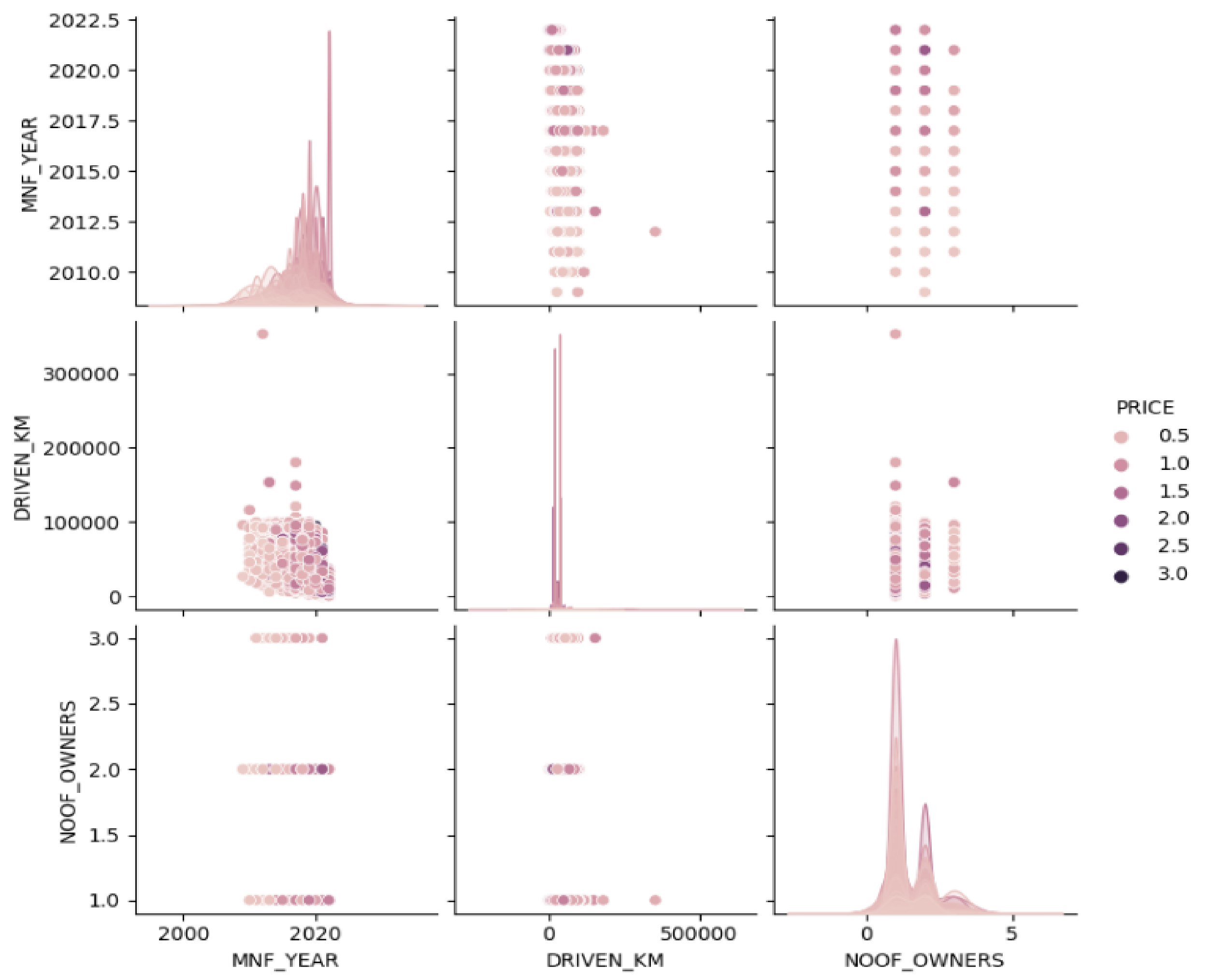


As we can see, the cars with 1 owner have a higher price, followed by cars with 2 owners and 3 owners.



As we can see, the cars with fuel type Diesel have a higher price, followed by cars with fuel type Petrol and Petrol+CNG.

3. Multivariate Analysis:

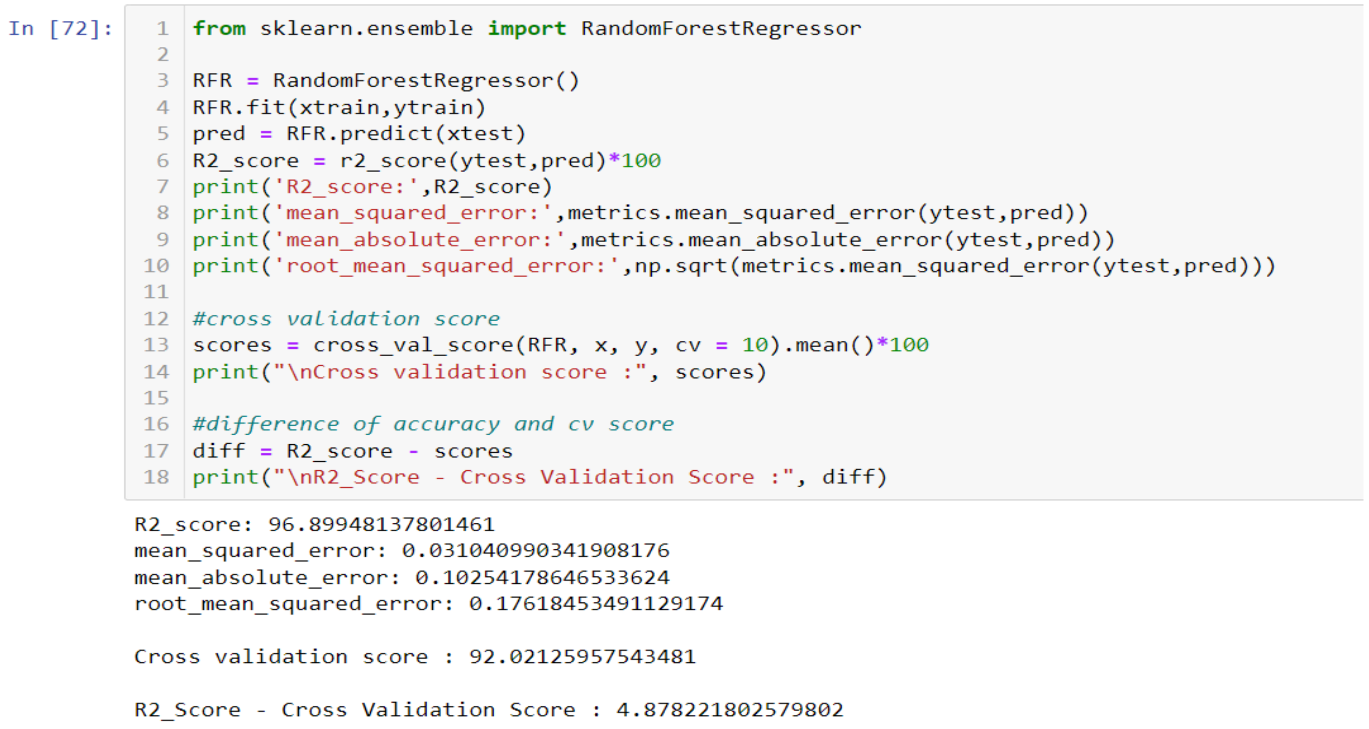


The above pairplot gives the pairwise relationship between the columns, which is plotted on the basis of the target variable 'PRICE'.

3.5 Run and Evaluate selected models

1. Model Building:

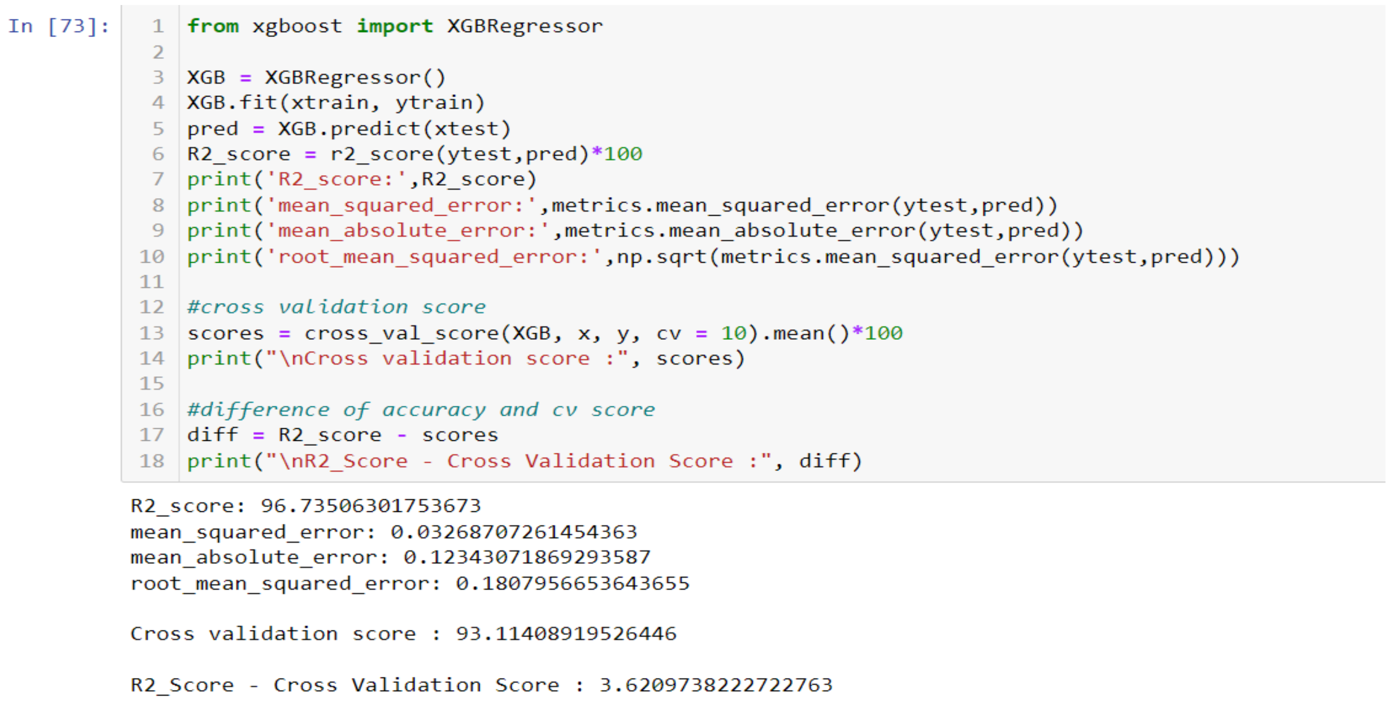
1) Random Forest Regressor:



```
In [72]:
from sklearn.ensemble import RandomForestRegressor

RFR = RandomForestRegressor()
RFR.fit(xtrain,ytrain)
pred = RFR.predict(xtest)
R2_score = r2_score(ytest,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(ytest,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(ytest,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(ytest,pred)))

#cross validation score
scores = cross_val_score(RFR, x, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 96.89948137801461
mean_squared_error: 0.031040990341908176
mean_absolute_error: 0.10254178646533624
root_mean_squared_error: 0.17618453491129174

Cross validation score : 92.02125957543481

R2_Score - Cross Validation Score : 4.878221802579802
```

2) XGB Regressor:



In [73]:

```python
from xgboost import XGBRegressor

XGB = XGBRegressor()
XGB.fit(xtrain, ytrain)
pred = XGB.predict(xtest)
R2_score = r2_score(ytest,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(ytest,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(ytest,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(ytest,pred)))

#cross validation score
scores = cross_val_score(XGB, x, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 96.73506301753673
mean_squared_error: 0.03268707261454363
mean_absolute_error: 0.12343071869293587
root_mean_squared_error: 0.1807956653643655

Cross validation score : 93.11408919526446

R2_Score - Cross Validation Score : 3.6209738222722763
```

3) Extra Trees Regressor:

In [74]:

```
from sklearn.ensemble import ExtraTreesRegressor

ETR = ExtraTreesRegressor()
ETR.fit(xtrain, ytrain)
pred=ETR.predict(xtest)
R2_score = r2_score(ytest,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(ytest,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(ytest,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(ytest,pred)))

#cross validation score
scores = cross_val_score(ETR, x, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

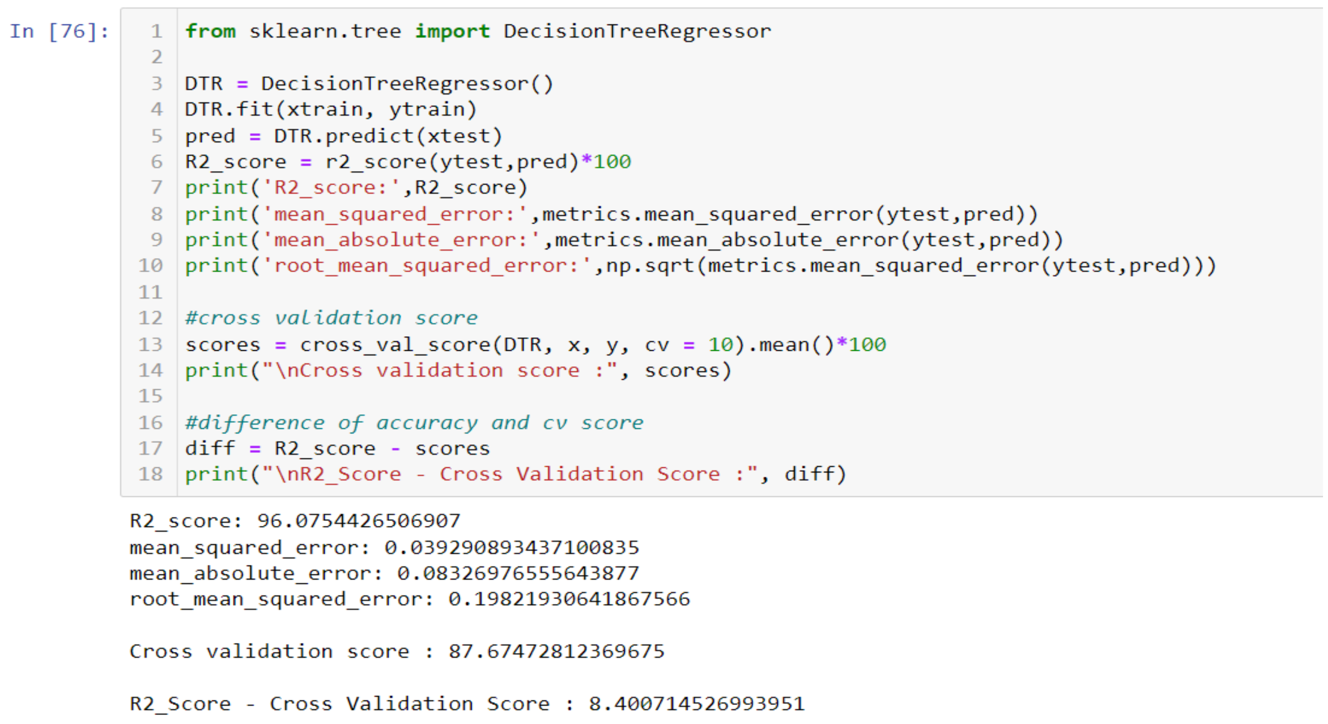
```
R2_score: 96.85939092662481
mean_squared_error: 0.03144235781171447
mean_absolute_error: 0.07582762150000384
root_mean_squared_error: 0.1773199306669007

Cross validation score : 91.48469813989145

R2_Score - Cross Validation Score : 5.3746927867333625
```

4) Gradient Boosting Regressor:

In [75]:
```python
from sklearn.ensemble import GradientBoostingRegressor

GBR = GradientBoostingRegressor()
GBR.fit(xtrain, ytrain)
pred = GBR.predict(xtest)
R2_score = r2_score(ytest,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(ytest,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(ytest,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(ytest,pred)))

#cross validation score
scores = cross_val_score(GBR, x, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 85.23068533674558
mean_squared_error: 0.14786369950105854
mean_absolute_error: 0.2889251949136638
root_mean_squared_error: 0.3845304922903495

Cross validation score : 81.80858149491029

R2_Score - Cross Validation Score : 3.4221038418352947
```

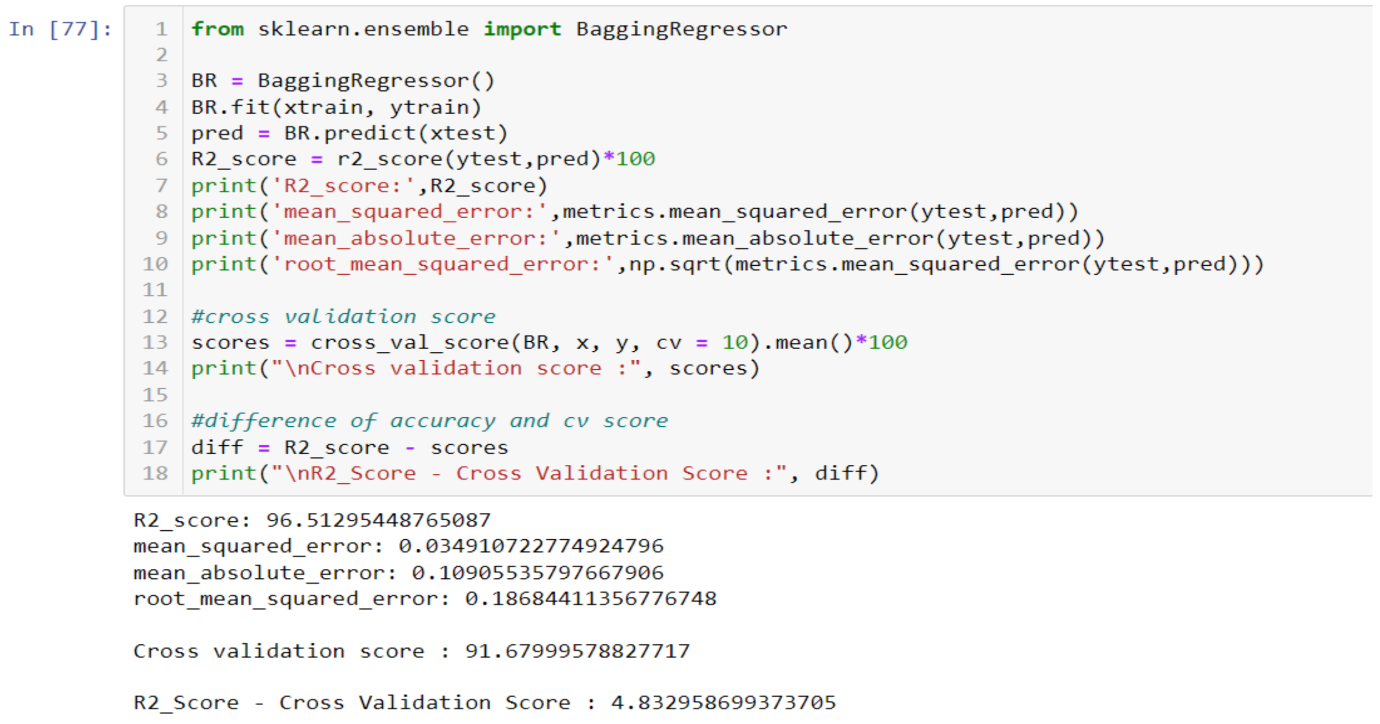
5) Decision Tree Regressor:



```
In [76]:
from sklearn.tree import DecisionTreeRegressor

DTR = DecisionTreeRegressor()
DTR.fit(xtrain, ytrain)
pred = DTR.predict(xtest)
R2_score = r2_score(ytest,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(ytest,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(ytest,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(ytest,pred)))

#cross validation score
scores = cross_val_score(DTR, x, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 96.0754426506907
mean_squared_error: 0.039290893437100835
mean_absolute_error: 0.08326976555643877
root_mean_squared_error: 0.19821930641867566

Cross validation score : 87.67472812369675

R2_Score - Cross Validation Score : 8.400714526993951
```

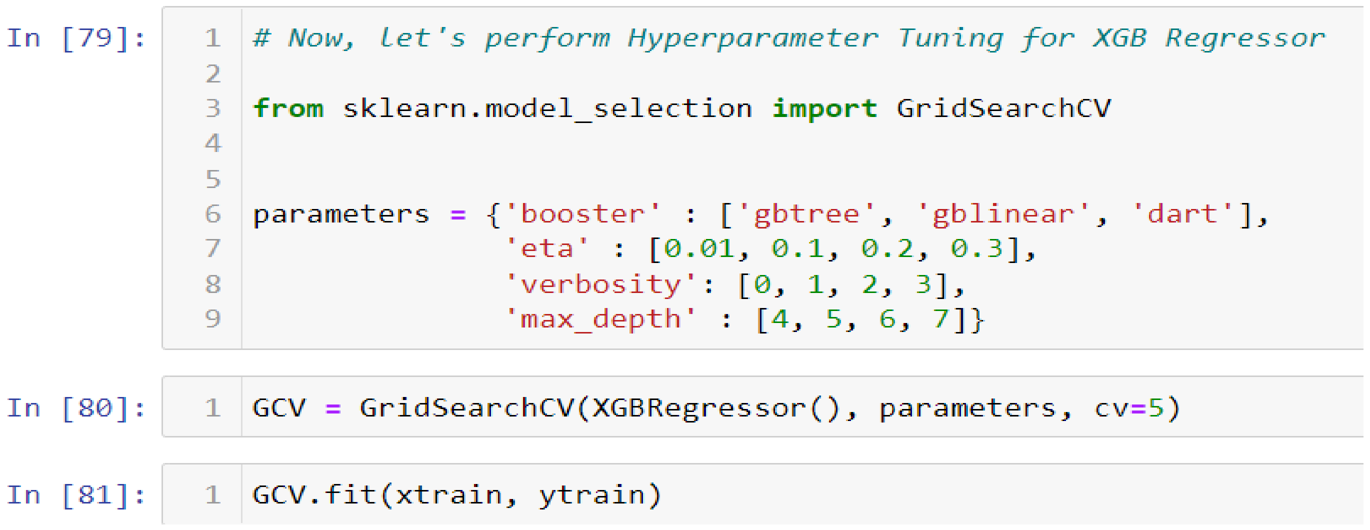
6) Bagging Regressor:



```
In [77]:
from sklearn.ensemble import BaggingRegressor

BR = BaggingRegressor()
BR.fit(xtrain, ytrain)
pred = BR.predict(xtest)
R2_score = r2_score(ytest,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(ytest,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(ytest,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(ytest,pred)))

#cross validation score
scores = cross_val_score(BR, x, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 96.51295448765087
mean_squared_error: 0.034910722774924796
mean_absolute_error: 0.10905535797667906
root_mean_squared_error: 0.18684411356776748

Cross validation score : 91.67999578827717

R2_Score - Cross Validation Score : 4.832958699373705
```

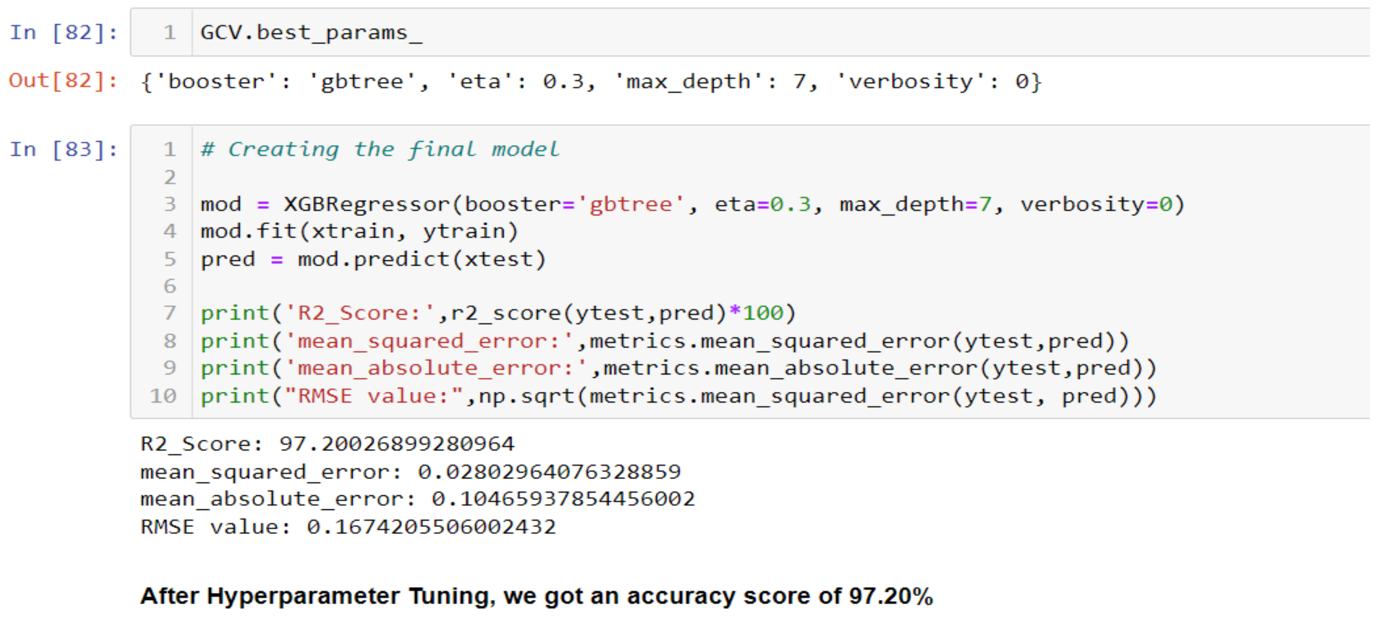
Hyper Parameter Tuning:



```
In [79]:
# Now, let's perform Hyperparameter Tuning for XGB Regressor

from sklearn.model_selection import GridSearchCV


parameters = {'booster' : ['gbtree', 'gblinear', 'dart'],
              'eta' : [0.01, 0.1, 0.2, 0.3],
              'verbosity': [0, 1, 2, 3],
              'max_depth' : [4, 5, 6, 7]}
```

```
In [80]:
GCV = GridSearchCV(XGBRegressor(), parameters, cv=5)
```

```
In [81]:
GCV.fit(xtrain, ytrain)
```



```
In [82]: GCV.best_params_

Out[82]: {'booster': 'gbtree', 'eta': 0.3, 'max_depth': 7, 'verbosity': 0}
```

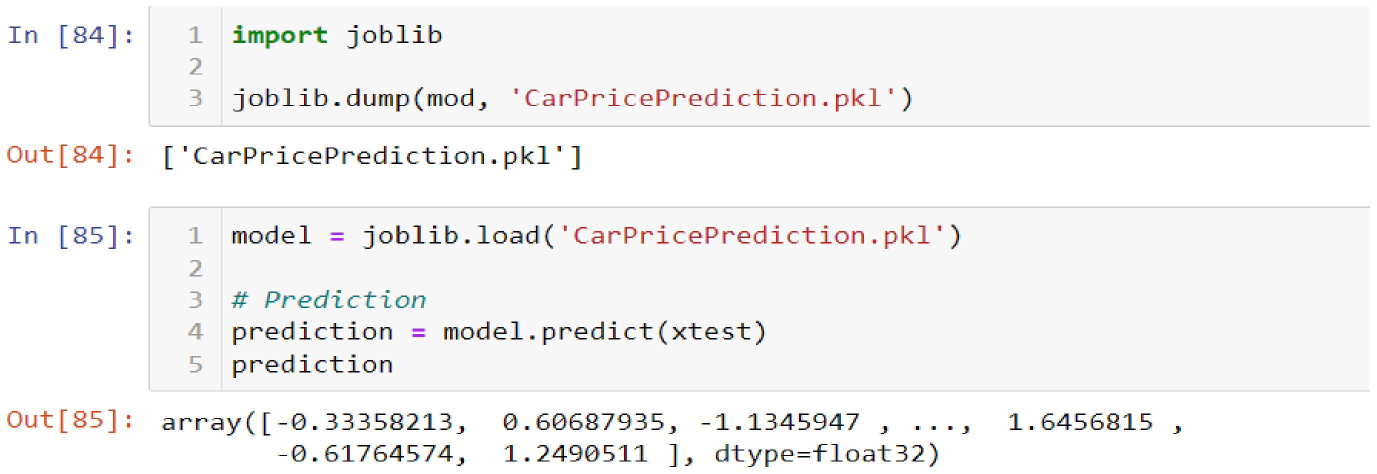
```
In [83]: # Creating the final model

         mod = XGBRegressor(booster='gbtree', eta=0.3, max_depth=7, verbosity=0)
         mod.fit(xtrain, ytrain)
         pred = mod.predict(xtest)

         print('R2_Score:',r2_score(ytest,pred)*100)
         print('mean_squared_error:',metrics.mean_squared_error(ytest,pred))
         print('mean_absolute_error:',metrics.mean_absolute_error(ytest,pred))
         print("RMSE value:",np.sqrt(metrics.mean_squared_error(ytest, pred)))

         R2_Score: 97.20026899280964
         mean_squared_error: 0.02802964076328859
         mean_absolute_error: 0.10465937854456002
         RMSE value: 0.1674205506002432
```

After Hyperparameter Tuning, we got an accuracy score of 97.20%

- ✓ **After Hyperparameter tuning, I have increases the accuracy from 96.73% to 97.20%. Also mse and rmse values has reduced which means error has reduced.**

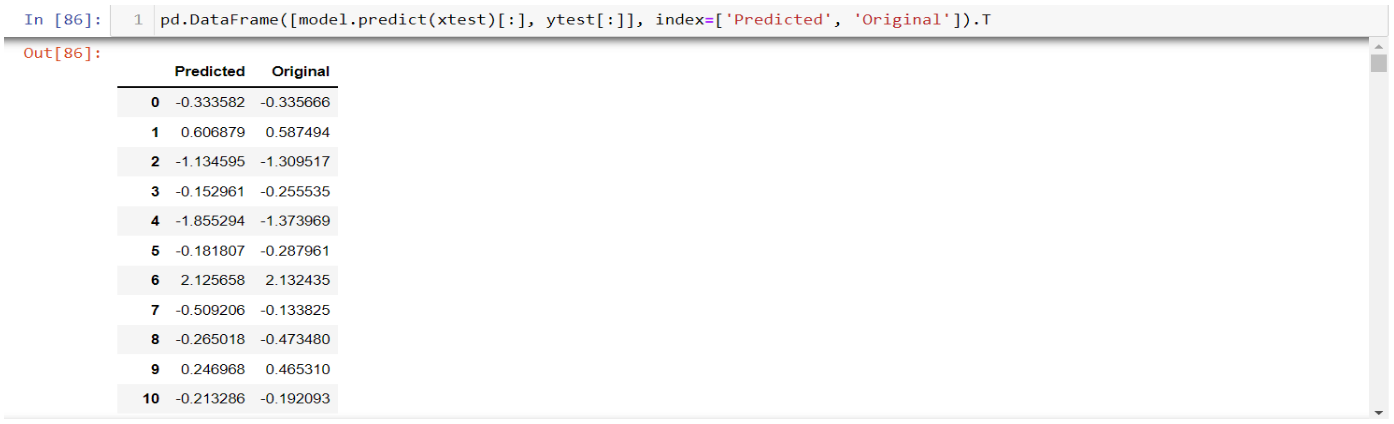
Saving the model and Predicting the Price:

- ✓ I have saved my best model using .pkl as follows.



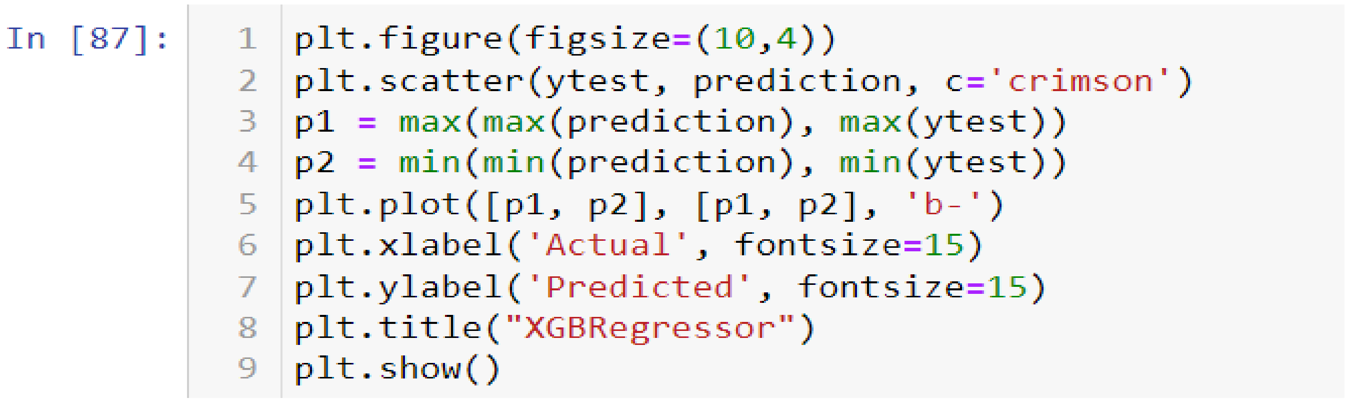
```
In [84]: import joblib

         joblib.dump(mod, 'CarPricePrediction.pkl')

Out[84]: ['CarPricePrediction.pkl']
```

```
In [85]: model = joblib.load('CarPricePrediction.pkl')

         # Prediction
         prediction = model.predict(xtest)
         prediction

Out[85]: array([-0.33358213,  0.60687935, -1.1345947 , ...,  1.6456815 ,
                -0.61764574,  1.2490511 ], dtype=float32)
```

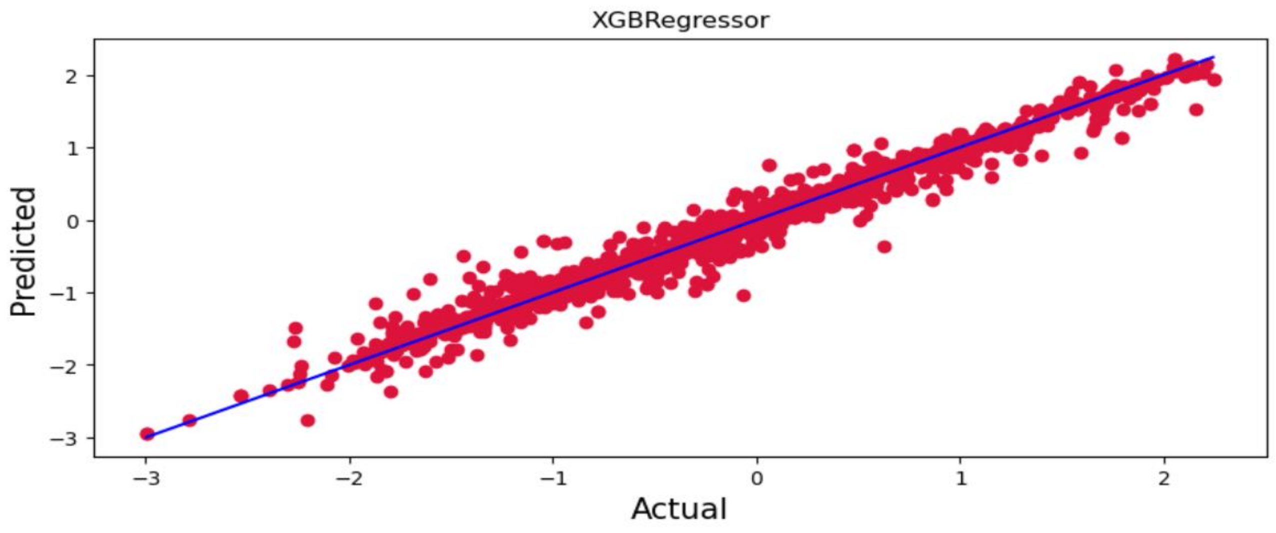


```
In [86]: pd.DataFrame([model.predict(xtest)[:], ytest[:]], index=['Predicted', 'Original']).T
Out[86]:
```

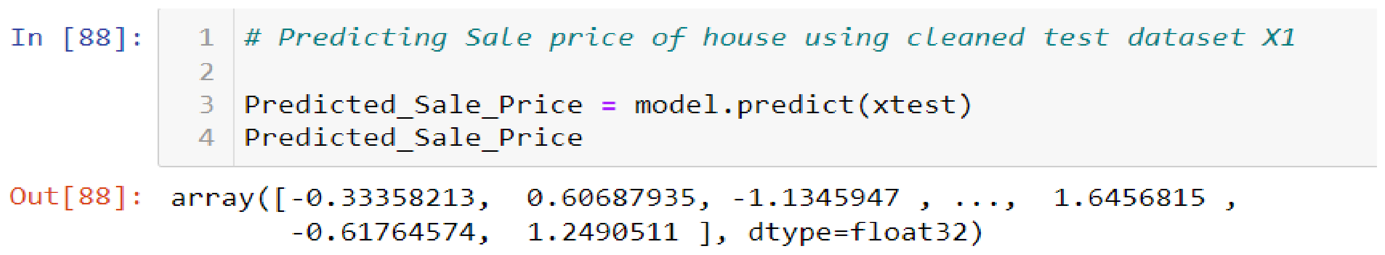
	Predicted	Original
0	-0.333582	-0.335666
1	0.606879	0.587494
2	-1.134595	-1.309517
3	-0.152961	-0.255535
4	-1.855294	-1.373969
5	-0.181807	-0.287961
6	2.125658	2.132435
7	-0.509206	-0.133825
8	-0.265018	-0.473480
9	0.246968	0.465310
10	-0.213286	-0.192093



```
In [87]: plt.figure(figsize=(10,4))
         plt.scatter(ytest, prediction, c='crimson')
         p1 = max(max(prediction), max(ytest))
         p2 = min(min(prediction), min(ytest))
         plt.plot([p1, p2], [p1, p2], 'b-')
         plt.xlabel('Actual', fontsize=15)
         plt.ylabel('Predicted', fontsize=15)
         plt.title("XGBRegressor")
         plt.show()
```

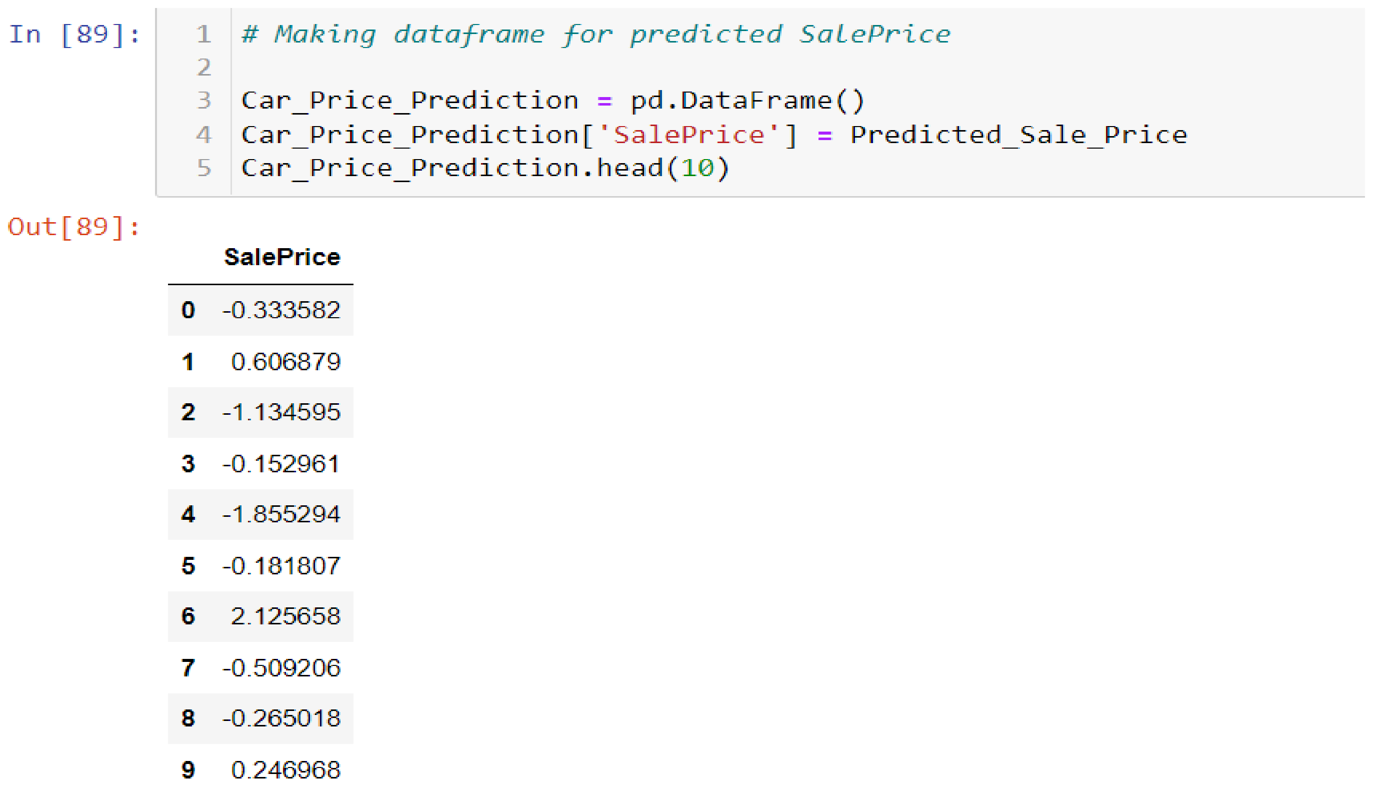


✓ Plotting Actual vs Predicted, To get better insight. Blue line is the actual line and red dots are the predicted values.



```
In [88]:  # Predicting Sale price of house using cleaned test dataset X1

          Predicted_Sale_Price = model.predict(xtest)
          Predicted_Sale_Price
```

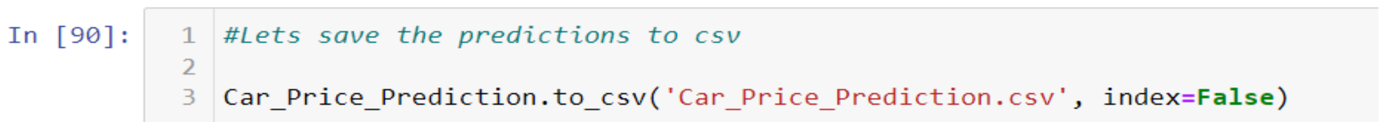
```
Out[88]: array([-0.33358213,  0.60687935, -1.1345947 , ...,  1.6456815 ,
                -0.61764574,  1.2490511 ], dtype=float32)
```



```
In [89]:  # Making dataframe for predicted SalePrice

          Car_Price_Prediction = pd.DataFrame()
          Car_Price_Prediction['SalePrice'] = Predicted_Sale_Price
          Car_Price_Prediction.head(10)
```

Out[89]:

	SalePrice
0	-0.333582
1	0.606879
2	-1.134595
3	-0.152961
4	-1.855294
5	-0.181807
6	2.125658
7	-0.509206
8	-0.265018
9	0.246968



```
In [90]:  #Lets save the predictions to csv

          Car_Price_Prediction.to_csv('Car_Price_Prediction.csv', index=False)
```

- ✓ **I have predicted the Price using the saved model, and the predictions look good. I have also saved my predictions for further analysis.**

3.6 Interpretation of the Results

- ✓ I have replace the null values present in the column VARIANT, with mode of that column.
- ✓ I have used suitable plotting techniques to get better insight on the data.
- ✓ I noticed outliers and skewness in the data. We have to choose proper methods to deal with the outliers and skewness. If we ignore the outliers and skewness, we may end up with a bad model which has less accuracy.
- ✓ Then, I have scaled the data, as it will have a good impact, i.e., it will prevent the model from getting baised.
- ✓ We have to use multiple models while building the model, so that we can select the best performing model.
- ✓ We have to use multiple metrics like mae, mse, rmse and r2_score, which will help us to decide the best model.
- ✓ I found XGBRegressor as the best model with 96.73% r2_score. I have improved the accuracy of the best model to 97.20%, by running hyper parameter tuning.
- ✓ At last, I have predicted the Price for using the saved model.

4.Conclusion

4.1 Key Findings and Conclusions of the Study

- ✓ The cars manufactured in the year 2022 have a higher price, followed by cars manufactured in the year 2021 and 2020. This means that, the latest cars have a higher price.
- ✓ The brand “MG” has the highest price, followed by “Jaguar” and “Audi”. This means that the luxury cars have a higher price.
- ✓ The model “Endeavour 2.0 TITANIUM PLUS 4X2 AT” has the highest price, followed by “Endeavour 3.2l 4X4 AT Titanium”.
- ✓ The Automatic variant has higher price as compared to Manual variant.
- ✓ The cars which are driven less than 100000 kms have a higher price.
- ✓ The cars with 1 owner have a higher price, followed by cars with 2 owners and 3 owners. This means that, as the number of owners increases, the price of the car decreases.
- ✓ The cars with fuel type Diesel have a higher price, followed by cars with fuel type Petrol and Petrol+CNG.

4.2 Learning Outcomes of the Study in respect of Data Science

The above research will help our client to study about the latest used car market and with the help of the model built he can easily predict the price ranges of the cars, and also will helps him to understand based on what factors the Car Price is decided.

4.3 Limitations of this work and Scope for Future Work

The limitation of the study is that in the volatile changing market we have taken the data, to be more precise we have taken the data at the time of pandemic, so when the pandemic ends the market correction might happen slowly. So based on that again the deciding factors of the used car prize might change and we have shortlisted and taken these data from the important cities across India, if the seller is from the different city our model might fail to predict the accurate prize of that used car.