FLIP ROBO

CAR : PRICE PREDICTION

Submitted by:

CHESTHA SHARMA

ACKNOWLEDGMENT

The project entitled “Car: Price Prediction” is done by me during my internship with Flip Robo Technologies. I am grateful to Data Trained and Flip Robo Technologies for their guidance during this project.

INTRODUCTION

- Cars are one of the necessary need of each and every person around the globe and therefore car market is the market which is one of the major contributors in the world’s economy. It is a very large market and there are various companies working in the domain.

- Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in car sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for car selling companies. Our problem is related to one such car-selling client.

- 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 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.

- 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 the clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, 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.

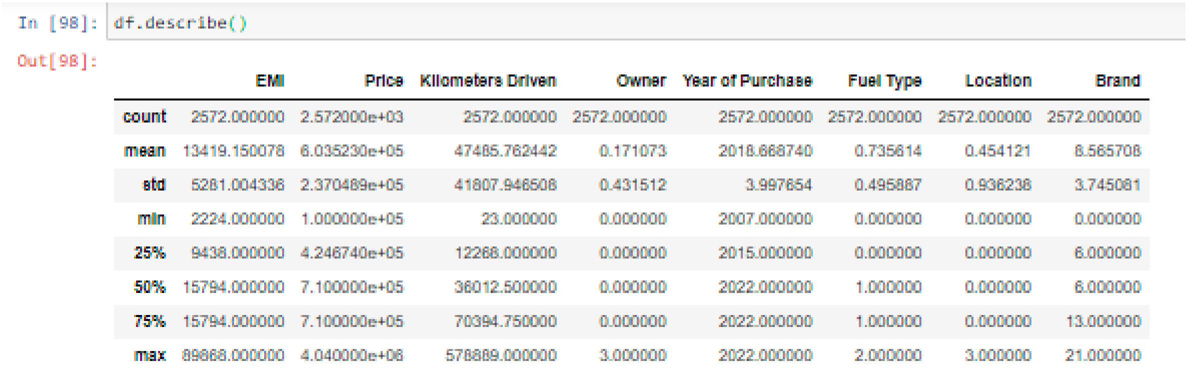
- For this client wants to know:

  - Which variables are important to predict the price of variable?
  - How do these variables describe the price of the cars?

Analytical Problem Framing

• Mathematical/ Analytical Modeling of the Problem

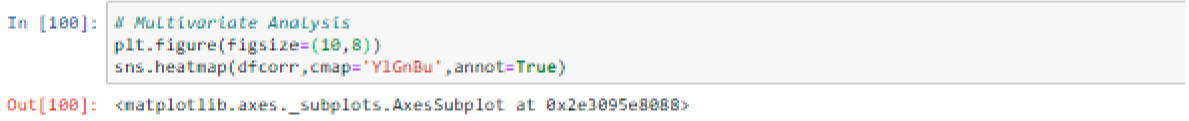
In this project we have performed various mathematical and statistical analysis such as we checked description or statistical summary of the data using describe, checked correlation using corr and also visualized it using heatmap. Then we have used zscore to plot outliers and remove them.



```
In [98]: df.describe()
```

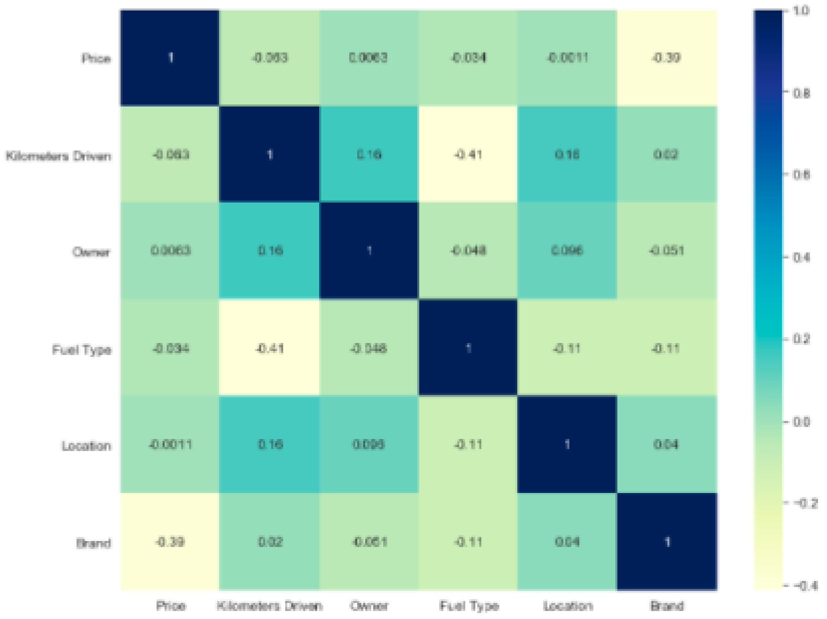
Out[98]:

	EMI	Price	Kilometers Driven	Owner	Year of Purchase	Fuel Type	Location	Brand
count	2572.000000	2.572000e+03	2572.000000	2572.000000	2572.000000	2572.000000	2572.000000	2572.000000
mean	13419.150078	6.035230e+05	47485.762442	0.171073	2018.668740	0.735614	0.454121	8.565708
std	5281.004336	2.370489e+05	41807.946508	0.431512	3.997654	0.495887	0.936238	3.745081
min	2224.000000	1.000000e+05	23.000000	0.000000	2007.000000	0.000000	0.000000	0.000000
25%	9438.000000	4.246740e+05	12268.000000	0.000000	2015.000000	0.000000	0.000000	6.000000
50%	15794.000000	7.100000e+05	36012.500000	0.000000	2022.000000	1.000000	0.000000	6.000000
75%	15794.000000	7.100000e+05	70394.750000	0.000000	2022.000000	1.000000	0.000000	13.000000
max	89868.000000	4.040000e+06	578889.000000	3.000000	2022.000000	2.000000	3.000000	21.000000



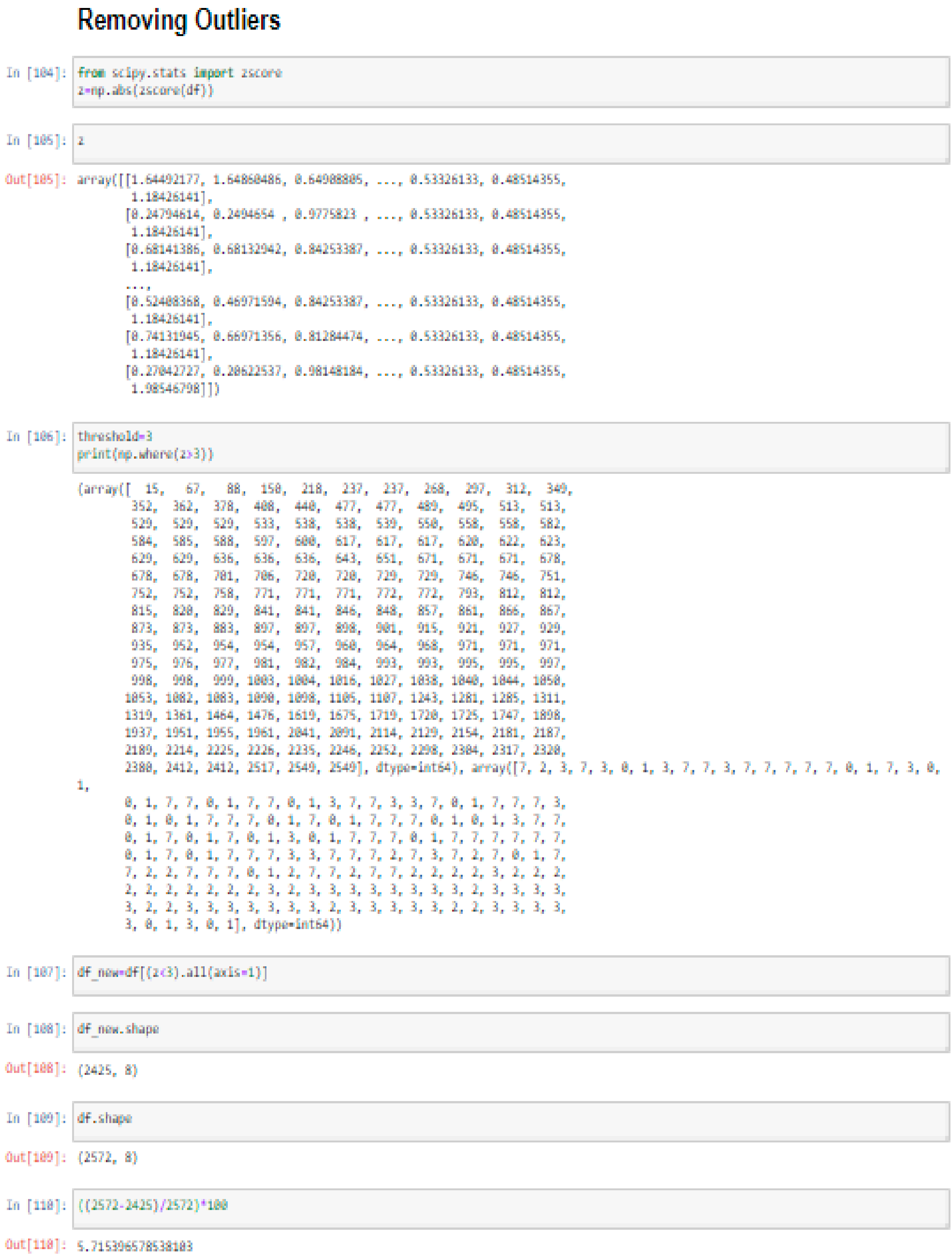
```
In [100]: # Multivariate Analysis
          plt.figure(figsize=(10,8))
          sns.heatmap(dfcorr,cmap='YlGnBu',annot=True)

Out[100]: <matplotlib.axes._subplots.AxesSubplot at 0x2e3095e8088>
```



	Price	Kilometers Driven	Owner	Fuel Type	Location	Brand
Price	1	-0.063	0.0063	-0.034	-0.0011	-0.39
Kilometers Driven	-0.063	1	0.16	-0.41	0.16	0.02
Owner	0.0063	0.16	1	-0.048	0.096	-0.051
Fuel Type	-0.034	-0.41	-0.048	1	-0.11	-0.11
Location	-0.0011	0.16	0.096	-0.11	1	0.04
Brand	-0.39	0.02	-0.051	-0.11	0.04	1

Removing the Outliers using Z-score



Removing Outliers

```
In [104]: from scipy.stats import zscore
          z=np.abs(zscore(df))
```

```
In [105]: z
```

```
Out[105]: array([[1.64492177, 1.64860486, 0.64988805, ..., 0.53326133, 0.48514355,
                  1.18426141],
                 [0.24794614, 0.2494654 , 0.9775823 , ..., 0.53326133, 0.48514355,
                  1.18426141],
                 [0.68141386, 0.68132942, 0.84253387, ..., 0.53326133, 0.48514355,
                  1.18426141],
                 ...,
                 [0.52408368, 0.46971594, 0.84253387, ..., 0.53326133, 0.48514355,
                  1.18426141],
                 [0.74131945, 0.66971356, 0.81284474, ..., 0.53326133, 0.48514355,
                  1.18426141],
                 [0.27042727, 0.28622537, 0.98148184, ..., 0.53326133, 0.48514355,
                  1.98546798]])
```

```
In [106]: threshold=3
          print(np.where(z>3))
```

```
(array([  15,   67,   88,  150,  218,  237,  237,  268,  297,  312,  349,
         352,  362,  378,  408,  440,  477,  477,  489,  495,  513,  513,
         529,  529,  529,  533,  538,  538,  539,  550,  558,  558,  582,
         584,  585,  588,  597,  600,  617,  617,  617,  620,  622,  623,
         629,  629,  636,  636,  636,  643,  651,  671,  671,  671,  678,
         678,  678,  701,  706,  720,  720,  729,  729,  746,  746,  751,
         752,  752,  758,  771,  771,  771,  772,  772,  793,  812,  812,
         815,  820,  829,  841,  841,  846,  848,  857,  861,  866,  867,
         873,  873,  883,  897,  897,  898,  901,  915,  921,  927,  929,
         935,  952,  954,  954,  957,  960,  964,  968,  971,  971,  971,
         975,  976,  977,  981,  982,  984,  993,  993,  995,  995,  997,
         998,  998,  999, 1003, 1004, 1016, 1027, 1038, 1040, 1044, 1050,
        1053, 1082, 1083, 1090, 1098, 1105, 1107, 1243, 1281, 1285, 1311,
        1319, 1361, 1464, 1476, 1619, 1675, 1719, 1720, 1725, 1747, 1898,
        1937, 1951, 1955, 1961, 2041, 2091, 2114, 2129, 2154, 2181, 2187,
        2189, 2214, 2225, 2226, 2235, 2246, 2252, 2298, 2304, 2317, 2320,
        2380, 2412, 2412, 2517, 2549, 2549], dtype=int64), array([7, 2, 3, 7, 3, 0, 1, 3, 7, 7, 3, 7, 7, 7, 7, 7, 0, 1, 7, 3, 0,
1,
        0, 1, 7, 7, 0, 1, 7, 7, 0, 1, 3, 7, 7, 3, 3, 7, 0, 1, 7, 7, 7, 3,
        0, 1, 0, 1, 7, 7, 7, 0, 1, 7, 0, 1, 7, 7, 7, 0, 1, 0, 1, 3, 7, 7,
        0, 1, 7, 0, 1, 7, 0, 1, 3, 0, 1, 7, 7, 7, 0, 1, 7, 7, 7, 7, 7, 7,
        0, 1, 7, 0, 1, 7, 7, 7, 3, 3, 7, 7, 7, 2, 7, 3, 7, 2, 7, 0, 1, 7,
        7, 2, 2, 7, 7, 7, 0, 1, 2, 7, 7, 2, 7, 7, 2, 2, 2, 2, 3, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3,
        3, 2, 2, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 2, 2, 3, 3, 3, 3,
        3, 0, 1, 3, 0, 1], dtype=int64))
```

```
In [107]: df_new=df[(z<3).all(axis=1)]
```

```
In [108]: df_new.shape
```

```
Out[108]: (2425, 8)
```

```
In [109]: df.shape
```

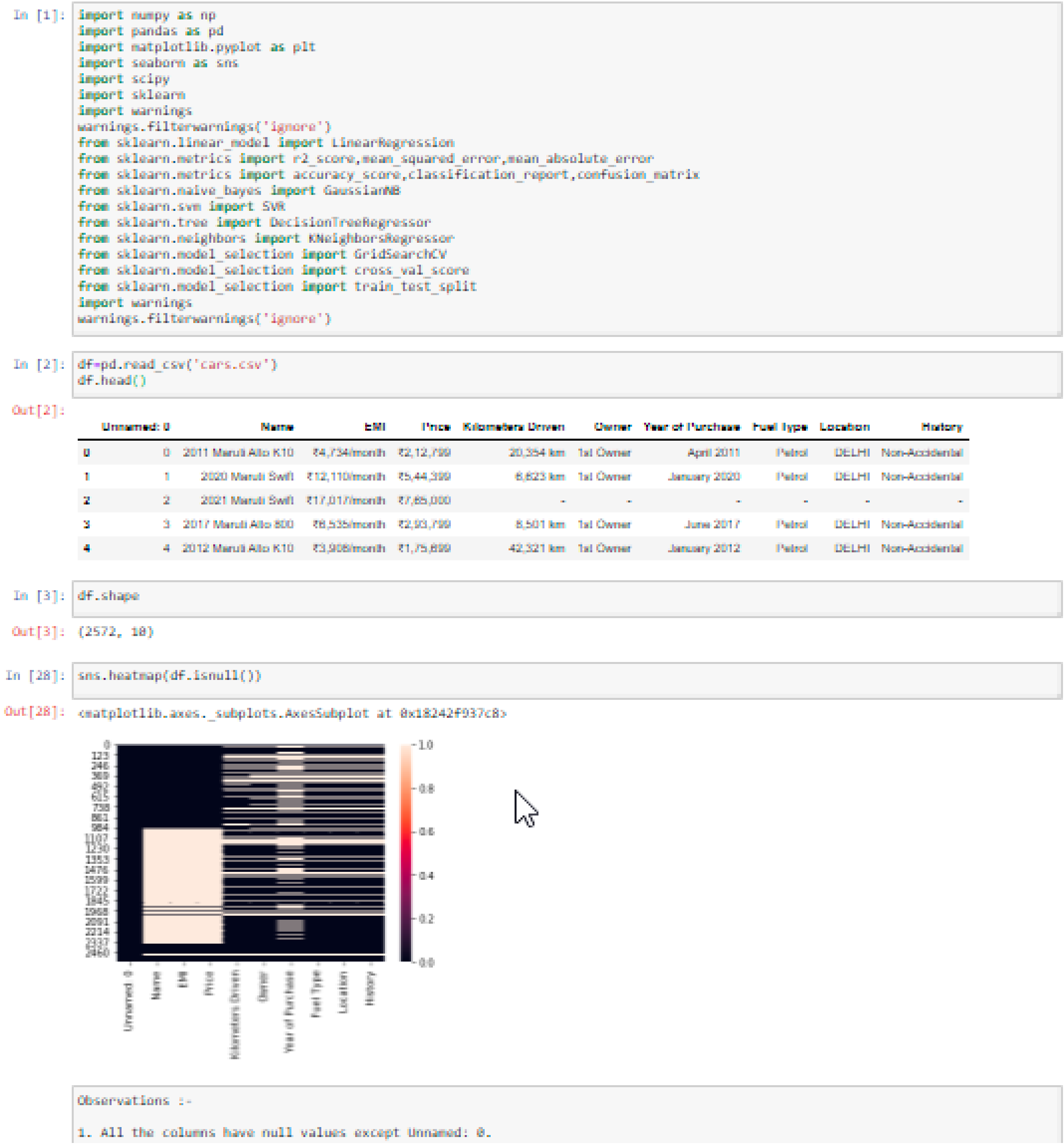
```
Out[109]: (2572, 8)
```

```
In [110]: ((2572-2425)/2572)*100
```

```
Out[110]: 5.715396578538103
```

- ## Data Sources and their formats

The sample data is extracted by using web scraping from selenium. It is stored in csv format and hence we import it using pandas. Then we further checked more about data using info, checked data types using dtypes, shapes using .shape, columns using .columns, null values using .isnull.sum, and further visualize it through heatmap as follows:



In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
import sklearn
import warnings
warnings.filterwarnings('ignore')
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df=pd.read_csv('cars.csv')
df.head()
```

Out[2]:

	Unnamed: 0	Name	EMI	Price	Kilometers Driven	Owner	Year of Purchase	Fuel Type	Location	History
0	0	2011 Maruti Alto K10	₹4,734/month	₹2,12,799	20,354 km	1st Owner	April 2011	Petrol	DELHI	Non-Accidental
1	1	2020 Maruti Swift	₹12,110/month	₹5,44,399	6,623 km	1st Owner	January 2020	Petrol	DELHI	Non-Accidental
2	2	2021 Maruti Swift	₹17,017/month	₹7,65,000	-	-	-	-	-	-
3	3	2017 Maruti Alto 800	₹6,535/month	₹2,93,799	8,501 km	1st Owner	June 2017	Petrol	DELHI	Non-Accidental
4	4	2012 Maruti Alto K10	₹3,908/month	₹1,75,899	42,321 km	1st Owner	January 2012	Petrol	DELHI	Non-Accidental

In [3]:
```
df.shape
```

Out[3]:
```
(2572, 10)
```

In [28]:
```
sns.heatmap(df.isnull())
```

Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x18242f937c8>
```

```
Observations :-

1. All the columns have null values except Unnamed: 0.
```

- ## Data Preprocessing Done

First we will determine whether there are any null values and since there were null values as well as NaN values present in the dataset we proceeded further by imputing them using Simple Imputer with mean and most frequent as strategies respectively. Next we did Label encoding using label encoder. Then we performed some data visualization in which we observed certain attributes were having skewness and outliers that were plotted using distplot and boxplot. Outliers were removed with the help of Zscore in which 147 rows were removed.

Data Inputs- Logic- Output Relationships

The data consists of 9 inputs and one output-"Price". Kilometers Driven is the least/negatively correlated column with target('Price') variable. Owner is highly correlated column with target variable followed by other attributes.

Hardware and Software Requirements and Tools Used

In this project we have used HP Pavilion PC with 64-bit operating system and have Windows 10 pro. We have used python to develop this project in which we have used various libraries such as numpy, pandas, matplotlib, seaborn for handling data or arrays and their visualization. For statistical purpose we have used zscore from scipy.stats to remove outliers. Lastly, to develop the model we have used various libraries and metrics from sklearn such as train_test_split, Linear Regression, Lasso, Ridge, Elastic Net, SVR, Decision Tree Regressor, KNeighbors Regressor, Random Forest Regressor,AdaBoostRegressor, mean_squared_error, mean_absolute_error and r2_score.

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

We have performed various mathematical and statistical analysis such as we checked description or statistical summary of the data using describe, checked correlation using corr and also visualized it using heatmap. Then we have used zscore to plot outliers and remove them. We have used distplot to find the distribution of all attributes.

Testing of Identified Approaches (Algorithms)

We have used following algorithms such as: LinearRegression, Lasso, Ridge, ElasticNet,SVR,DecisionTreeRegressor,KNeighborsRegressor,RandomForest Regressor and AdaBoostRegressor.

Run and Evaluate selected models

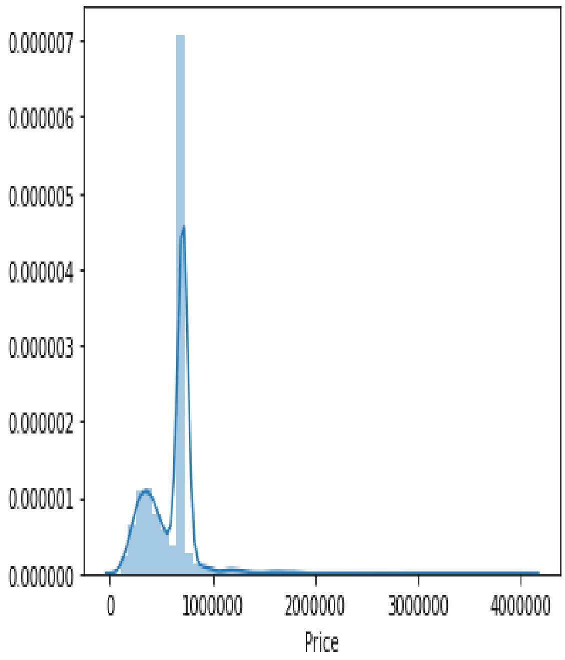
We have formed a loop where all the algorithms will be used one by one and their corresponding Score, Mean Absolute Error, Mean Squared Error, RMSE and r2_score will be evaluated.

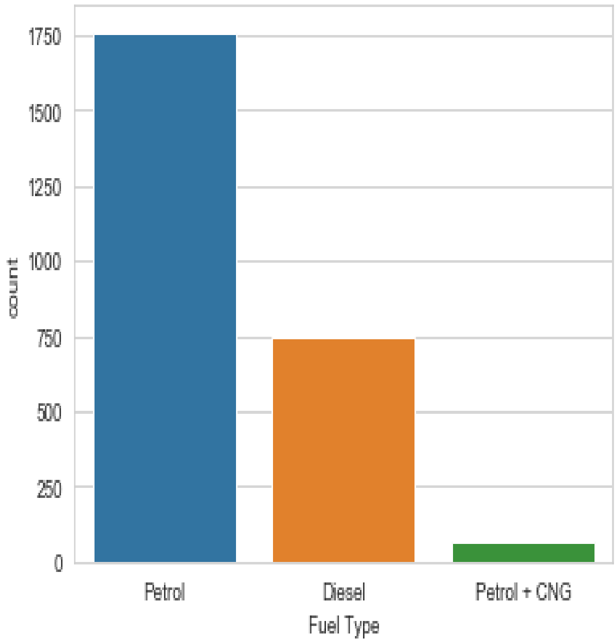
- I chose Decision Tree Regressor as our best model since it's giving us best score and it's performing well. It's r2_score is also satisfactory and it shows that our model is neither underfitting/overfitting. Then there is no need to perform hyperparamter as it is giving 100% accuracy. Its r2_score is also sat isfactory. Hence we saved Decision Tree Regressor as our final model using joblib.

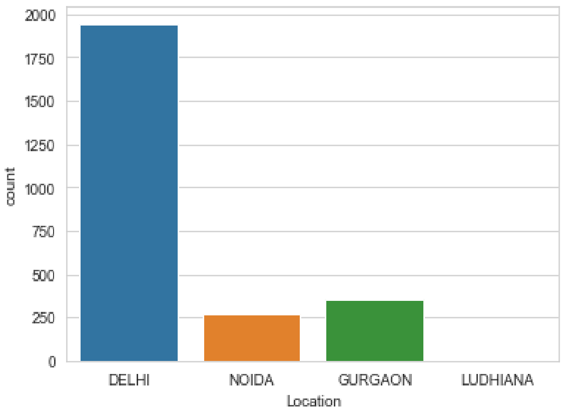
Key Metrics for success in solving problem under consideration

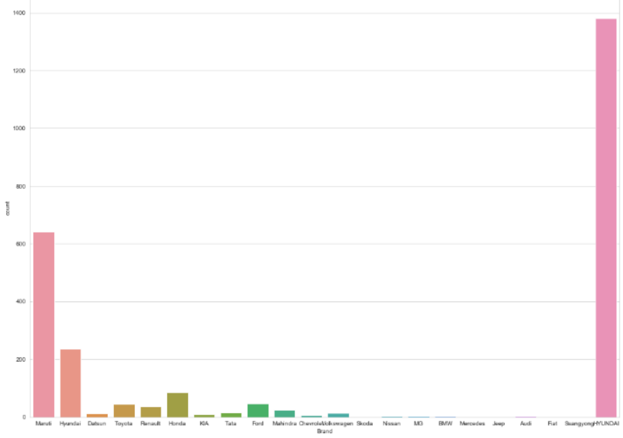
Key metrics used for finalising the model was Score and r2_score. Since in case of Decision Tree Regressor it's giving us good score among all other models and it's performing well. It's r2_score is also satisfactory and it shows that our model is neither underfitting/overfitting .

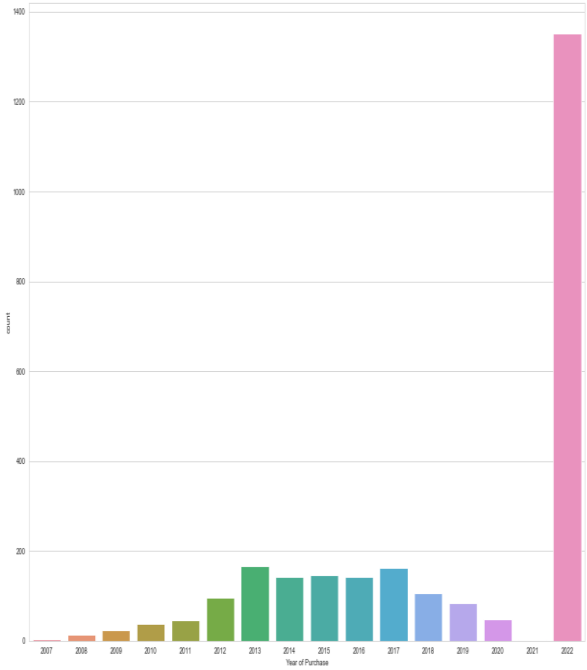
Data Visualization

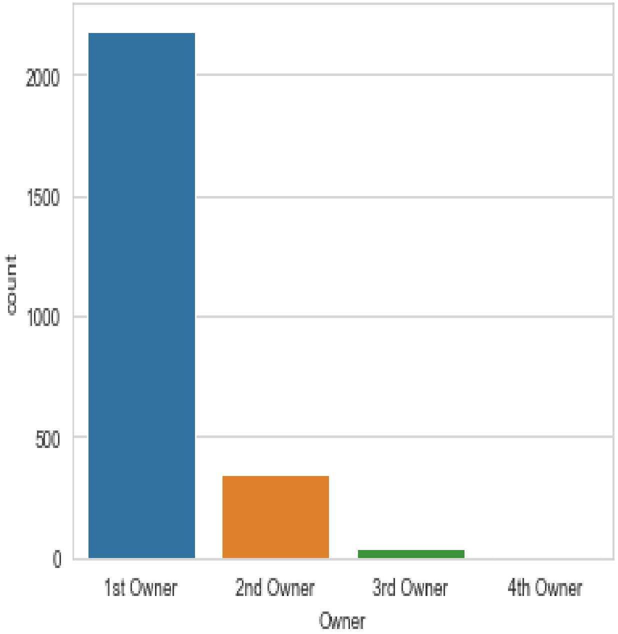


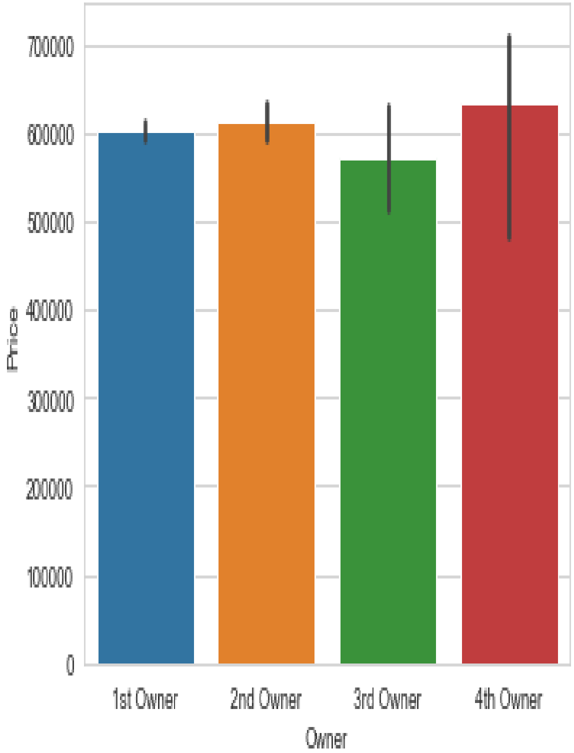


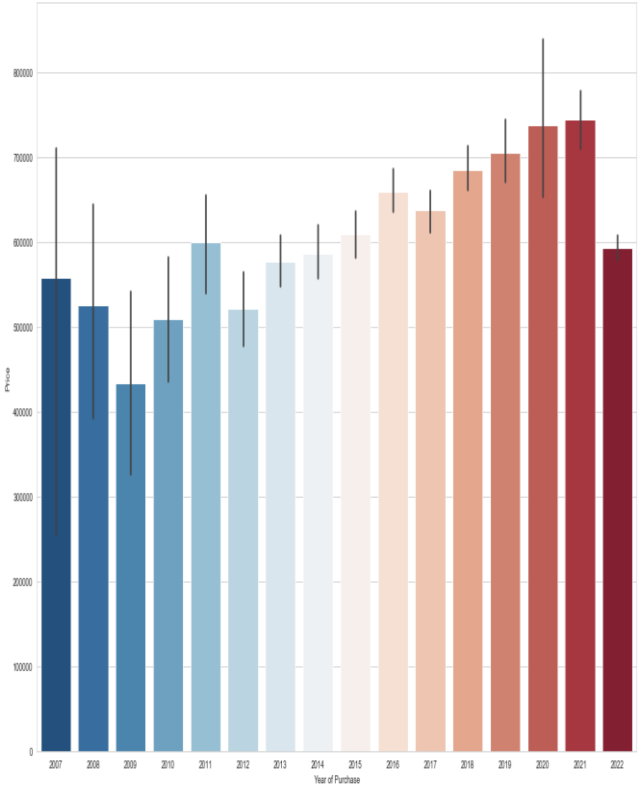


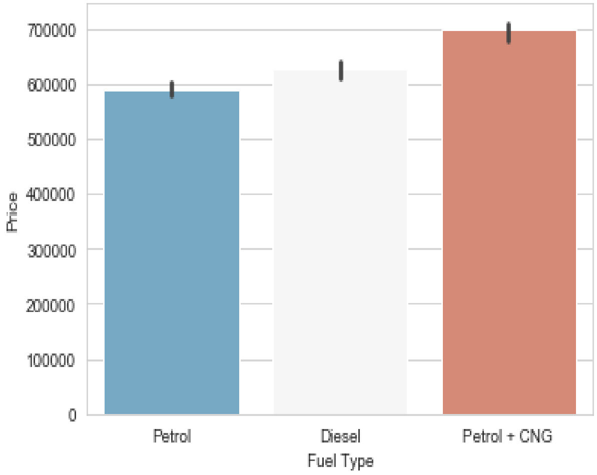


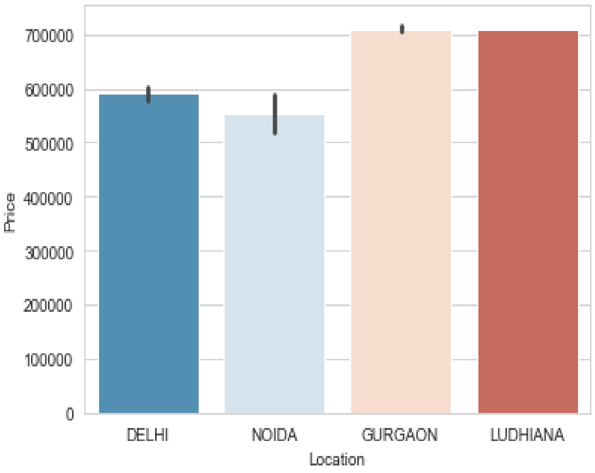

1400
1200
1000
800
600
400
200
0
count
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Year of Purchase

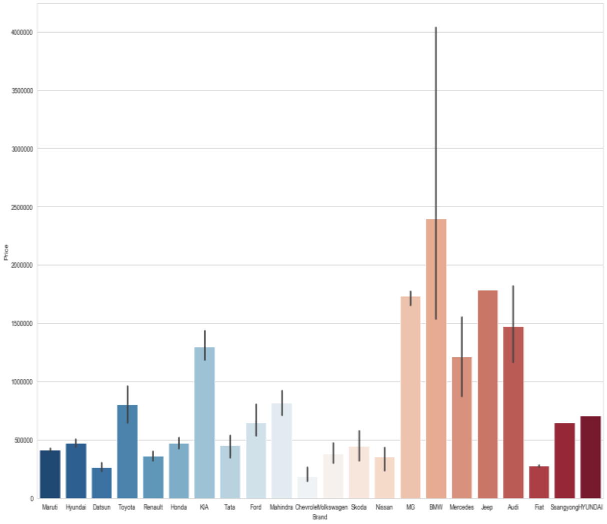

2000
1500
1000
500
0
count
1st Owner
2nd Owner
3rd Owner
4th Owner
Owner

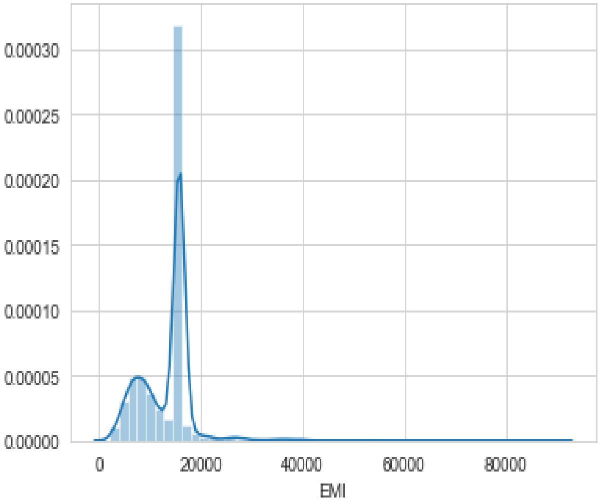

700000
600000
500000
400000
300000
200000
100000
0
Price
1st Owner
2nd Owner
3rd Owner
4th Owner
Owner

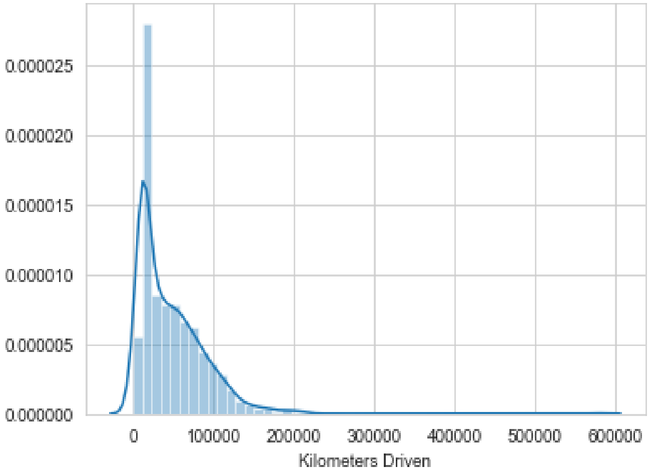

800000
700000
600000
500000
400000
300000
200000
100000
0
Price
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Year of Purchase


700000
600000
500000
400000
300000
200000
100000
0
Price
Petrol
Diesel
Petrol + CNG
Fuel Type


700000
600000
500000
400000
300000
200000
100000
0
Price
DELHI
NOIDA
GURGAON
LUDHIANA
Location


4000000
3500000
3000000
2500000
2000000
1500000
1000000
500000
0
Price
Maruti
Hyundai
Datsun
Toyota
Renault
Honda
KIA
Tata
Ford
Mahindra
Chevrolet
Volkswagen
Skoda
Nissan
MG
BMW
Mercedes
Jeep
Audi
Fiat
Ssangyong
HYUNDAI
Brand


0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
0
20000
40000
60000
80000
EMI


0.000025
0.000020
0.000015
0.000010
0.000005
0.000000
0
100000
200000
300000
400000
500000
600000
Kilometers Driven

Conclusion

- Price of cars is majorly less than 1000000.
- Majorly cars are driven by petrol.
- Least number of cars are driven by Petrol + CNG.
- Highest number of cars are located in Delhi.
- Lowest number of cars are located in Ludhiana.
- Hyundai cars are present majorly in the dataset.
- Jeep and SsangYong are present in the least number in dataset.
- High number of cars whose brands are Maruti and Hyundai are pres ent in the dataset.
- Low number of cars whose brands are KIA, Chevrolet, Nissan, Audi, MG, BMW, Mercedes, Skoda and Fiat are present in the dataset.
- Maximum number of cars are purchased in the year 2022.
- Minimum number of cars are purchased in the year 2021.
- High number of cars are purchased in the year 2013, 2017 and 2015 also.
- Low number of cars are purchased in the year 2009, 2008 and 2007 also.
- First owners are present majorly in the dataset.
- Fourth owners are present in least number in the dataset.
- Fourth owned cars have the highest price.
- Third owned cars have the lowest price.
- Cars purchased in the year 2021 have the highest prices.
- Cars purchased in the year 2009 have the lowest prices.
- Cars purchased in the year 2020, 2019, 2018, 2017 and 2015 have also high prices.
- Cars purchased in the year 2009, 2010, 2012 and 2008 have also low prices.
- Cars, which uses Petrol + CNG, have the highest price.
- Cars that uses only Petrol have the lowest price.
- Cars in the location of Gurgaon and Ludhiana have the highest price
- Cars in the location of Noida have the lowest prices.
- BMW has the highest price.
- Datson, Chevrolet, Nisaan and Fiat have the lowest prices.
- Jeep, MG and Audi also have high prices.

- EMI price is majorly less than 20000 in the dataset.
- Very few number of cars have EMI greater than 20000.
- Highest number of cars have EMI in the range of 15000-20000
- Kilometers Driven are majorly less than 100000 km in the dataset.
- Very few number of cars have Kilometers Driven greater than 2000 00 km in the dataset.

Learning Outcomes of the Study in respect of Data Science

With the help of visualization tools such as matplotlib and seaborn we have visualized the impact of each attributes on our target variable. For cleaning the data and plotting outliers we have used distplot and boxplot and for removing outliers we have used zscore which is a statistical tool. At last we got Decision Tree as our best model.

Limitations of this work and Scope for Future Work

The model is working well and we have performed hyperparameter tuning and we have concluded our project by choosing Decision tree Regre ssor as our best model.