FLIP ROBO

Car Price Prediction

Submitted by:

Priya Rajput

ACKNOWLEDGMENT

I would like to express my gratitude towards Flip-Robo for providing me this opportunity to show case my talent and also for their constant support and guidance. Also It is indeed a pleasure for me to have worked on this project.

I express my deepest thanks to Miss Sapna Verma, for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make my project easier. I choose this moment to acknowledge his contribution gratefully.

TABLE OF CONTENTS

INTRODUCTION

Business Problem Framing

In this project, we have to predict car price valuation using new machine learning models using the dataset. Because with the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models.

Conceptual Background of the Domain Problem

1. Prepare our own dataset using web scraping.
2. Check whether the project is a regression type or a classification type.
3. Check whether our dataset is balanced or imbalanced. If it is an imbalanced one, we will apply sampling techniques to balance the dataset.
4. Model building and find the accuracy of the model.
5. Build a model with good accuracy and also go for hyperparameter tuning.

REVIEW OF LITERATURE

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.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

If you look at data science, we are actually using mathematical models to model (and hopefully through the model to explain some of the things that we have seen) business circumstances, environment etc and through these model, we can get more insights such as the outcomes of our decision undertaken, what should we do next or how shall we do it to improve the odds. So mathematical models are important, selecting the right one to answer the business question can bring tremendous value to the organization. Here we are using AdaBoostRegressor with accuracy 77% after hyper parameter tuning.

DATA SOURCES AND THEIR FORMATS

Data Source: The read_csv function of the pandas library is used to read the content of a CSV file into the python environment as a pandas DataFrame. The function can read the files from the OS by using proper path to the file. Data description: Pandas describe() is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values.

Data Pre-processing Done

First, we have imported the necessary libraries and dataset.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
#Importing CSV file from the Dataset
df=pd.read_csv('Cars24.csv')
df.head()
```

	Unnamed: 0	Brand	Model	Year	Variant	Location	Version	Number of Owners	KmDriven	Price
0	0	Maruti	Swift Dzire	2013	LDI BS IV Manual	New Delhi	Diesel	2nd Owner	49944	303099
1	1	Maruti	Swift	2012	VDI Manual	New Delhi	Diesel	1st Owner	129639	297999
2	2	Volkswagen	Vento	2014	HIGHLINE PETROL Manual	New Delhi	Petrol	1st Owner	62625	446899
3	3	Maruti	Ertiga	2014	VDI ABS Manual	New Delhi	Diesel	1st Owner	64013	491299
4	4	Maruti	Swift Dzire	2013	VDI BS IV Manual	New Delhi	Diesel	1st Owner	40212	370699

```python
df.drop('Unnamed: 0', axis=1, inplace=True)
```

```python
df.shape
```

```
(183, 9)
```

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 183 entries, 0 to 182
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Brand             183 non-null    object
 1   Model             183 non-null    object
 2   Year              183 non-null    int64 
 3   Variant           183 non-null    object
 4   Location          183 non-null    object
 5   Version           183 non-null    object
 6   Number of Owners  183 non-null    object
 7   KmDriven          183 non-null    int64 
 8   Price             183 non-null    int64 
dtypes: int64(3), object(6)
memory usage: 13.0+ KB
```

```python
df.isnull().sum()
```

```
Brand               0
Model               0
Year                0
Variant             0
Location            0
Version             0
Number of Owners    0
KmDriven            0
Price               0
dtype: int64
```

Data Inputs- Logic- Output Relationships

EDA was performed by creating valuable insights using various visualization libraries.

Hardware and Software Requirements and Tools Used

Hardware required:
Processor: core i3
RAM: 8 GB

Software required:
Anaconda 3- language used Python 3
Microsoft Excel

STATISTICAL SUMMARY

To see statistical information about the non-numerical columns in our dataset:

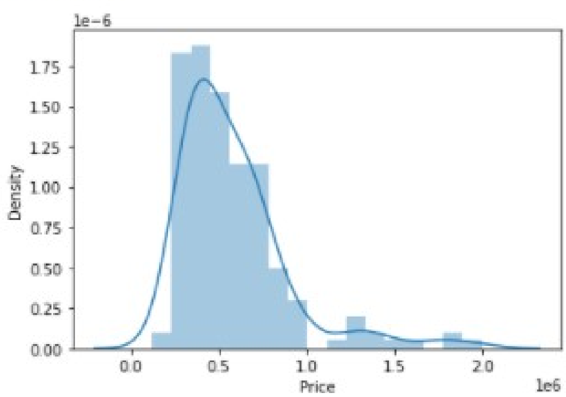
```
df.describe()
```

	Brand	Model	Year	Variant	Location	Version	Number of Owners	KmDriven	Price
count	183.000000	183.000000	183.000000	183.000000	183.000000	183.000000	183.000000	183.000000	1.830000e+02
mean	2.448087	16.273224	7.901639	37.218579	3.803279	0.584699	0.147541	55231.972678	5.774444e+05
std	1.061972	8.786334	2.612554	18.434210	1.956839	0.505123	0.399288	45490.437051	3.100881e+05
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2400.000000	1.203990e+05
25%	2.000000	8.000000	6.000000	25.500000	2.000000	0.000000	0.000000	25650.500000	3.648990e+05
50%	2.000000	17.000000	8.000000	39.000000	4.000000	1.000000	0.000000	44638.000000	5.045990e+05
75%	3.000000	24.000000	10.000000	51.000000	5.000000	1.000000	0.000000	64361.000000	6.846990e+05
max	5.000000	33.000000	13.000000	73.000000	7.000000	2.000000	2.000000	353288.000000	1.997999e+06

Exploratory Data Analysis

Let us explore our data and visualize

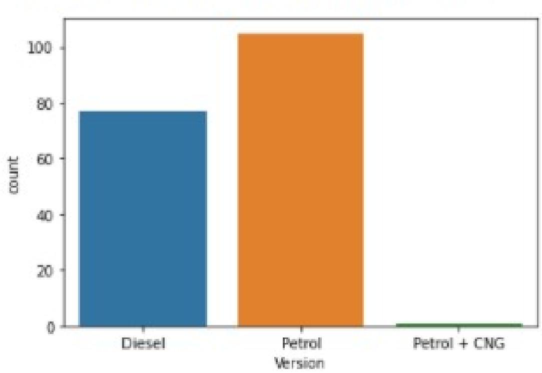
```python
#checking the Target column
sns.distplot(df['Price'])
plt.show()
```



As from above plot we see data is not normally distributed

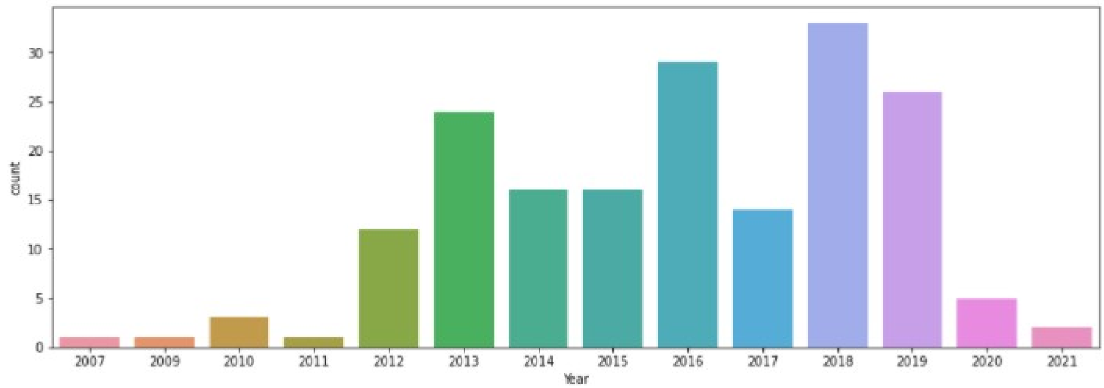
```python
sns.countplot(x = 'Version', data = df)
```

```
<AxesSubplot:xlabel='Version', ylabel='count'>
```



```python
plt.figure(figsize=(15,5))
sns.countplot(x = 'Year', data = df)
```

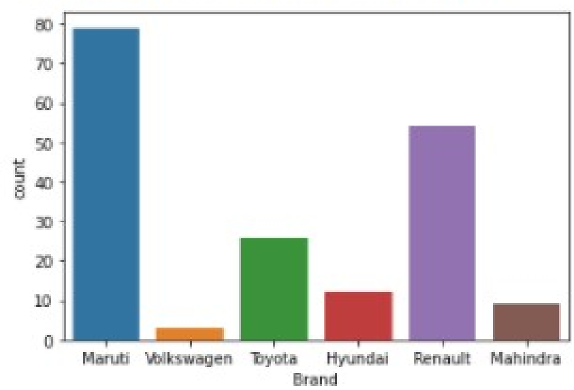
```
<AxesSubplot:xlabel='Year', ylabel='count'>
```



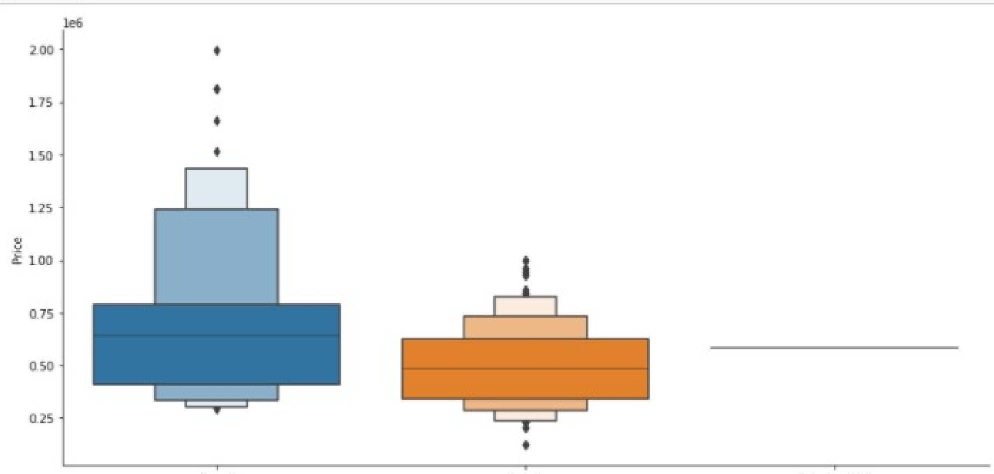
From above plot we see in between 2012 to 2019 ,count of cars are more as compared to rest of the years.

```python
sns.countplot(x = 'Brand', data = df)
```

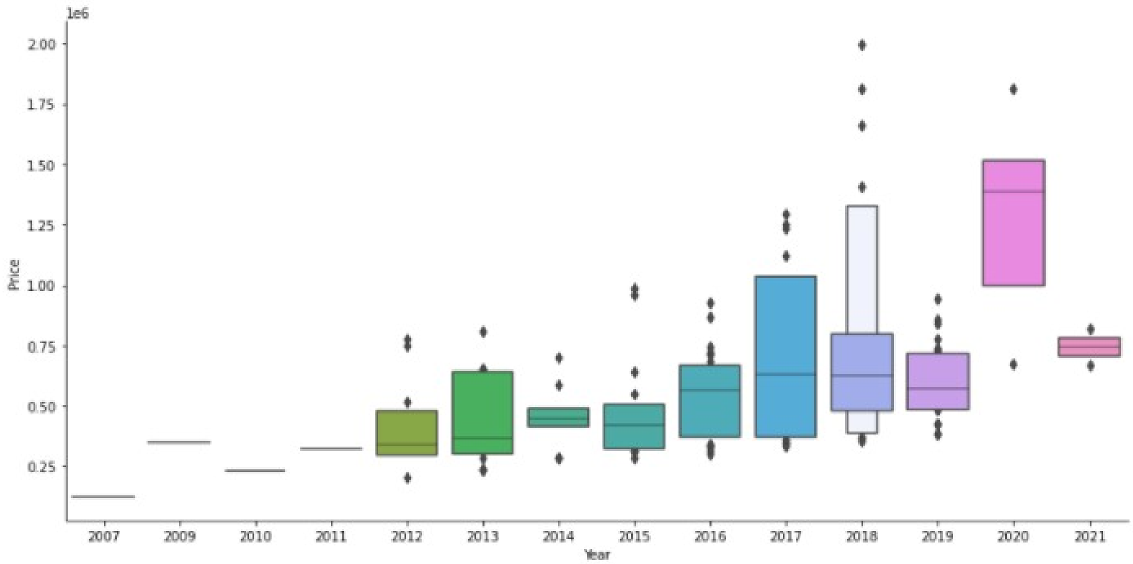
```
<AxesSubplot:xlabel='Brand', ylabel='count'>
```




```python
# Relation between price nad fuel

sns.catplot(y = 'Price', x = 'Version',data= df.sort_values("Price", ascending = False),
           kind = "boxen", height = 6, aspect = 2)

plt.tight_layout
plt.show()
```



```
# Relation between Price and Manufacturing year

sns.catplot(y ='Price', x ='Year',data=df.sort_values("Price",ascending=False),kind="boxen",height=6,aspect=2)
plt.tight_layout
plt.show()
```



Correlation matrix

A correlation matrix is simply a table which displays the correlation. The measure is best used in variables that demonstrate a linear relationship between each other. The fit of the data can be visually represented in a heatmap.

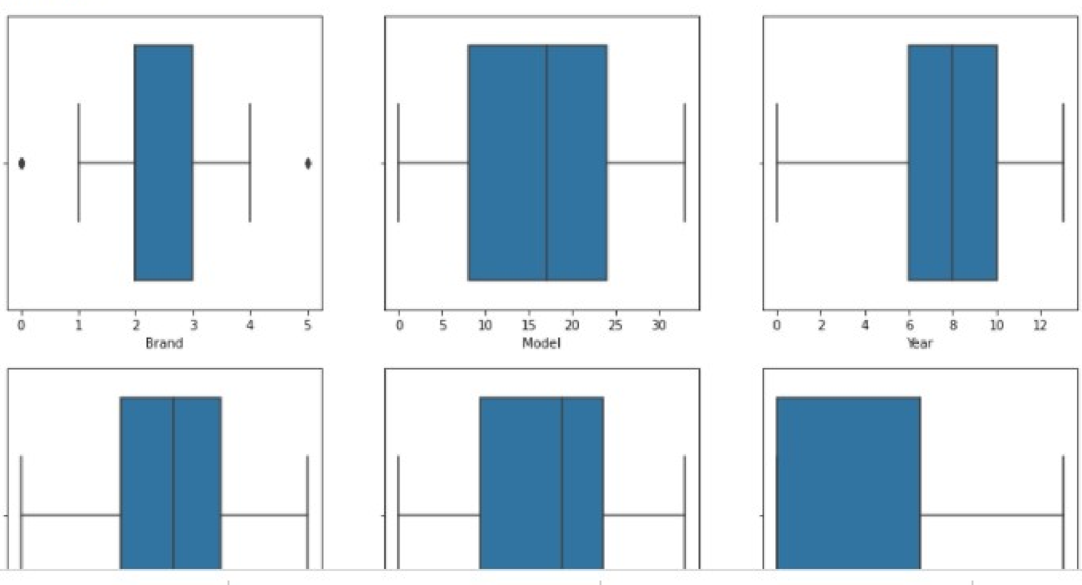
```
df.corr()
```

	Brand	Model	Year	Variant	Location	Version	Number of Owners	KmDriven	Price
Brand	1.000000	-0.118009	0.083306	-0.387299	0.135190	0.164444	0.037599	0.115463	0.105054
Model	-0.118009	1.000000	-0.129036	0.027718	-0.156642	-0.230561	0.047960	0.081666	0.026788
Year	0.083306	-0.129036	1.000000	-0.074050	-0.142449	0.293634	-0.175630	-0.610420	0.466907
Variant	-0.387299	0.027718	-0.074050	1.000000	-0.134668	-0.110573	-0.161166	-0.144811	0.029458
Location	0.135190	-0.156642	-0.142449	-0.134668	1.000000	-0.016403	-0.096260	0.089553	-0.229075
Version	0.164444	-0.230561	0.293634	-0.110573	-0.016403	1.000000	-0.103164	-0.460520	-0.299948
Number of Owners	0.037599	0.047960	-0.175630	-0.161166	-0.096260	-0.103164	1.000000	0.077605	-0.103484
KmDriven	0.115463	0.081666	-0.610420	-0.144811	0.089553	-0.460520	0.077605	1.000000	-0.109102
Price	0.105054	0.026788	0.466907	0.029458	-0.229075	-0.299948	-0.103484	-0.109102	1.000000

```
#Let's check the correlation by using the Heatmap (in order to check teh relation between features)

plt.figure(figsize=(15,8))
sns.heatmap(df.corr(),cmap='YlGnBu',annot = True, linewidth=0.5, fmt='.2f')
plt.show()
```

Price is correlated with Year

```
#visualizing data for outliers
plt.figure(figsize=(15,50))
graph=1
for column in df:
    if graph<=30:
        ax=plt.subplot(10,3,graph)
        sns.boxplot(df[column],orient='v')
        plt.xlabel(column,fontsize=10)

    graph+=1
plt.show()
```



```
from scipy.stats import zscore
z=np.abs(zscore(df))
z.shape
```

```
(183, 9)
```

```
threshold=3
print(np.where(z>3))
```

```
(array([ 66,  75, 119, 120, 128, 133, 155, 161, 163, 163], dtype=int64), array([6, 6, 8, 7, 8, 8, 8, 8, 2, 6], dtype=int64))
```

```
df_new=df[(z<3).all(axis=1)]
print('Old DataFrame',df.shape)
print('New DataFrame',df_new.shape)
print('total_dropped_rows',df.shape[0]-df_new.shape[0])
```

```
Old DataFrame (183, 9)
New DataFrame (174, 9)
total_dropped_rows 9
```

```
loss_percentage=(183-174)/183*100
print(loss_percentage,'%')
```

```
4.918032786885246 %
```

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

1. Used heatmap to visualize it and check the correlation among the data.
2. Get a clear view of the columns visually, we have used distribution plots.
3. For checking outliers, we have used boxplots.
4. For scaling the data, we have used Standard Scaler method.
5.For training and testing the data, we have imported train_test_split library from scikit-learn.
6.For model building, we have used different regressor models out of which AdaBoost Regressor model is better model for dataset and then we done hyperparameter tuning (RandomizedSearchCV).

```
#Hyperparameter tuning using RandomizedSearchCV
from sklearn.model_selection import RandomizedSearchCV
#List of parameters to pass
n_estimators = [10,50,100]
loss=['linear','square','exponential']
#max_features = ['auto', 'sqrt']
#max_depth = [2, 3, 5]
#min_samples_split = [2, 4, 6]
#min_samples_leaf = [1, 2, 4, 6]
learning_rate=[0.1]
```

```
#Creating random grid
ab=AdaBoostRegressor()
random_grid = {'n_estimators': n_estimators,
               'loss':loss,
               'learning_rate':learning_rate}
```

```
ng 5 fold cross validation,
inations
imator = ab, param_distributions = random_grid,scoring='neg_mean_squared_error', n_ite
```

```
ab.fit(X_train,y_train)
y_pred=ab.predict(X_test)
```

```
ab_random.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
[CV] END ....learning_rate=0.1, loss=linear, n_estimators=10; total time=   0.1s
[CV] END ....learning_rate=0.1, loss=linear, n_estimators=10; total time=   0.0s
[CV] END ....learning_rate=0.1, loss=linear, n_estimators=10; total time=   0.0s
[CV] END ....learning_rate=0.1, loss=linear, n_estimators=10; total time=   0.0s
[CV] END ....learning_rate=0.1, loss=linear, n_estimators=10; total time=   0.0s
[CV] END ....learning_rate=0.1, loss=linear, n_estimators=50; total time=   0.0s
```

Interpretation of the Results

In the visualization part, I have seen how my data looks like using heatmap, boxplot, distribution plots, histogram etc. In the pre-processing part, we have cleaned my data using many methods like LabelEncoder etc. In the modelling part, we have designed our model using algorithm like AdaBoostRegressor.

CONCLUSION

Key Findings and Conclusions of the Study

The key findings are we have to study the data very clearly so that we are able to decide which data are relevant for our findings. The techniques that we have used are heatmap, Label Encoder etc. The conclusion of our study is we have to achieve a model with good accuracy and f1-score.

Learning Outcomes of the Study in respect of Data

Science We will develop relevant programming abilities. We will demonstrate proficiency with statistical analysis of data. We will develop the ability to build and assess data-based models. We will execute statistical analysis with professional statistical software. The best algorithm for this project according to my work is AdaBoost Regressor because the accuracy that I have achieved is quite satisfactory than the other model.

Limitations of this work and Scope for Future Work

The scope for future work is to collect as many data as we can so that the model can be built more efficiently.