FLIP ROBO

CAR PRICE PREDICTION PROJECT

Submitted by:

Anita Thapa

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my SME Swati Ma'am for giving me an opportunity to work on the Car price prediction project. During my research on this project internet sites like towardsdatascience, medium, cardekho,cars24, and Stack Overflow helped me understand the main problem statement, also to collect data and in finding their solutions.

INTRODUCTION

- Business Problem Framing

  Model to predict used cars price valuation to help the sellers of used cars to understand the present trend in the car market so as to cope up with the losses faced by covid-19 impact.

- Conceptual Background of the Domain Problem

  The Automobile industry was badly impacted by covid-19 so alot of changes happened in the car market as the car valuation kept on changing. Now the cars which are more in demand are costlier in comparison with cars in lower demand. So to understand the present trend in the car market data is being collected from various online cars selling websites to model a car price valuation for in depth analysis.

- Review of Literature

  The research was done on Car prices in the various online cars selling websites like cardekho, cars24 etc in order to collect the data using the selenium library. Using websites like medium, towardsdatascience and Kaggle helped in modeling the dataset and in predicting the car valuation. Stack overflow website helped to resolve problems faced during data scraping and modeling.

- Motivation for the Problem Undertaken

  The main objective was to analyse the present car market changes and model a car price valuation prediction. In order to understand the automobile industry and their market trends this project was taken up so that a new car price valuation model is build to help sellers to understand the customer’s choices in the market.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

The data was present in excel format so using pandas the excel file was read. Then the dataset details were analysed using .shape() to find the number of rows and columns, using dtypes() we found the dataset datatype like object and numeric datatype. Dataset has 7085 Rows and 9 Columns, 4 columns are numeric and rest 5 columns are object type and the Target column is Price. Dataset is a Regression model.

```
#Importing Libraries:
import numpy as np
import pandas as pd
```

```
#Reading excel file and converting it in dataframe
ds= pd.read_excel("C:\\Users\\hp\\Desktop\\Cars_Data.xlsx")
df=pd.DataFrame(ds)
df.head()
```

	Unnamed:0	Model Year	Brand	Car Name	Variant	Distance Travelled	Number of Owners	fuel type	Price
0	0	2012	Maruti	Swift Dzire	Manual	118117	1st	Diesel	316399
1	1	2016	Renault	Kwid	Manual	46028	2nd	Petrol	277599
2	2	2013	Maruti	Swift	Manual	114506	1st	Diesel	341599
3	3	2014	Maruti	Ritz	Manual	43382	1st	Diesel	344199
4	4	2013	Hyundai	i20	Manual	64361	1st	Diesel	355799

Dataset in Dataframe format. Regression Model

```
# Rows & Columns in dataset:
df.shape
(7085, 9)
```

Dataset has 7085 Rows and 9 Columns

```
# Datatype of dataset
df.dtypes
Unnamed:0                int64
Model Year               int64
Brand                   object
Car Name                object
Variant                 object
Distance Travelled       int64
Number of Owners        object
fuel type               object
Price                    int64
dtype: object
```

There are 5 Object datatype and 4 numeric datatype.

- Data Sources and their formats

Data was scraped from various websites like cardekho and car24 using Selenium library in Python and saved as excel file. Then data was imported in python using pandas . Information of dataset is found using datasetname.info(). As per the information each column has count of 7085 but Variant columns have nan values and the datatypes are int64 and object type as shown in below figure.

```
# Information about data

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7085 entries, 0 to 7084
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Unnamed:0          7085 non-null   int64 
 1   Model Year         7085 non-null   int64 
 2   Brand              7085 non-null   object
 3   Car Name           7085 non-null   object
 4   Variant            6861 non-null   object
 5   Distance Travelled 7085 non-null   int64 
 6   Number of Owners   7085 non-null   object
 7   fuel type          7085 non-null   object
 8   Price              7085 non-null   int64 
dtypes: int64(4), object(5)
memory usage: 498.3+ KB
```

- Data Pre-processing Done

Dataset had 9 columns and 7084 rows. In Variant column nan values were present which was filled by mode vale of the variant column. Then the value in Variant column were changed to int64 by replacing Manual and Automatic by 0 and 1.

Then Number of Owners column data was also replaced by numeric values 1, 2,3 and 4.

In Fuel type column Petrol + LPG and Petrol + CNG was assumed as Petrol as count value of Petrol was more than CNG and LPG.

Data Pre Processing:

```
# Replacing values of Number of Owners

df['Number of Owners'] = df['Number of Owners'].replace({'1st':'1', '2nd':'2', '3rd':'3', '4th':'4' })
df['Number of Owners'].value_counts()
```

```
1    5296
2    1706
3      76
4       7
Name: Number of Owners, dtype: int64
```

```
# Number of Owners column datatype changed from object to int64

df['Number of Owners']=df['Number of Owners'].astype('int64')
```

```
# Replacing values of variant column
df['variant'] = df['variant'].replace({'manual':'Manual', 'automatic':'Automatic'})
df['variant'].value_counts()
```

```
Manual       5789
Automatic    1072
Name: variant, dtype: int64
```

```
# finding mode of variant column
df['variant'].mode()
```

```
0    Manual
dtype: object
```

```
# Filling nan value in dataset with mode
df['variant'].fillna(df['variant'].mode()[0],inplace=True)
```

```
# variant column values replaced
df['variant'] = df['variant'].replace({'Manual':0, 'Automatic':1})
df['variant'].value_counts()
```

```
0    6013
1    1072
Name: variant, dtype: int64
```

```
#checking values of fuel type column
df['fuel type'].unique()
```

```
array(['Diesel', 'Petrol', 'Petrol + CNG', 'Petrol + LPG', 'CNG', 'LPG',
       'Electric(Battery)'], dtype=object)
```

```
# fuel type column value replaced
df['fuel type'] = df['fuel type'].replace({'Petrol + CNG':'Petrol', 'Petrol + LPG':'Petrol', 'Electric(Battery)':'Electric'})
df['fuel type'].value_counts()
```

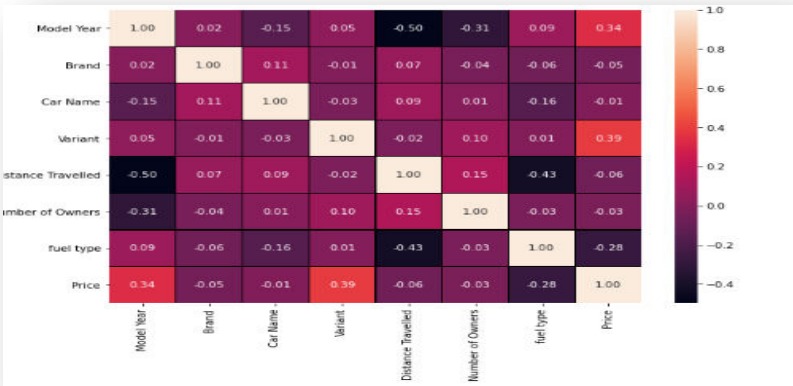
```
Petrol      4728
Diesel      2320
CNG           33
LPG            3
Electric       1
Name: fuel type, dtype: int64
```

```
# fuel type column value replaced to numeric value
df['fuel type'] = df['fuel type'].replace({'Diesel':0, 'Petrol':1, 'CNG':2, 'LPG':3, 'Electric':4})
df['fuel type'].value_counts()
```

```
1    4728
0    2320
2      33
3       3
4       1
Name: fuel type, dtype: int64
```

- Data Inputs- Logic- Output Relationships

  Output Column is Price column and rest other columns are Input Columns. Using Correlation we can find out the relationship between Input and output columns. Variant column is positively correlated with output column.



- State the set of assumptions (if any) related to the problem under consideration

  In Fuel type column Petrol + LPG and Petrol + CNG is assumed as Petrol as count value of Petrol was more than CNG and LPG.

- Hardware and Software Requirements and Tools Used

  Libraries used were:

  1. Numpy : It is a Numerical Python library used for numerical computation like arrays in Python.
  2. Panda: It was used for reading and converting excel/csv file into dataframe.
  3. matplotlib: this library was used for plotting the graph in the EDA.
  4. Seaborn: This library is also used for visualization as it has helped to plot different types of plots like countplot, displot, boxplot and heatmap in the notebook.
  5. Label encoder: It helped to encode the object datatype into numeric datatype as the Machine learning algorithm works on numeric datatype only.
  6. sklearn.preprocessing and power transform: It was used to remove the skewness from the dataset.
  7. Standard Scaler : It was used to scale the feature column of the dataset before model selection.
  8. sklearn: This library was used to import train_test_split, metrics like accuracy score, classification report and also importing Algorithms likes Linear Regression, Decision Tree Regressor, SVR, Random Forest Regressor and K-Neighbors Regressor..
  9. pickle: This library was used to save the final model.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

The Standard Scaled data was used in Model building. As the model is Regressor so Linear Regressor library along with r2 score, cross validation score, means square error metrics were imported. Then 4 Algorithms were used which were Decision Tree Regressor, SVR, Random Forest Regressor and K-Neighbors Regressor. Then after selecting best model, then the final model was deployed and the accuracy of model was 85.4% as it predicted values approximately equal to actual values.

Model Building

```
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

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

```
for i in range(0,100):
    train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.2,random_state=i)
    lr.fit(train_x,train_y)
    pred_train=lr.predict(train_x)
    pred_test=lr.predict(test_x)
    print(f"At random state {i},the training accuracy is:- {r2_score(train_y,pred_train)}")
    print(f"At random state {i},the testing accuracy is:- {r2_score(test_y,pred_test)}")
    print("\n")
```

- Testing of Identified Approaches (Algorithms)

Algorithms used for the training and testing are:

1. Linear Regression
2. Decision Tree Regressor
3. SVR
4. Random Forest Regressor
5. K-Neighbors Regressor

- Run and Evaluate selected models

Out of all the 5 algorithm used the best algorithm used is Decision Tree Classifier as the accuracy score and cross validation score.

3. KNeighborsRegressor

```
# Importing Libraries and Hyper parameter tuning:
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error,mean_absolute_error

parameters = {'n_neighbors':list(range(0,10)),
              'weights':['uniform', 'distance'],
              'algorithm':['auto', 'ball_tree', 'kd_tree', 'brute']
              }
kn =KNeighborsRegressor()
clf = GridSearchCV(kn,parameters)
clf.fit(train_x,train_y)

print(clf.best_params_)
```

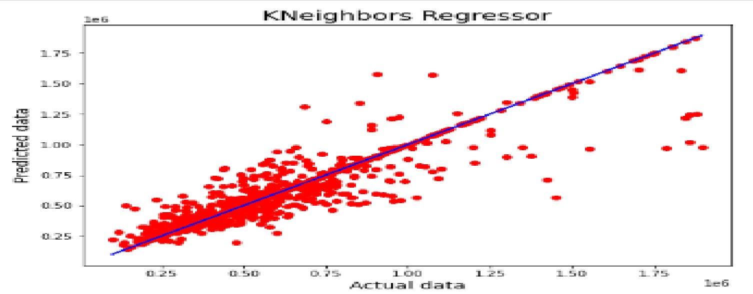
```
{'algorithm': 'auto', 'n_neighbors': 9, 'weights': 'distance'}
```

```
kn =KNeighborsRegressor(n_neighbors=9,algorithm="auto",weights='distance')
kn.fit(train_x,train_y)
kn.score(train_x,train_y)
predkn = kn.predict(test_x)
print('kn score',kn.score(train_x,train_y))
kns = r2_score(test_y,predkn)
print('R2 Score:',kns*100)

knscore = cross_val_score(kn,x,y,cv=7)
knc = knscore.mean()
print('Cross Val Score:',knc*100)
print("Mean squared error:",mean_squared_error(test_y,predkn))
print("Mean absolute error:",mean_absolute_error(test_y,predkn))
```

```
kn score 0.9999827980029782
R2 Score: 85.4984830547524
Cross Val Score: 84.93799993173525
Mean squared error: 13047800805.2547
Mean absolute error: 48619.812441345406
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,6))
plt.scatter(x=test_y, y=predkn, color='r')
plt.plot(test_y,test_y, color='b')
plt.xlabel('Actual data',fontsize=14)
plt.ylabel('Predicted data',fontsize=14)
plt.title('KNeighbors Regressor',fontsize=18)
plt.show()
```

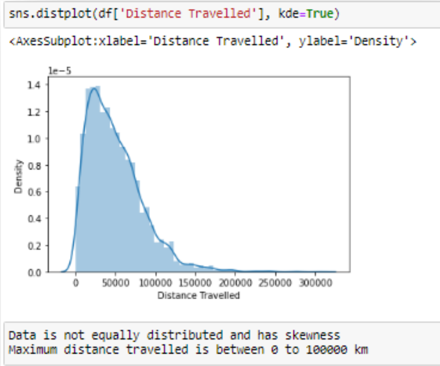


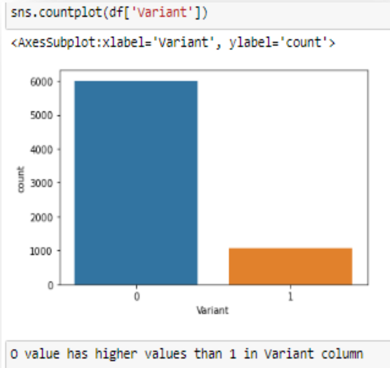
- Key Metrics for success in solving problem under consideration

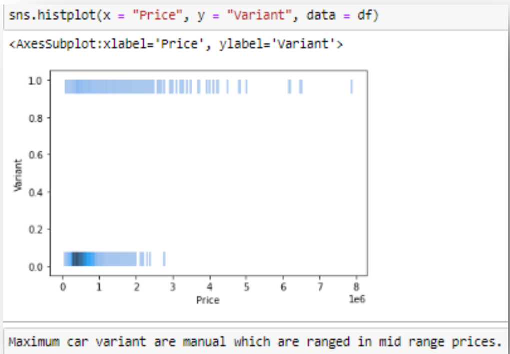
  The main metric used was to find out the accuracy of the model which helped to determine the best model is comparing r2 score with the cross validation score of the algorithm. For model selection the algorithm must have small difference between cross validation score and r2 score then that algorithm is considered to be the best model.

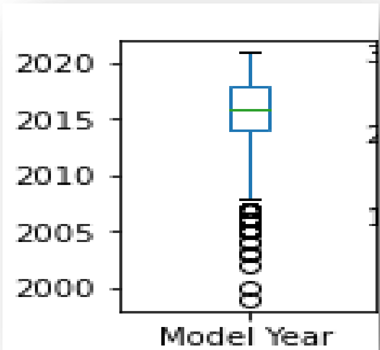
- Visualizations

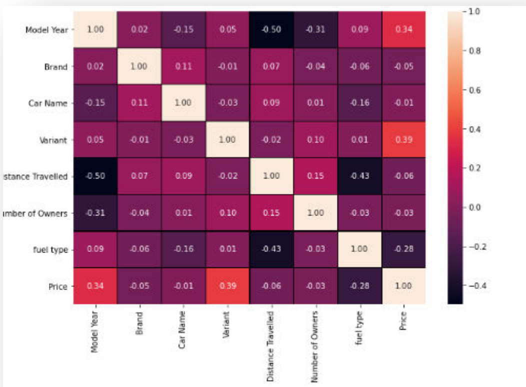
  Different plots were used in the problem like displot, countplot, histplot, heatmap, boxplot and pyplot using matplotlib and seaborn library as shown in the images below.

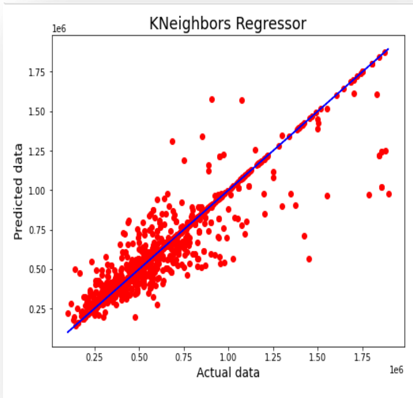












- Interpretation of the Results

  From the visualizations, preprocessing and modeling of the data it was interpreted that maximum cars added in the websites online have manufacturing year between 2017 and 2018 and have petrol fuel type. Maximum cars listed are mid ranged as they must be in demand in the car market. As per the model the data can be predicted with 85.4% accuracy.

CONCLUSION

- Key Findings and Conclusions of the Study

  From the project it is inferred that maximum cars having manufacturing year 2017 and 2018 are in great demand.

- Learning Outcomes of the Study in respect of Data Science

  Using Visualization libraries like matplotlib and seaborn it was easy to find the correlation between the dataset columns and the plotting of each column along the target column helped to analyse the problem pretty well. As visualization helped to get better insight of the data like skewness present in the data.

- Limitations of this work and Scope for Future Work

  As the model accuracy is 85.4% so it can approximately predict the correct value as sometimes it might predict wrong values but chances are less.