CAR PRICE PREDICTION

Submitted By:

Dumpala Aditya

ACKNOWLEDGMENT

Data source:- cars24 and olx

Refences :- geeksforgeeks, google, analyticalvidya

INTRODUCTION

- **Problem statement:-**
  Collect the used cars data from various sites and built the model for predicting the car price based on collected data
- **Conceptual Background of the Domain Problem:-**
  - **Problem type:-** Regression
  - **Target variable:-** price
  - **Important features:-**
    1)year of manufacture:- year of vehicle manufactured

    2)kilometers driven:- no of kilometers vehicle is driven

    3)variant of the model:- variant of the model

    4)fuel type:- type of fuel used

5)no of owners:- no of owners for the vehicle

6)transmission:- transmission of the vehicle

- **Motivation for the Problem Undertaken:-**
  due to corona pandemic 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. due to corona pandemic, customers are facing problems with their previous prices so to overcome this problem I am taking data of used cars data present in various sites this will helps us to reducing difference varying price before pandemic and after pandemic

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

- **Mathematical / analytical modelling of the problem:**
  This is regression problem so our target is in the form of continuous and other features of the data set are some are continuous and some are categorical,

  To understand statistical information about the features or data set iam used data.describe() it tells that mean median and standard deviation of the data set and also it tells that min and max value of the each feature

- **Data Sources and their formats:**
  To understand information of the data set we can use data.info()
  - Variant:- is one of the important feature in the data set it has so many categories so I am used get dummies to encode this feature
  - Year:- is another important feature this tells that manufacture of the vehicle year it is in the form of int so I am keep it as it is
  - Price:- is our target variable but is in the form of object type iam converted it into numerical
  - Owner:- is another feature it is also a categorical feature

- Transmission :- it tells that transmission of the vehicle
- Fuel type :- type of fuel used for the vehicle

- **Data Preprocessing Done:**
  in the data set we don't have data with proper manner we have to set it in the proper format so let's understand the each feature and the techniques to convert it , techniques used are

  - Get dummies used for encoding categorical features
  - Plots using in this anlysis are distribution plot and scatter plot , box plot ,countplot
  - Log and square root transformation for removing skewness of the continuous features
  - Boxplot for identifying the outliers in the features
  - For scaling the data iam using minmax scaler
  - To check multicollinearity iam using vif method(variance inflation factor)

- Data Inputs- Logic- Output Relationships:-
  Some of the features are high correlation with the target variable

  <u>Features vs price:</u>

  1)We can take year if the vehicle is recent one the price is high it means that less tenure of the vehicle has high price

  2)owner of the vehicle is more price is slightly decreases

  3) kilometers driven increases price is decreases

  4) transmission automatic has high price compared to manual

- State the set of assumptions (if any) related to the problem under consideration
  To remove outliers I can assume that in the price feature it is possible that price is high
  Min price is 60000 and max price is 57lakhs it is possible

But in the feature kilometers driven there are highest kilometers driven is 999999 (10lakhs) it does not make sence so, the life time of the vehicle is 3.5lakh kilometers so iam removing data which has more than 3.5 lakh kilometers

- Hardware and Software Requirements and Tools Used

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error,mean_squared_error
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Lasso
```

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  - Our out put variable is continuous in nature so we can use regression model,

- Testing of Identified Approaches (Algorithms)

Algorithms I am used for train and test

```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor
```

- Run and Evaluate selected models
  - **Linear Regression:-**

```
#linear regression
lm=LinearRegression()
lm.fit(x_train,y_train)

print_score(lm,x_train,x_test,y_train,y_test,train=True)
print_score(lm,x_train,x_test,y_train,y_test,train=False)
kfolds(lm,'linearregression')
```

```
=======train results===========
accuracy score:61.965786%


=======test results=========
r2score is:62.147984%


linearregression score on cross validation: -7.0748786537874086e+25%
```

Linear regression is used to predict the relationship between two variables by applying a linear equation to observe the data, there two variables one is dependent another one is independent where dependent variable is called target variable and independent variable is called features

Equation used in this model is Y=mx+c

Where 'Y' is dependent variable and it is plotted along with y-axis

Where 'm' is the slop

Where 'c' is interceptor

Where x is independent variable and it is plotted along with the x-axis

  - **Random Forest regressor:-**

```
#random Forest
from sklearn.ensemble import RandomForestRegressor
rfc=RandomForestRegressor()
rfc.fit(x_train,y_train)

print_score(rfc,x_train,x_test,y_train,y_test,train=True)
print_score(rfc,x_train,x_test,y_train,y_test,train=False)

kfolds(rfc,'RandomForestRegressor')
```

```
=======train results===========
accuracy score:94.138310%


=======test results=========
r2score is:63.720410%


RandomForestRegressor score on cross validation: 58.620714257853216%
```

Random forest is an ensemble technique, and ensemble technique is nothing but take multiple algorithms or same algorithm multiple times and put together their each results if the problem is classification type it uses voting system i.e it gives category of majority votes or if the problem is regression type it takes average of all the models this is called ensemble technique and also this combination of the model is more powerful than original model

- Gradient boosting regressor

```
#GradientBoostingRegressor
from sklearn.ensemble import GradientBoostingRegressor
gbr=GradientBoostingRegressor()
gbr.fit(x_train,y_train)

print_score(gbr,x_train,x_test,y_train,y_test,train=True)
print_score(gbr,x_train,x_test,y_train,y_test,train=False)

kfolds(gbr,'GradientBoostingRegressor')
```

```
=======train results===========
accuracy score:77.399999%


=======test results=========
r2score is:65.144244%


GradientBoostingRegressor score on cross validation: 57.5534699546854%
```

Gradient boosting is the model to built models sequentially and these subsequent models try to reduce the errors of previous model,

- Bagging regressor:-

```
#BaggingRegressor
from sklearn.ensemble import BaggingRegressor
bgr=BaggingRegressor()
bgr.fit(x_train,y_train)

print_score(bgr,x_train,x_test,y_train,y_test,train=True)
print_score(bgr,x_train,x_test,y_train,y_test,train=False)

kfolds(bgr,'BaggingRegressor')
```

```
=======train results===========
accuracy score:92.167700%


=======test results=========
r2score is:58.831465%


BaggingRegressor score on cross validation: 56.51966679062463%
```

Bagging is one of the ensemble technique and it is useful for both regression and classification problem, this is used with decision trees where it significantly raises the stability of the models and improving accuracy and reducing the variance this overcomes the challenge of overfitting problem

- Decision tree regressor

```
#DecisionTreeRegressor
from sklearn.tree import DecisionTreeRegressor
dtr=DecisionTreeRegressor()
dtr.fit(x_train,y_train)

print_score(dtr,x_train,x_test,y_train,y_test,train=True)
print_score(dtr,x_train,x_test,y_train,y_test,train=False)

kfolds(dtr,'DecisionTreeRegressor')
```

```
=======train results===========
accuracy score:100.000000%


=======test results=========
r2score is:17.326484%


DecisionTreeRegressor score on cross validation: 12.051655194123503%
```

Decision tree regression enables one to divide the data into multiple splits. These splits typically answer a simple if-else condition. The algorithm decides the optimal number of splits in the data. Since this method of splitting data closely resembles the branches of a tree, this is probably is known as a decision tree.

- XGRegressor:-

```
from xgboost import XGBRegressor
xgbr=XGBRegressor()

xgbr.fit(x_train,y_train)

print_score(xgbr,x_train,x_test,y_train,y_test,train=True)
print_score(xgbr,x_train,x_test,y_train,y_test,train=False)
```

```
=======train results===========
accuracy score:98.950612%


=======test results=========
r2score is:60.547745%
```

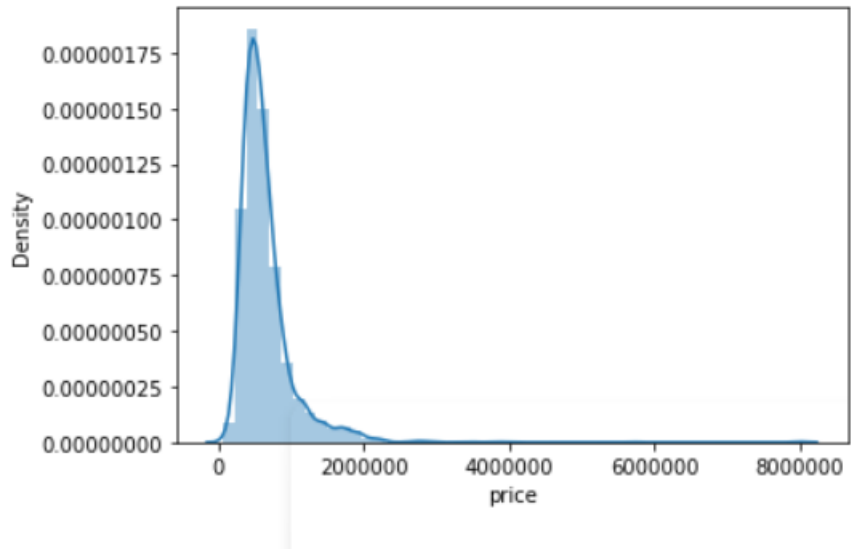
- Key Metrics for success in solving problem under consideration
  - Choosing PCA rather than vif :-

    after scaling the data iam using vif for checking multicollinearity after removing some of the features based on vif score r2score and cross validation score of the model is very low, then iam choosing PCA method actually it is not feature selection method but it reduce the dimensions of the data by using this the score of r2 and cross validation are increases more
  - Checking mse,mae and rmse for choosing the best model

- Visualizations

```
plt.ticklabel_format(style='plain')
sns.distplot(data['price'])
```

```
<AxesSubplot:xlabel='price', ylabel='Density'>
```

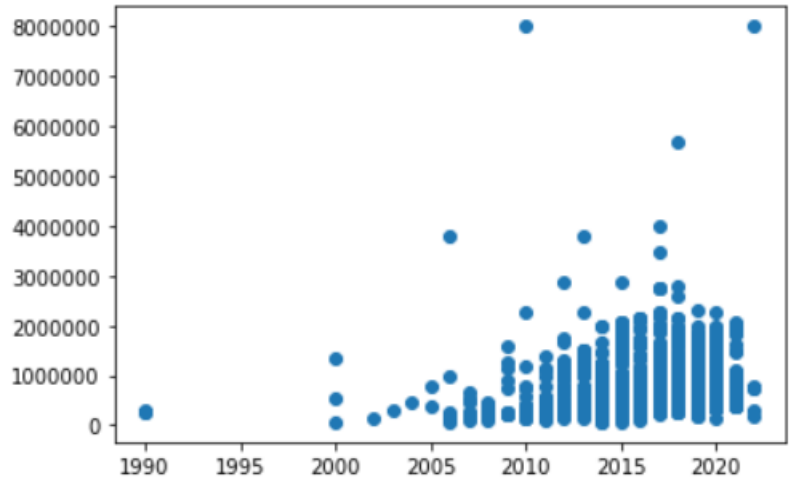


Observation:-

Price is skewed right side due to high range of the price value.

```
plt.ticklabel_format(style='plain')
plt.scatter(x='year',y='price',data=data)
```

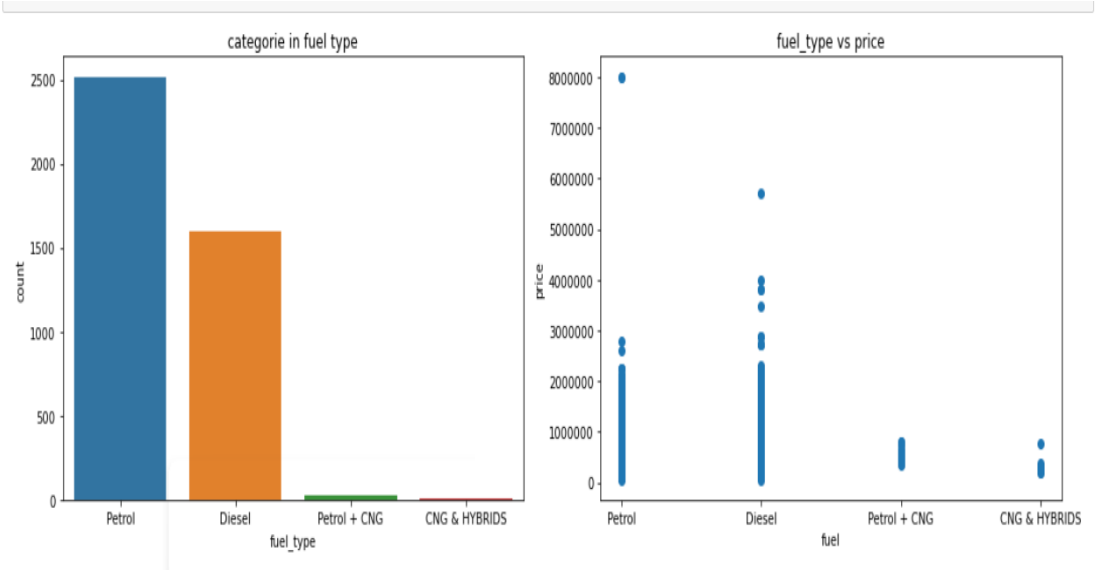
```
<matplotlib.collections.PathCollection at 0x14956a5b8e0>
```



Observation:-

Comparing two features price and year we can observe that recent year vehicle has high price comparing with old vehicle

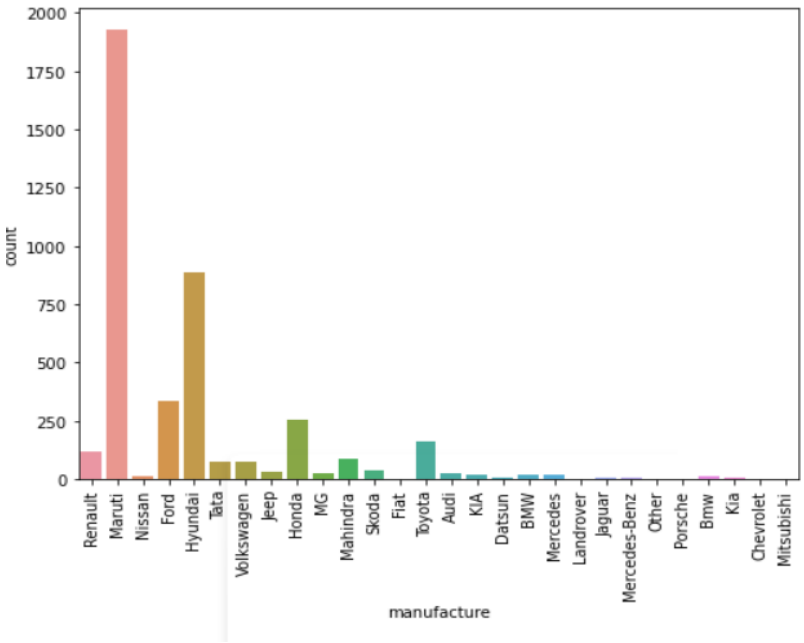
Fuel_type:-



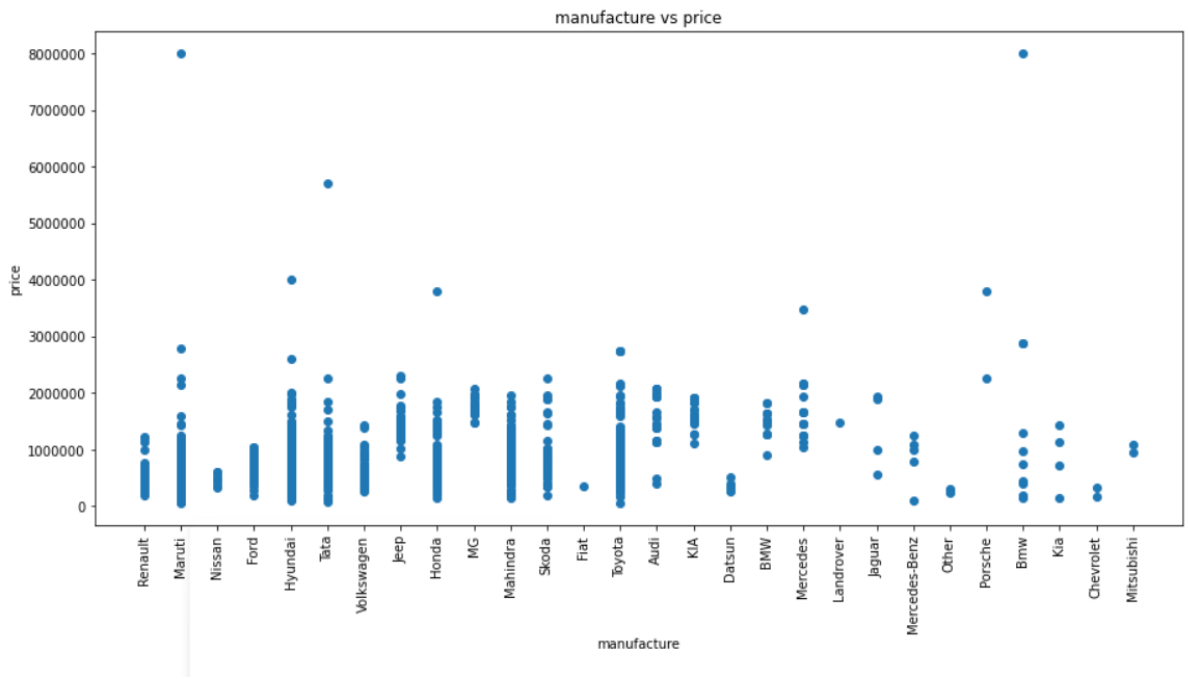
Observation:-

Vehicles with fuel type petrol are high count but vehicles with fuel_type diesel are having high price compared to petrol we can see

Manufacture:-



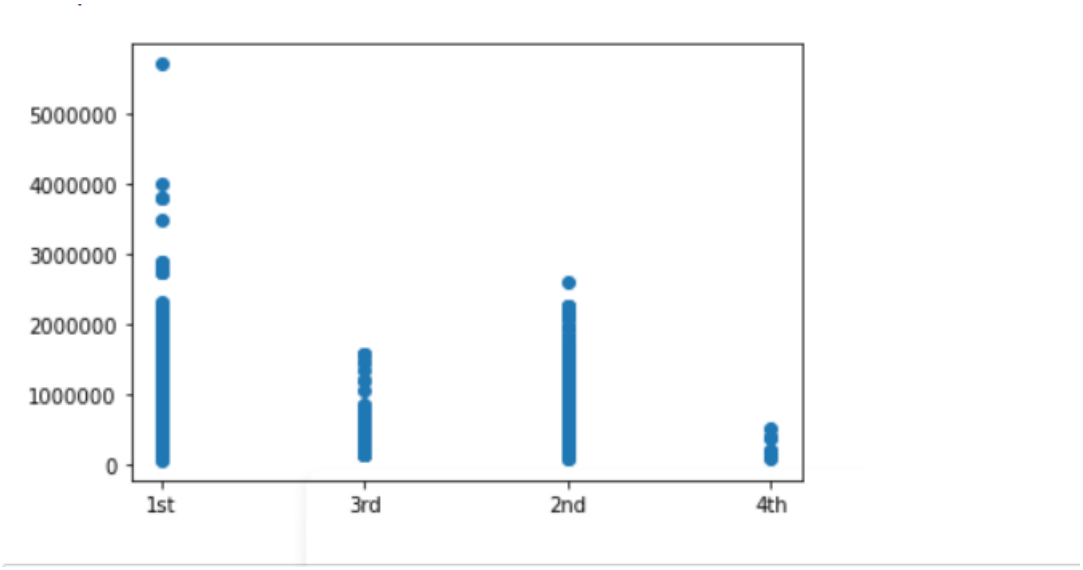
Manufacture vs price:-



Observation:-

There are many companies are manufacturing vahicles but we can observe that high price is 57lakhs and minimum is 60k but we can see that MG, Porsche, landrover, kia, bmw are has high minimum price because they are well known and branded cars offcourse other cars also branded but when features are more attractive in this branded cars

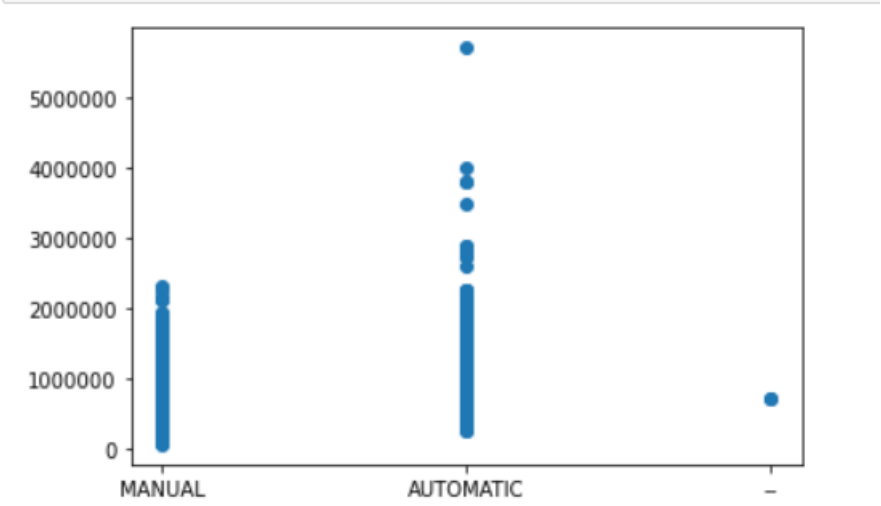
Owner:-



Observation:-

We can observe that owner 1st vehicle is high price this is natural and we observe that increasing no of owners the price of the vehicle is decreasing we can see, so our observation is vehicle with 1st owner has high price

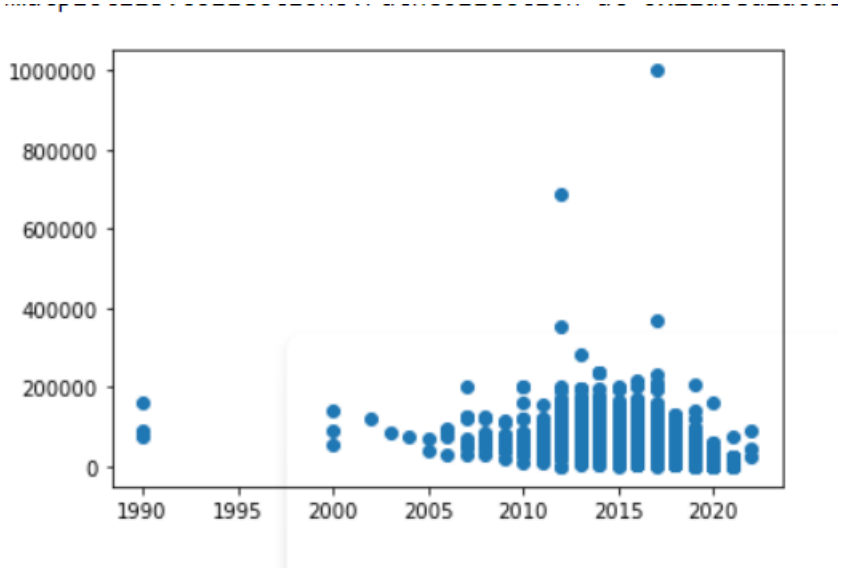
Transmission:-



There are two types of transmissions are there one is manual and another one is automatic we can see that vehicle with automatic transmission has high price compared to manual transmission vehicle

And there is another category '-' this is non usefull data we will remove it later

Year vs kilometers driven :-



Observation:-

We can see that most of the vehicles are between 2005 to 2020 almost all the vehicles are between range of 10km to 2.5 lakh km but we observe one thing is some of the data points are above 4lakh and 6lakh but the average life of car is 3.5 lakh km only so the data which is above 3lakh iam consider as outliers so I can remove those data points

CONCLUSION

- Key Findings and Conclusions of the Study
  - Rate is based on year and kilometers driven

- Learning Outcomes of the Study in respect of Data Science
  - Data in the data set is very clumsy and combination of various data from this iam learnt is first of all we have to understand the data present in the data set then we proceed with that data
  - Handling skewed data
  - Visualize the data by using seaborn and matplotlib
  - Encoding the data by using get_dummies
  - Scaling the data
  - Applying PCA for decreasing dimensionality
  - Applying different algorithms to the trian and test data
  - Checking error rate by using mae,mse,rmse
  - Tunnign the model with best parameters
  - Comparing actual and predicted values

- Limitations of this work and Scope for Future Work
  - data is not same in all the websites this is one of the challenge to collecting the data from various sites

END