PROJECT REPORT ON :

- "CAR PRICE PREDICTION"

SUBMITTED BY / AUTHOR :

- AMAN KUMAR PATEL

SME :

- MD. KASHIF

ACKNOWLADGEMENT

Contents:

1.INTRODUCTION

1.1 Business Problem Framing:

Car price prediction is somehow interesting and popular problem. As per information that was gotten from the Agency for Statistics of BiH, 921.456 vehicles were registered in 2014 from which 84% of them are cars for personal usage. This number is increased by 2.7% since 2013 and it is likely that this trend will continue, and the number of cars will increase in future. This adds additional significance to the problem of the car price prediction. Accurate car price prediction involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, age, horsepower and mileage. The fuel type used in the car as well as fuel consumption per mile highly affect price of a car due to a frequent changes in the price of a fuel. Different features like exterior color, door number, type of transmission, dimensions, safety, air condition, interior, whether it has navigation or not will also influence the car price. In this report, we applied different methods and techniques in order to achieve higher precision of the used car price prediction.

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.

1.2 Conceptual Background of the Domain Problem

The prices of new cars in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So, customers buying a new car can be assured of the money they invest to be worthy. But due to the increased price of new cars and the incapability of customers to buy new cars due to the lack of funds, used cars sales are on a global increase. There is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features. Even though there are websites that offers this service, their prediction method may not be the best. Besides, different models and systems may contribute on predicting power for a used car's actual market value. It is important to know their actual market value while both buying and selling.

There are lots of individuals who are interested in the used car market at some points in their life because they wanted to sell their car or buy a used car. In this process, it's a big corner to pay too much or sell less than it's market value.

There are one of the biggest target group that can be interested in results of this study. If used car sellers better understand what makes a car desirable, what are the important features for a used car, then they may consider this knowledge and offer a better service.

1.3 Review of Litrarture:

The second-hand car market has continued to expand even as the reduction in the market of new cars. According to the recent report on India's pre-owned car market by Indian Blue Book, nearly 4 million used cars were purchased and sold in 2018-19. The second-hand car market has created the business for both buyers and sellers. Most of the people prefer to buy the used cars because of the affordable price and they can resell that again after some years of usage which may get some profit. The price of used cars depends on many factors like fuel type, colour, model, mileage, transmission, engine, number of seats etc., The used cars price in the market will keep on changing. Thus the evaluation model to predict the price of the used cars is required.

1.4 Motivation for the Problem Undertaken

There are websites that offers an estimate value of a car. They may have a good prediction model. However, having a second model may help them to give a better prediction to their users. Therefore, the model developed in this study may help online web services that tells a used car's market value.

2.Analytical Problem Framing

2.1 Mathematical/ Analytical Modeling of the Problem

As a first step I have scrapped the required data from car dekho website. I have fetched data for different locations and saved it to excel format.

In this particular problem I have car_price as my target column and it was a continuous column. So clearly it is a regression problem and I have to use all regression algorithms while building the model. There was null values in the dataset. Also, I observed some unnecessary entries in some of the columns like in some columns I found more than 50% null values so I decided to drop those columns. If I keep those columns as it is, it will create high skewness in the model. Since we have scrapped the data from cardekho website the raw data was not in the format, so we have use feature engineering to extract the required feature format. To get better insight on the features I have used ploting like distribution plot, bar plot, reg plot, strip plot and count plot. With these ploting 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 z-score method and I removed skewness using yeo-johnson method. I have used all the regression algorithms while building model then tunned the best model and saved the best model. At last I have predicted the car-price using saved model.

2.2 Data Sources and their formats

The data was collected from cardekho.com website in excel format. The data was scrapped using selenium. After scrapping required features the dataset is saved as excel file.

Also, my dataset was having 12608 rows and 20 columns including target. In this perticular datasets I have object type of data which has been changed as per our analysis about the dataset. The information about features is as follows.

2.3 Data Preprocessing Done

- ✓ As a first step I have scrapped the required data using selenium from cardekho website.
- ✓ And I have imported required libraries and I have imported the dataset which was in excel format.
- ✓ Then I did all the statistical analysis like checking shape, nunique, value counts, info etc.....
- ✓ While checking for null values I found null values in the dataset and I replaced them using imputation technique.
- ✓ I have also droped Unnamed:0, cargo_volume and Insp_score column as I found they are useless.
- ✓ Next as a part of feature extraction I converted the data types of all the columns and I have extracted usefull information from the raw dataset. Thinking that this data will help us more than raw data.

2.4 Data Inputs- Logic- Output Relationships

- ✓ Since I had numerical columns I have plotted dist plot to see the distribution of skewness in each column data.
- ✓ I have used bar plot for each pair of categorical features that shows the relation between label and independent features.
- ✓ I have used reg plot and strip plot to see the relation between numerical columns with target column.
- ✓ I can notice there is a linear relationship between maximum columns and target.

2.5 Software required and tool 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.

Software/s required: -

1.Anaconda

Libraries required :-

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

```python
In [1]: #importing required libraries
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import datetime as dt

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

•**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 LabelEncoder**
- ✓ **from sklearn.preprocessing import StandardScaler**
- ✓ **from sklearn.ensemble import RandomForestRegressor**
- ✓ **from sklearn.tree import DecisionTreeRegressor**
- ✓ **from xgboost import XGBRegressor**
- ✓ **from sklearn.ensemble import GradientBoostingRegressor**
- ✓ **from sklearn.ensemble import ExtraTreesRegressor**
- ✓ **from sklearn.metrics import classification_report**
- ✓ **from sklearn.metrics import accuracy_score**
- ✓ **from sklearn.model_selection import cross_val_score**

With this sufficient libraries we can go ahead with our model building

3.Data Analysis and Visualization

3.1   Identification of possible problem-solving approaches (methods)

✓ **Since the data collected was not in the format we have to clean it and bring it to the proper format for our analysis. To remove outliers I have used z-score method. And to remove skewness I have used yeo-johnson method. We have dropped all the unnecessary columns in the dataset according to our understanding. Use of Pearson's correlation coefficient to check the correlation between dependent and independent features. Also I have used Standardisation to scale the data. After scaling we have to remove multicolinearity using VIF. Then followed by model building with all Regression algorithms**

3.2 Testing of Identified Approaches (Algorithms)

Since car_price was my target and it was a continuous column with improper format which has to be changed to continuous float datatype column, so this perticular problem was Regression problem. And I have used all Regression algorithms to build my model. By looking into the difference of r2 score and cross validation score I found DecisionTreeRegressor as a 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

I have used bar plots to see the relation of categorical feature with target and I have used 2 types of plots for numerical columns one is disp plot for univariate and reg plot, strip plot for bivariate analysis.

Univariate analysis for all numerical values:
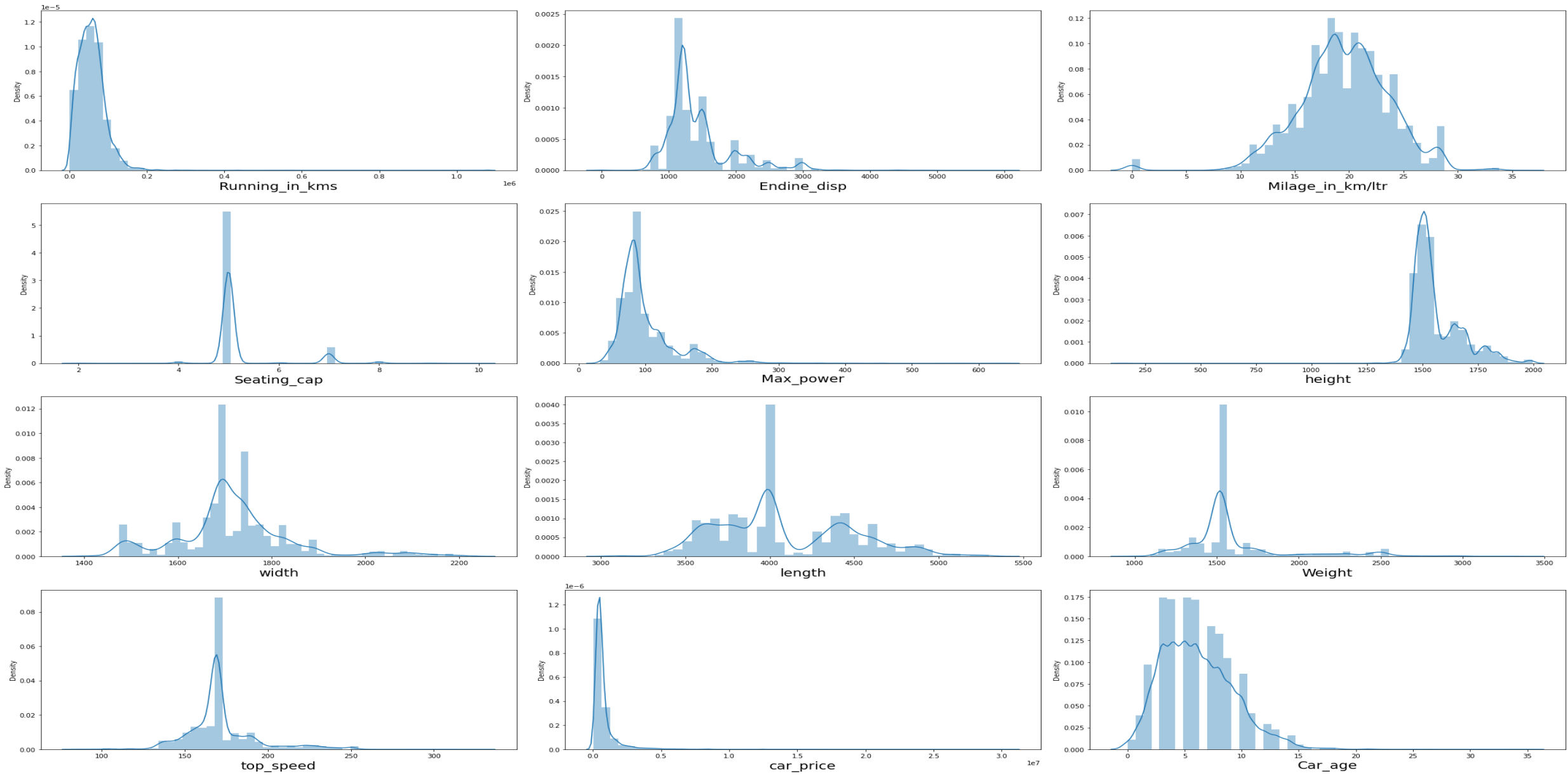
Univariate analysis for categorical column:
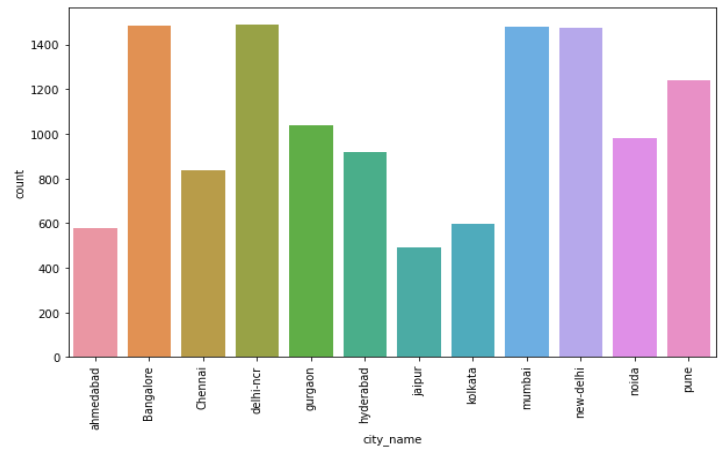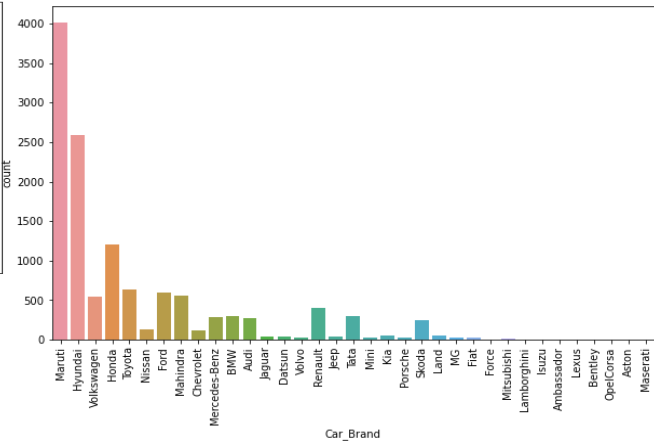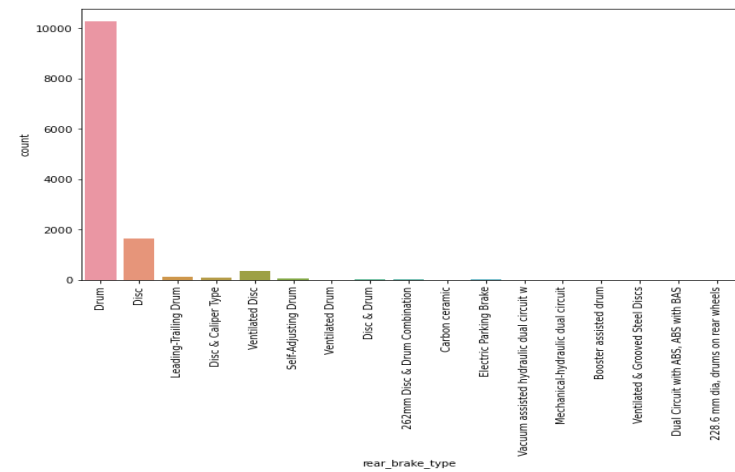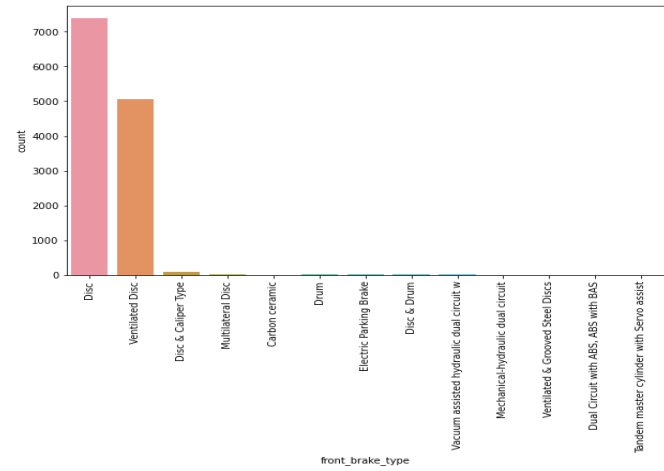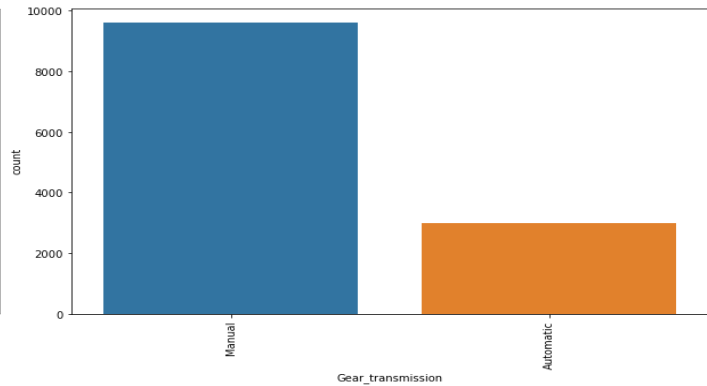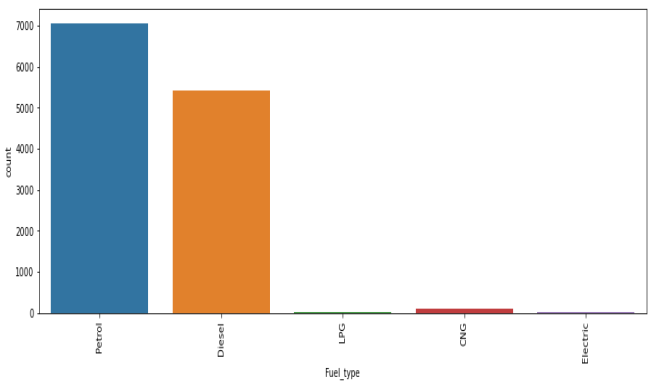
Observations:

- ✓ Maximum cars are petrol driven and also diesel driven.
- ✓ Maximum cars are with Manual gear transmission.
- ✓ Disc front brake cars are more in number followed by Ventilated Disc.
- ✓ Drum rare break cars are more in number.
- ✓ Maximum cars under sale are Maruti followed by Hyundai.
- ✓ In Bangalore, delhi-ncr, mumbai and new-delhi we can find maximum cars for sale. Since these are most populated places.

Bivariate Analysis for all numerical columns:
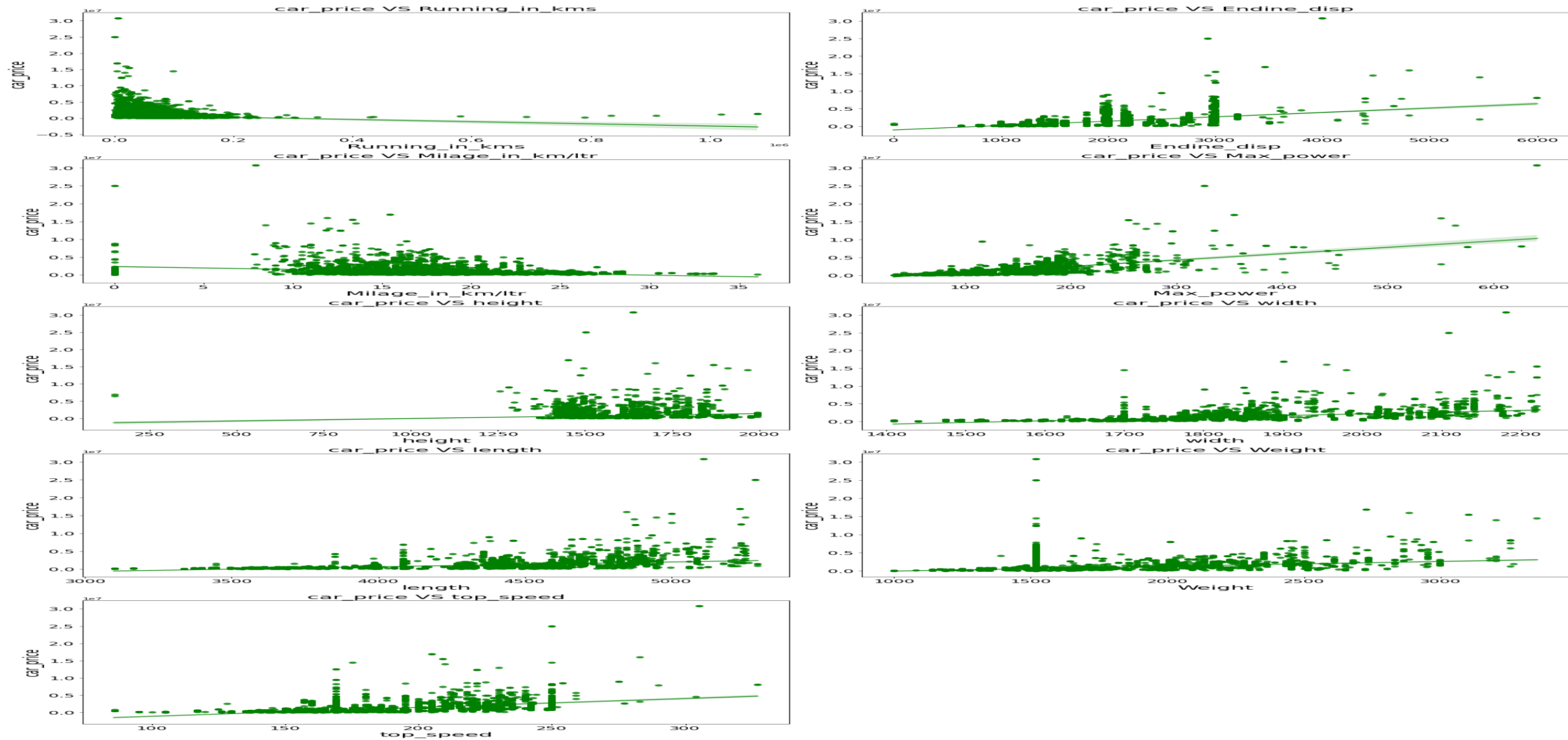
Observation:

- Maximum cars are having below 20k driven kms. And car price is high for less driven cars.
- Maximum cars are having 1000-3000 Endine_disp. And car price is high for 3000 Endine_disp.
- Maximum cars are having milage of 10-25kms. And ,milage has no proper relation with car price.
- As Max_power is increasing car price is also increasing.
- Car_price has no proper relation with height.
- As the width is increasing car price is also increasing.
- As length is increasing car price is also increasing.
- Weight also has linear relationship with car price.
- As top_speed is increasing car price is also increasing.

•Bivariate Analysis for categorical columns:
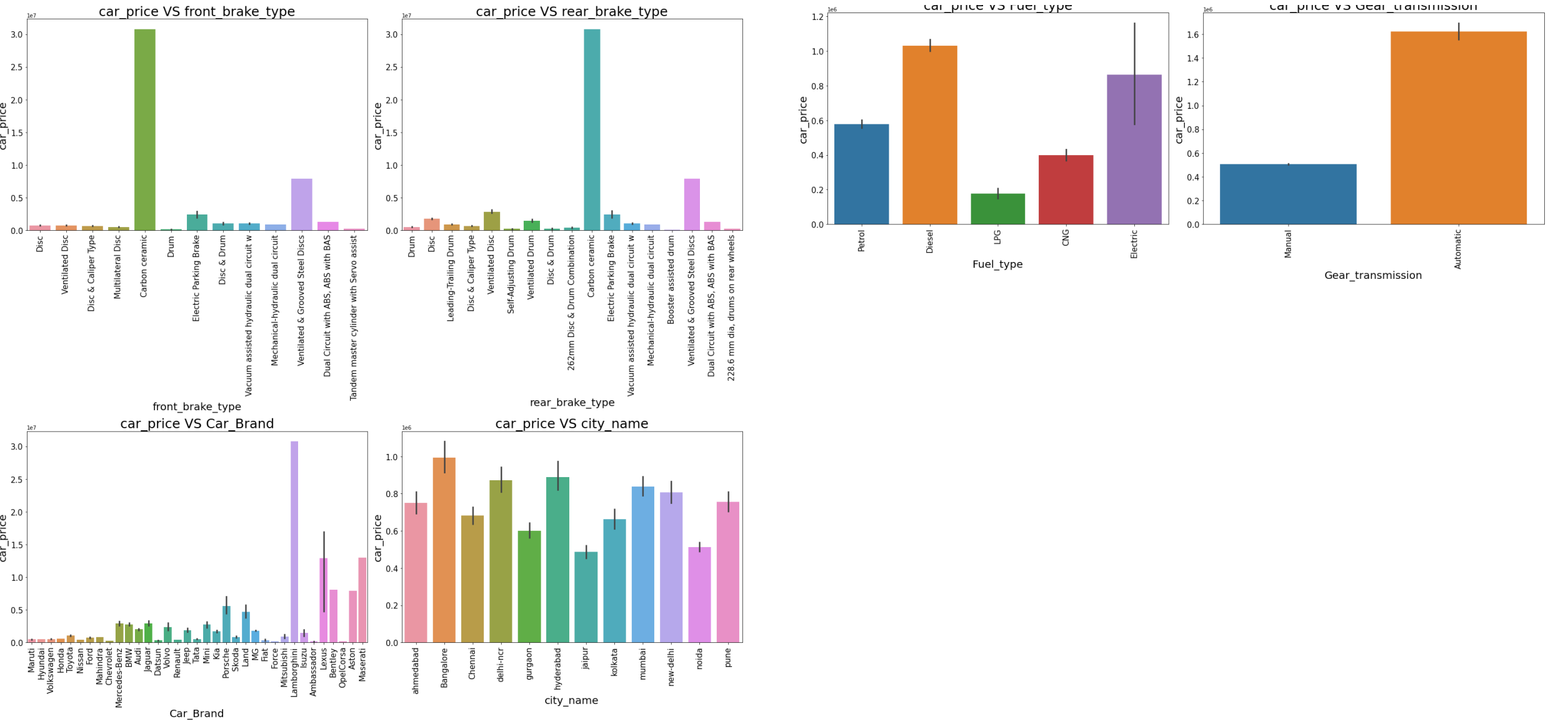
3.5 Run and Evaluate selected models

1. Model Building:

1) RandomForestRegressor:

i) RandomForestRegressor:

```
In [105]:    1  RFR=RandomForestRegressor()
             2  RFR.fit(X_train,y_train)
             3  pred=RFR.predict(X_test)
             4  R2_score = r2_score(y_test,pred)*100
             5  print('R2_score:',R2_score)
             6  print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
             7  print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
             8  print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
             9
            10  #cross validation score
            11  scores = cross_val_score(RFR, X, y, cv = 10).mean()*100
            12  print("\nCross validation score :", scores)
            13
            14  #difference of accuracy and cv score
            15  diff = R2_score - scores
            16  print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 96.59040318013854
mean_squared_error: 8898886369.94504
mean_absolute_error: 50460.44764563319
root_mean_squared_error: 94333.90890843568

Cross validation score : 93.04199593407235

R2_Score - Cross Validation Score : 3.5484072460661906
```

- **RandomForestRegressor has given me 96.59% r2_score and the difference between r2_score and cross validation score is 3.54%, but still we have to look into multiple models.**

ii) XGBRegressor:

```
[106]:    1  XGB=XGBRegressor()
          2  XGB.fit(X_train,y_train)
          3  pred=XGB.predict(X_test)
          4  R2_score = r2_score(y_test,pred)*100
          5  print('R2_score:',R2_score)
          6  print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
          7  print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
          8  print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
          9
         10  #cross validation score
         11  scores = cross_val_score(XGB, X, y, cv = 10).mean()*100
         12  print("\nCross validation score :", scores)
         13
         14  #difference of accuracy and cv score
         15  diff = R2_score - scores
         16  print("\nR2_Score - Cross Validation Score :", diff)
```

```
[12:19:10] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
R2_score: 94.90530613827012
mean_squared_error: 13296909916.473036
mean_absolute_error: 71774.88464033019
root_mean_squared_error: 115312.22795728577
[12:19:10] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[12:19:11] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[12:19:12] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[12:19:12] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[12:19:13] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[12:19:14] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[12:19:14] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[12:19:15] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[12:19:16] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
[12:19:16] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.

Cross validation score : 90.3356013590216

R2_Score - Cross Validation Score : 4.569704779248525
```

- **XGBRegressor has given me 94.90% r2_score and the difference between r2_score and cross validation score is 4.56%, but still we have to look into multiple models.**

iii) GradientBoostingRegressor:

```
In [107]:   1  GBR=GradientBoostingRegressor()
            2  GBR.fit(X_train,y_train)
            3  pred=GBR.predict(X_test)
            4  R2_score = r2_score(y_test,pred)*100
            5  print('R2_score:',R2_score)
            6  print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
            7  print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
            8  print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
            9
           10  #cross validation score
           11  scores = cross_val_score(GBR, X, y, cv = 10).mean()*100
           12  print("\nCross validation score :", scores)
           13
           14  #difference of accuracy and cv score
           15  diff = R2_score - scores
           16  print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 94.92463739776873
mean_squared_error: 13246456243.867523
mean_absolute_error: 71285.612608969
root_mean_squared_error: 115093.25020985167

Cross validation score : 90.21103303115027

R2_Score - Cross Validation Score : 4.713604366618469
```

- **GradientBoostingRegressor has given me 94.92% r2_score and the difference between r2_score and cross validation score is 4.71%, but still we have to look into multiple models.**

iv) DecisionTreeRegressor:

In [108]:
```
 1  DTR=DecisionTreeRegressor()
 2  DTR.fit(X_train,y_train)
 3  pred=DTR.predict(X_test)
 4  R2_score = r2_score(y_test,pred)*100
 5  print('R2_score:',R2_score)
 6  print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
 7  print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
 8  print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
 9
10  #cross validation score
11  scores = cross_val_score(DTR, X, y, cv = 10).mean()*100
12  print("\nCross validation score :", scores)
13
14  #difference of accuracy and cv score
15  diff = R2_score - scores
16  print("\nR2_Score - Cross Validation Score :", diff)
```

R2_score: 92.08778253214294
mean_squared_error: 20650513213.352684
mean_absolute_error: 63565.747460087085
root_mean_squared_error: 143702.86431853988

Cross validation score : 88.88438259743741

R2_Score - Cross Validation Score : 3.20339993470553

- **DecisionTreeRegressor has given me 92.08% r2_score and the difference between r2_score and cross validation score is 3.20%, but still we have to look into multiple models.**

v) Bagging Regressor:

```
In [109]:    1  BR=BaggingRegressor()
             2  BR.fit(X_train,y_train)
             3  pred=BR.predict(X_test)
             4  R2_score = r2_score(y_test,pred)*100
             5  print('R2_score:',R2_score)
             6  print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
             7  print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
             8  print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
             9
            10  #cross validation score
            11  scores = cross_val_score(BR, X, y, cv = 10).mean()*100
            12  print("\nCross validation score :", scores)
            13
            14  #difference of accuracy and cv score
            15  diff = R2_score - scores
            16  print("\nR2_Score - Cross Validation Score :", diff)
```

R2_score: 96.09573341664313
mean_squared_error: 10189951046.668966
mean_absolute_error: 54223.09514916949
root_mean_squared_error: 100945.28739207673

Cross validation score : 92.22516173338849

R2_Score - Cross Validation Score : 3.8705716832546386

- **Bagging Regressor has given me 96.09% r2_score and the difference between r2_score and cross validation score is 3.87%, but still we have to look into multiple models.**

Hyper Parameter Tunning:

Hyper parameter tunning for best model:

```
In [110]:   1  #importing necessary libraries
            2  from sklearn.model_selection import GridSearchCV
```

```
In [111]:   1  parameter = {'criterion':['squared_error', 'friedman_mse', 'absolute_error', 'poisson'],
            2               'splitter':['best','random'],
            3               'max_features':['auto','sqrt','log2'],
            4               'min_samples_split':[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],
            5               'max_depth':[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]}
```

Giving DecisionTreeRegressor parameters.

```
In [112]:   1  GCV=GridSearchCV(DecisionTreeRegressor(),parameter,cv=10)
```

```
In [119]:   1  Best_mod=DecisionTreeRegressor(criterion='friedman_mse',max_depth=15,max_features='auto',min_samples_split=4,splitter='rando
            2  Best_mod.fit(X_train,y_train)
            3  pred=Best_mod.predict(X_test)
            4  print('R2_Score:',r2_score(y_test,pred)*100)
            5  print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
            6  print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
            7  print("RMSE value:",np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2_Score: 92.94036037484042
mean_squared_error: 18425325384.838062
mean_absolute_error: 70852.59114405994
RMSE value: 135739.91817014647
```

- I have choosed all parameters of DecisionTreeRegressor, after tunning the model with best parameters I have incresed my model accuracy from 91.79% to 92.29%.

Saving the model and predictions from saved best model.

Saving the model:

```
In [120]:   1  # Saving the model using .pkl
            2  import joblib
            3  joblib.dump(Best_mod,"Car_Price.pkl")

Out[120]: ['Car_Price.pkl']
```

```
In [121]:   1  # Loading the saved model
            2  model=joblib.load("Car_Price.pkl")
            3
            4  #Prediction
            5  prediction = model.predict(X_test)
            6  prediction

Out[121]: array([ 354500.        ,  1650000.        ,   400000.        ,  ...,
                   557000.        ,   200000.        ,   397142.85714286])
```

✓ I have predicted the Car Price using saved model, and the predictions look good. The Predicted values are almost same as actual values.

Ploting the predicted values v/s actual values:

```
In [119]: plt.figure(figsize=(10,5))
          plt.scatter(y_test, prediction, c='crimson')
          p1 = max(max(prediction), max(y_test))
          p2 = min(min(prediction), min(y_test))
          plt.plot([p1, p2], [p1, p2], 'b-')
          plt.xlabel('Actual', fontsize=15)
          plt.ylabel('Predicted', fontsize=15)
          plt.title("DecisionTreeRegressor")
          plt.show()
```
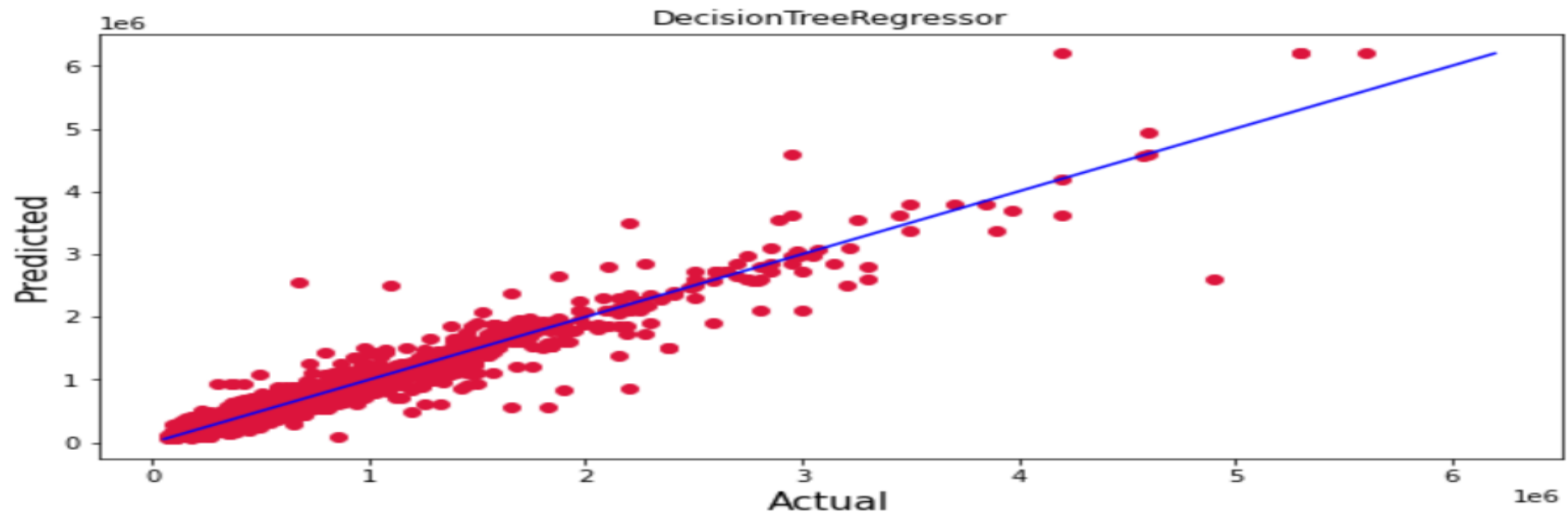

- **Plotting Actual vs Predicted, To get better insight. Bule line is the actual line and red dots are the predicted values.**

5.6 Interpretation of the Results

- ✓ The dataset was scrapped from cardekho website.
- ✓ The dataset was very challenging to handle it had 20 features with 12608 samples.
- ✓ Firstly, the datasets were having any null values, so I have used imputation method to replace the nan values.
- ✓ And there was huge number of unnecessary entries in all the features so I have used feature extraction to get the required format of variables.
- ✓ And proper ploting for proper type of features will help us to get better insight on the data. I found both numerical columns and categorical columns in the dataset so I have choosen reg plot, strip plot and bar plot to see the relation between target and features.
- ✓ I notice a huge amount of outliers and skewness in the data so we have choose proper methods to deal with the outliers and skewness. If we ignore this outliers and skewness we may end up with a bad model which has less accuracy.
- ✓ Then scaling dataset has a good impact like it will help the model not to get baised. Since we have removed outliers and skewness from the dataset so we have to choose Standardisation.
- ✓ We have to use multiple models while building model using dataset as to get the best model out of it.
- ✓ And we have to use multiple metrics like mse, mae, rmse and r2_score which will help us to decide the best model.
- ✓ I found DecisionTreeRegressor as the best model with 91.79% r2_score. Also I have improved the accuracy of the best model by running hyper parameter tunning.
- ✓ At last I have predicted the used car price using saved model. It was good!! that I was able to get the predictions near to actual values.

4.CONCLUSION

4.1 Key Findings and Conclusions of the Study

In this project report, we have used machine learning algorithms to predict the used car prices. We have mentioned the step by step procedure to analyze the dataset and finding the correlation between the features. Thus we can select the features which are correlated to each other and are independent in nature. These feature set were then given as an input to five algorithms and a hyper parameter tunning was done to the best model and the accuracy has been improved. Hence we calculated the performance of each model using different performance metrics and compared them based on those metrics. Then we have also saved the best model and predicted the used car price. It was good the the predicted and actual values were almost same

4.2 Learning Outcomes of the Study in respect of Data Science

I found that the dataset was quite interesting to handle as it contains all types of data in it and it was self scrapped from cardekho website using selenium. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analysed. New analytical techniques of machine learning can be used in used car price research. **The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove unrealistic values and null values. This study is an exploratory attempt to use five machine learning algorithms in estimating used car price prediction, and then compare their results.**

To conclude, the application of machine learning in predicting used car price is still at an early stage. We hope this study has moved a small step ahead in providing some methodological and empirical contributions to crediting online platforms, and presenting an alternative approach to the valuation of used car price. Future direction of research may consider incorporating additional used car data from a larger economical background with more features.

4.3 Limitations of this work and Scope for Future Work

✓ First draw back is scrapping the data as it is fluctuating process.
✓ Followed by more number of outliers and skewness these two will reduce our model accuracy.
✓ Also, we have tried best to deal with outliers, skewness and null values. So it looks quite good that we have achieved a accuracy of 92.29% even after dealing all these drawbacks.
✓ Also, this study will not cover all Regression algorithms instead, it is focused on the chosen algorithm, starting from the basic ensembling techniques to the advanced ones.

THANK

YOU

FLIP ROBO