FLIP ROBO

Project Report On
Car Price Prediction

Submitted by

Pritam Sangle

ACKNOWLEDGMENT

I would like to express my sincere thanks and gratitude to my SME as well as "FlipRobo Technologies" team for letting me work on "Used Car Price Prediction" project also huge thanks to my academic team "Data Trained". Their suggestions and directions have helped me in the completion of this project successfully. This project also helped me in doing lots of research wherein I came to know about so many new things.

Contents:

1. INTRODUCTION

1.1 Business Problem Framing:

Car price prediction is a somehow interesting and popular problem. As per the 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 this trend will likely continue, and the number of cars will increase in the future. This adds additional significance to the problem of car price prediction. Accurate car price prediction involves expert knowledge because price usually depends on many distinctive features and factors. Typically, the 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 affects the price of a car due to frequent changes in the price of fuel. Different features like exterior color, door number, type of transmission, dimensions, safety, air condition, interior, and whether it has navigation or not will also influence the car price. In this report, we applied different methods and techniques to achieve higher precision in the used car price prediction.

With the covid 19 impact on the market, we have seen a 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 the 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 a car price valuation model.

1.2 Conceptual Background of the Domain Problem

The prices of new cars in the industry are 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 offer this service, their prediction method may not be the best. Besides, different models and systems may contribute to predicting the power of 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 point 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 its market value.

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

1.3 Review of Literature

The second-hand car market has continued to expand even with 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 a business for both buyers and sellers. Most people prefer to buy 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, color, 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

Some websites offer an estimated 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 tell 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 the carsdekho 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 were 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 the card echo website the raw data was not in the format, so we have used feature engineering to extract the required feature format. To get a better insight into the features I have used plotting like distribution plot, bar plot, reg plot, strip plot,t, and count plot. With this plotting, I was able to understand the relation between the features in a better manner. Also, I found outliers and skewness in the dataset so I removed outliers using the z-score method and I removed skewness us the ing yeo-johnson method. I have used all the regression algorithms while building the model then tunned the best model and saved the best model. At, last I have predicted the car price using the saved model.

2.2 Data Sources and their formats

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

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

Features Information:

- Car_Name: Name of the car with Year
- Fuel_type: Type of fuel used for car engine
- Running_in_kms: Car running in km till the date

- Endine_disp : Engine displacement/engine CC
- Gear_transmission: Type of gear transmission used in car
- Milage_in_km/ltr : Overall mileage of car in Km/ltr
- Seating_cap: Availability of several seats in the car
- color: Car color
- Max_power: Maximum power of engine used in a car in bhp
- front_brake_type: type of brake system used for front-side wheels
- rear_brake_type: type of brake system used for back-side wheels
- cargo_volume: the total cubic feet of space in a car's cargo area.
- height: Total height of the car in mm
- width: Width of the car in mm
- length: TOtal length of the car in mm
- Weight: Gross weight of the car in kg
- Insp_score: inspection rating out of 10
- top_speed: Maximum speed limit of the car in km per hour
- City_url: Url of the page of cars from a particular city
- Car_price: Price of the car

2.3 Data Preprocessing Done

- ✓ As a first step, I scrapped the required data using selenium from the cardekho website.
- ✓ And I have imported the required libraries and I have imported the dataset which was in excel format.
- ✓ Then I did all the statistical analysis like checking shape, uniqueness, value counts, info, etc…..
- ✓ While checking for null values I found null values in the dataset and I replaced them using the imputation technique.
- ✓ I have also dropped Unnamed:0, cargo_volume, and Insp_score columns as I found they are useless.
- ✓ Next as a part of feature extraction, I converted the data types of all the columns and I extracted useful information from the raw dataset. Think 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 a dist plot to see the distribution of skewness in each column of data.
- ✓ I have used a bar plot for each pair of categorical features that shows the relation between the 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 Hardware and Software Requirements and Tools 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.

Hardware required: -

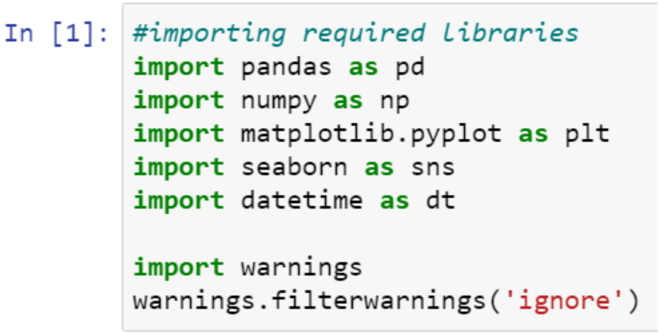
1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software/s required: -

1. Anaconda

Libraries required:-

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



```
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 that 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 the exploration and understanding of data.
- ✓ **Import matplotlib. pyplot as plt:** matplotlib. pyplot is a collection of functions that make matplotlib work like MATLAB. Each plot 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 these sufficient libraries, we can go ahead with our model building.

2. **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 the z-score method. And to remove skewness I have used the log transformation 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 multicollinearity 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 an improper format that has to be changed to a continuous float datatype column, this particular problem was a Regression problem. And I have used all Regression algorithms to build my model. By looking into the difference between the r2 score and cross-validation score I found DecisionTreeRegressor as the best model with the least difference. Also, to get the best model we have to run through multiple models, and to avoid the confusion of overfitting we have to go through cross-validation. Below is 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 the problem under consideration

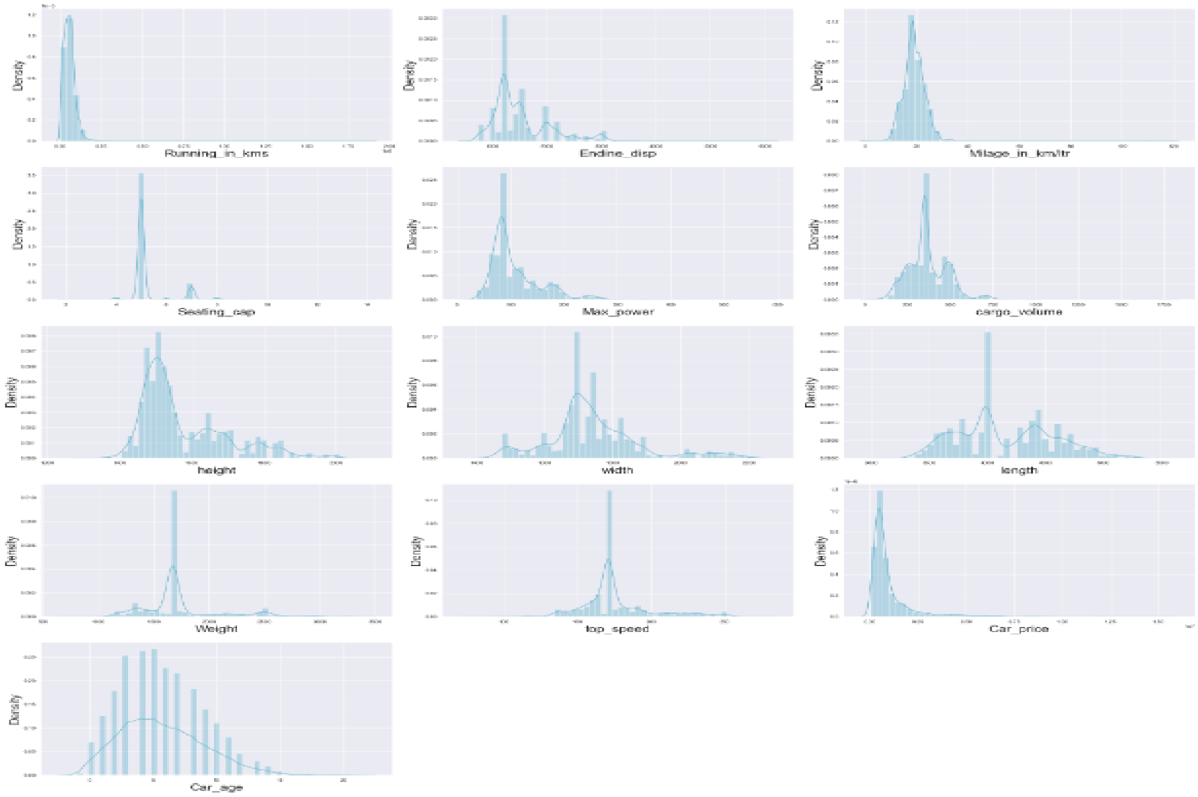
I have used the following metrics for evaluation:

- I have used mean absolute error which gives the magnitude of difference between the prediction of observation and the true value of that observation.
- I have used root mean square deviation as one of the most commonly used measures for evaluating the quality of predictions.
- I have used the 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.

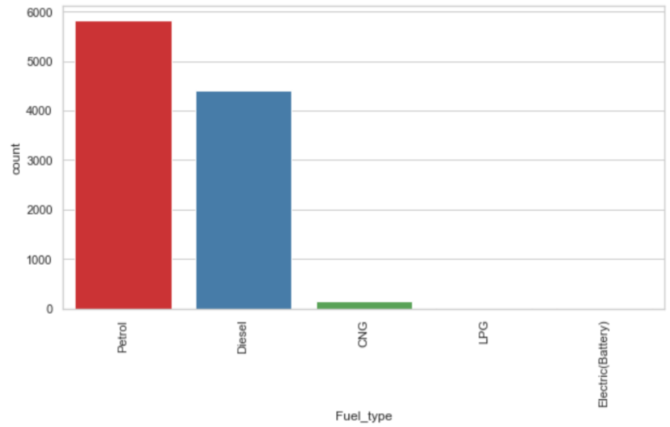
1. Univariate Analysis for numerical columns:

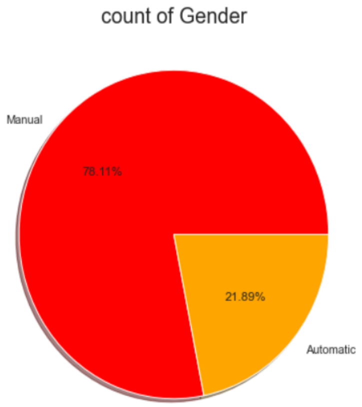


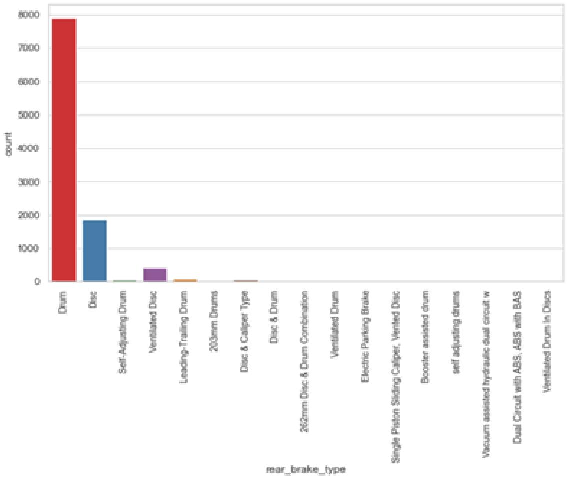
Observations:

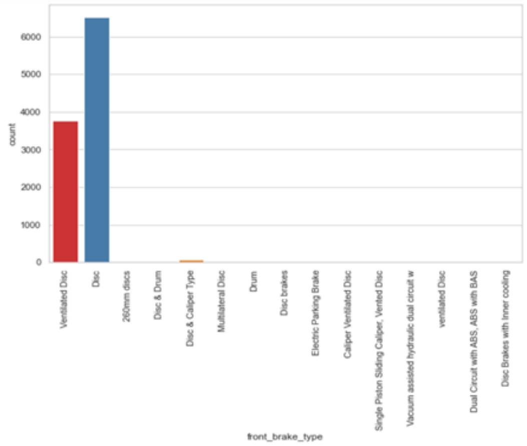
- ✓ We can see that there is skewness in most of the columns so we have to treat them using suitable methods.

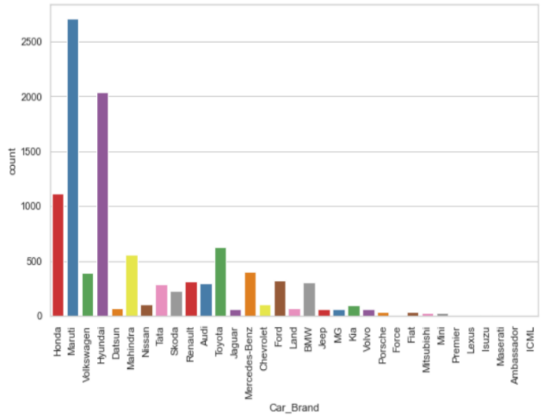
2. Univariate analysis for a categorical column:

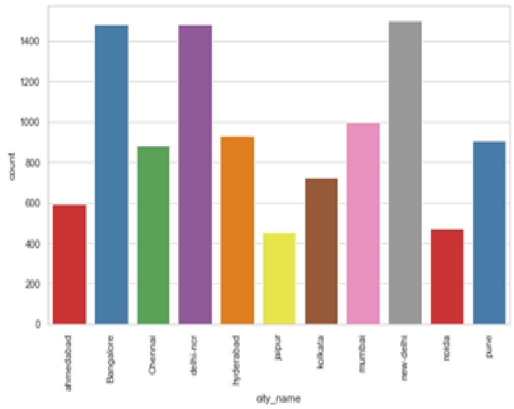








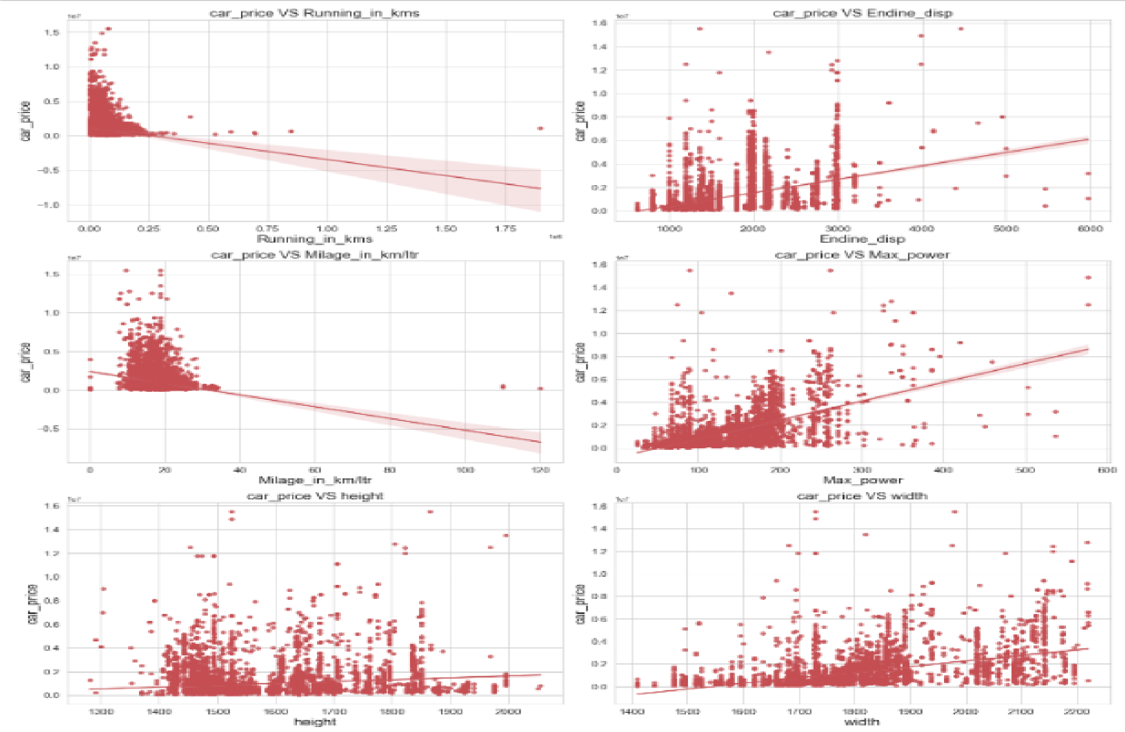


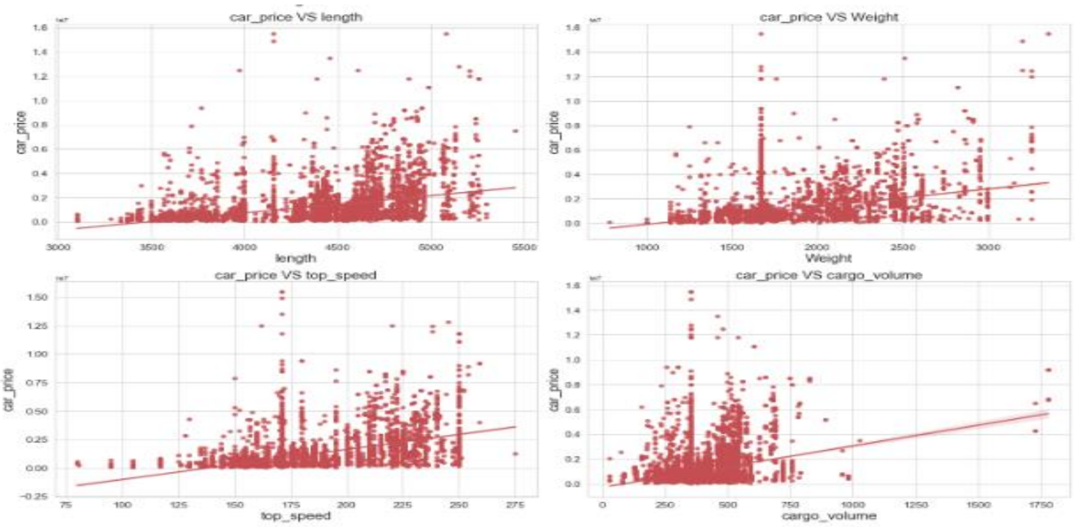


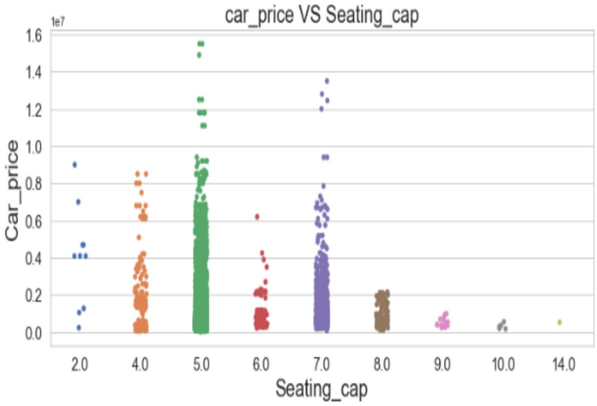
Observations:

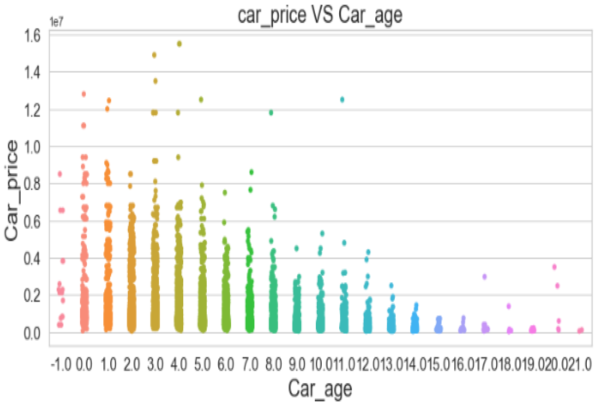
- ✓ Maximum cars are petrol driven followed by diesel driven.
- ✓ Maximum cars are with Manual gear transmission. (Almost 80%)
- ✓ Disc front brake cars are more in number followed by Ventilated Disc.
- ✓ Drum rare break cars are more in number.
- ✓ Maximum cars under sale are Maruti followed by Hyundai.
- ✓ In Bangalore, Delhi-NCR, Mumbai, and New Delhi we can find maximum cars for sale. Since these are the most populated places.

3. Bivariate analysis for numerical columns:





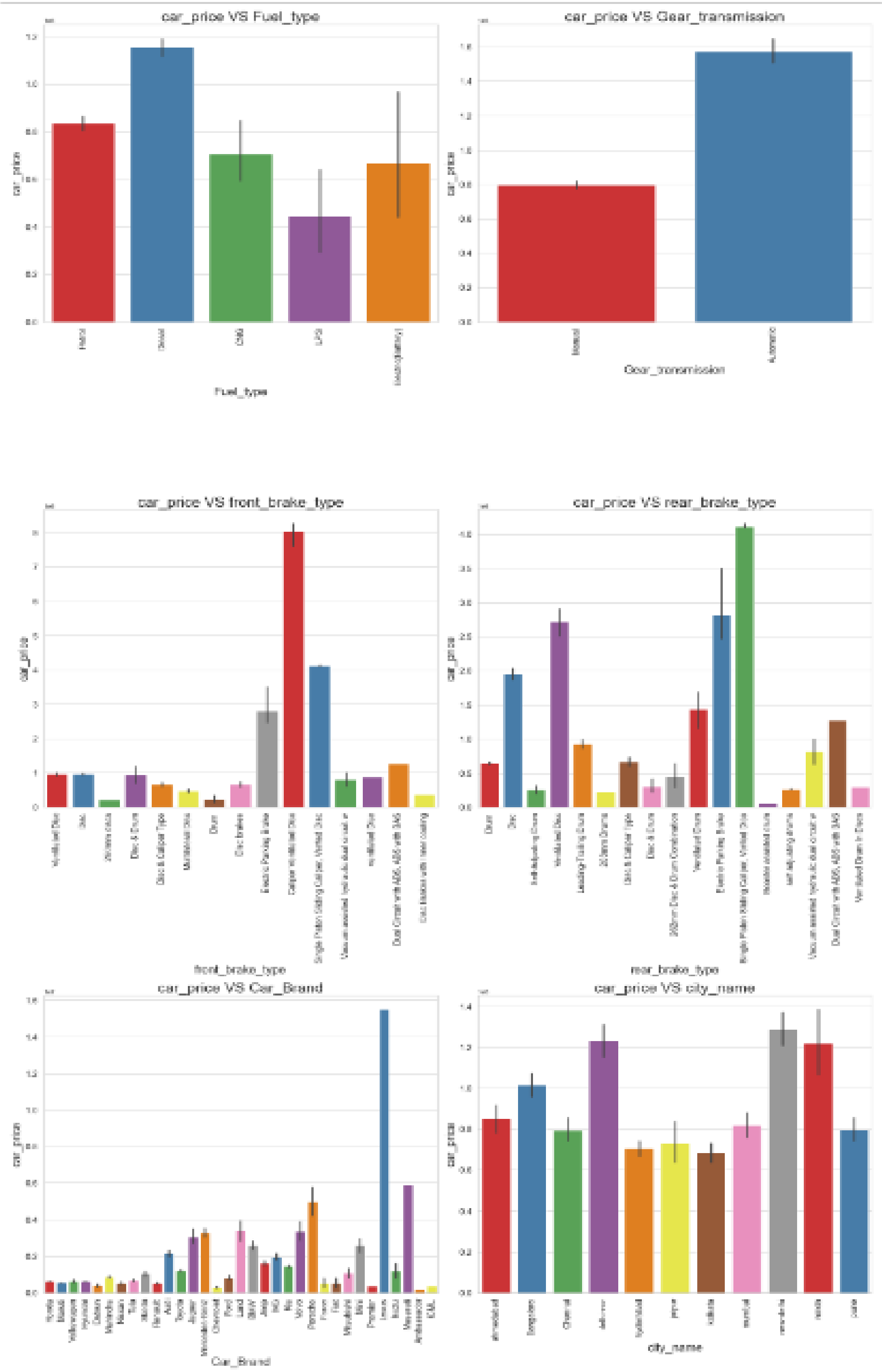




Observations:

- ✓ Maximum cars are having below 20k driven km. And car price is high for less driven cars.
- ✓ Maximum cars are having 1000-3000 Endine_disp. And car price is high for higher Endine_disp.
- ✓ Maximum cars are having milage of 10-30kms. 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 is also directly proportional to car price.
- ✓ As top_speed is increasing car price is also increasing.
- ✓ cargo_volume is also directly proportional to the price of the car.
- ✓ Cars with 5, 7 & 4 seats are having the highest price.
- ✓ As the age of the car increases the car price decreases.

4. Bivariate Analysis for categorical columns:



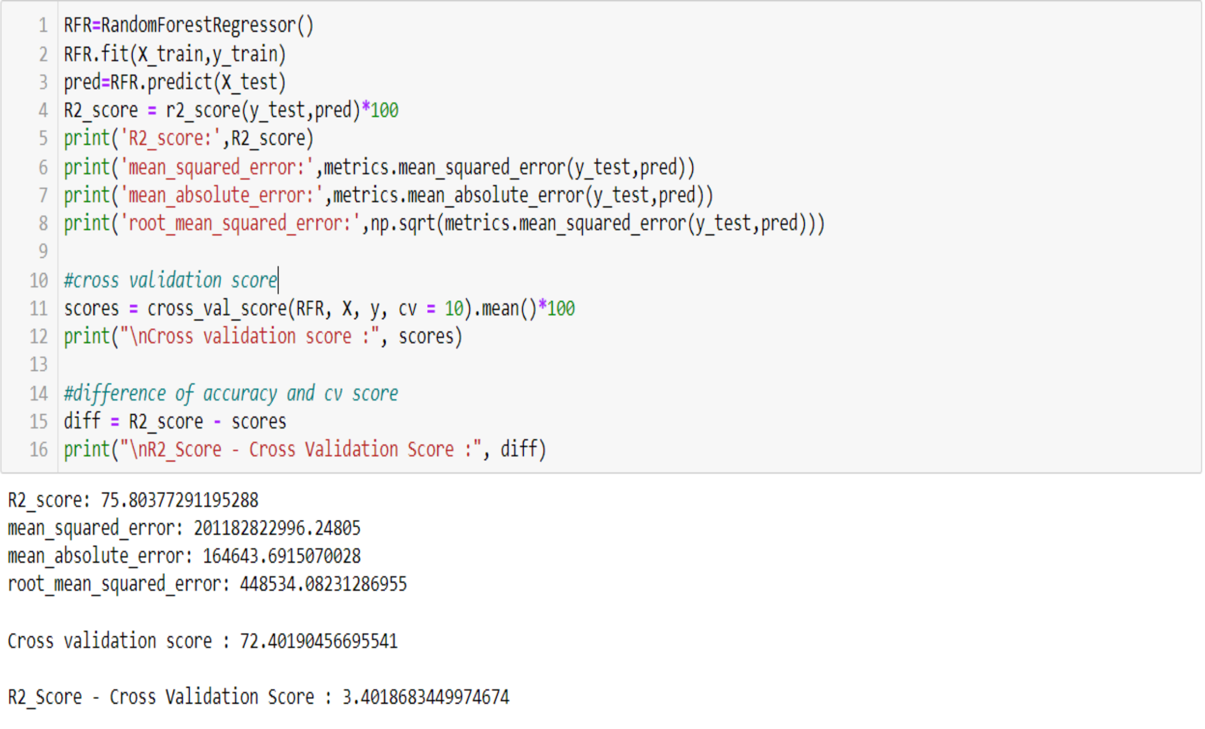
Observations:

- ✓ For Diesel, Petrol & Electric cars, the price is high compared to LPG and CNG.
- ✓ Cars with automatic gear are costlier than manual gear cars.
- ✓ Cars with Caliper ventilated front disc brakes are costlier compared to other cars.
- ✓ Cars with the single-piston sliding caliper and Vented rear disc brakes are costlier compared to other cars.
- ✓ Lexus brand cars are having 16oida16t sale price. (Lexus is the luxury vehicle division of the Japanese automaker Toyota)
- ✓ In New Delhi, Delhi-NCR, Noida & Bangalore car prices are high as they are highly populated cities.

3.5 Run and Evaluate selected models

1. Model Building:

1) RandomForestRegressor:



```
RFR=RandomForestRegressor()
RFR.fit(X_train,y_train)
pred=RFR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(RFR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

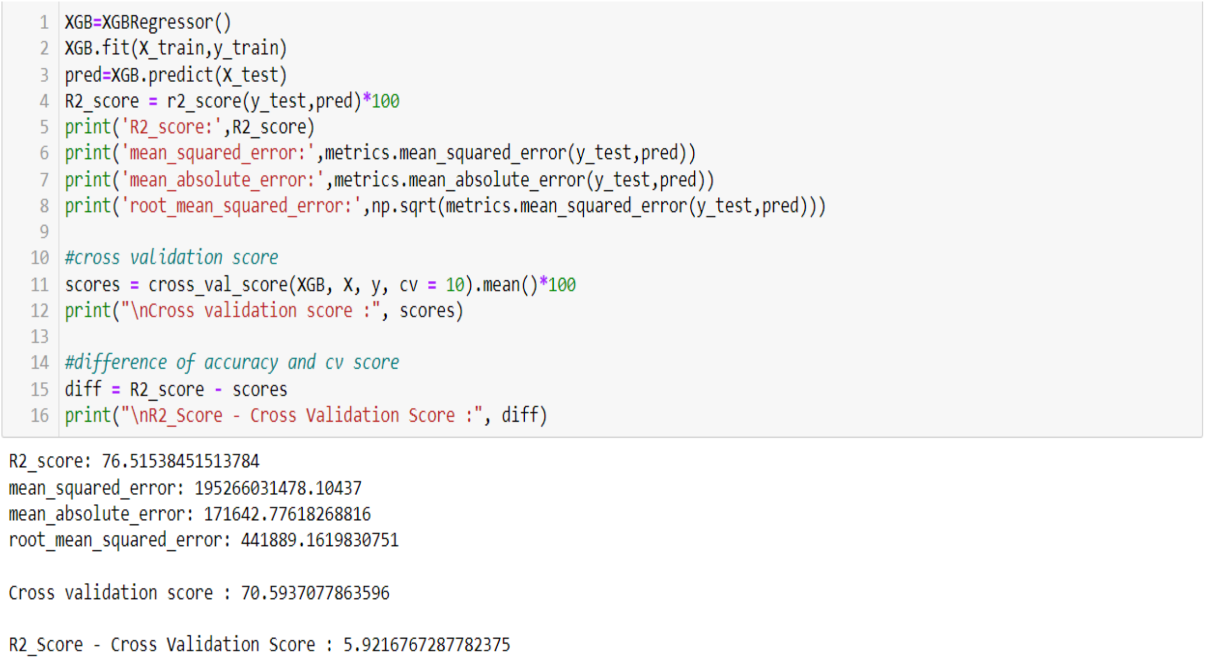
```
R2_score: 75.80377291195288
mean_squared_error: 201182822996.24805
mean_absolute_error: 164643.6915070028
root_mean_squared_error: 448534.08231286955

Cross validation score : 72.40190456695541

R2_Score - Cross Validation Score : 3.4018683449974674
```

- RandomForestRegressor has given me a 75.80% r2_score and the difference between r2_score and cross-validation score is 3.40, but still, we have to look into multiple models.

2) XGBRegressor:



```
XGB=XGBRegressor()
XGB.fit(X_train,y_train)
pred=XGB.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(XGB, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

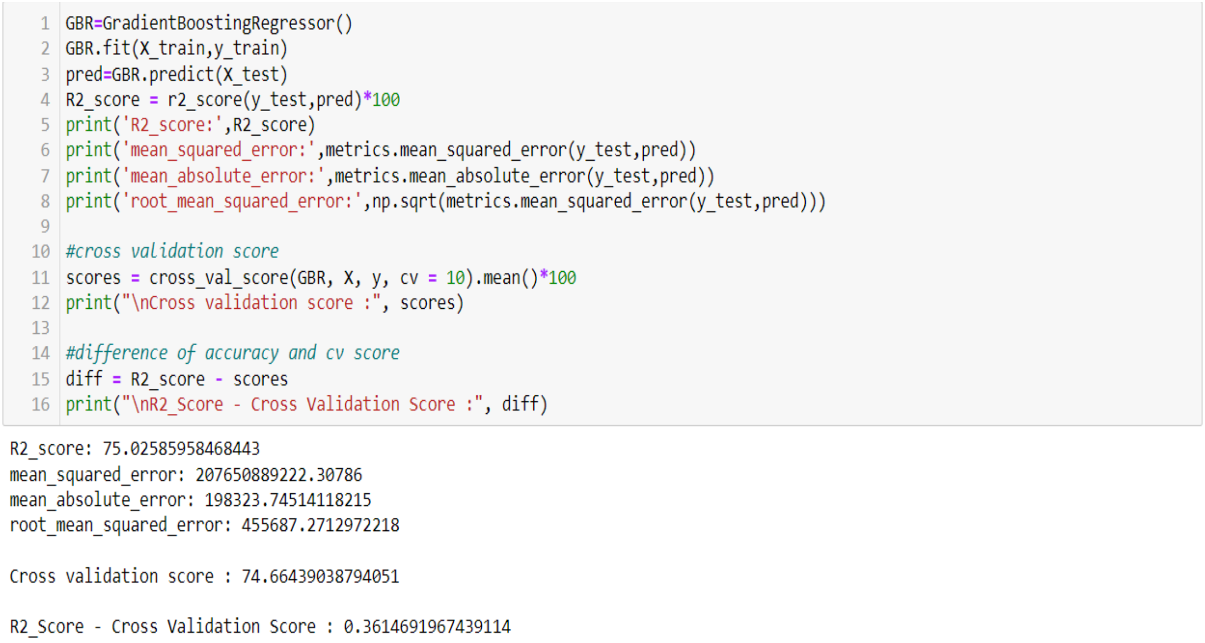
```
R2_score: 76.51538451513784
mean_squared_error: 195266031478.10437
mean_absolute_error: 171642.77618268816
root_mean_squared_error: 441889.1619830751

Cross validation score : 70.5937077863596

R2_Score - Cross Validation Score : 5.9216767287782375
```

- XGBRegressor is giving me 76.51% r2_score and the difference between r2_score and cross-validation score is 5.92.

3) GradientBoostingRegressor:



```
GBR=GradientBoostingRegressor()
GBR.fit(X_train,y_train)
pred=GBR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(GBR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

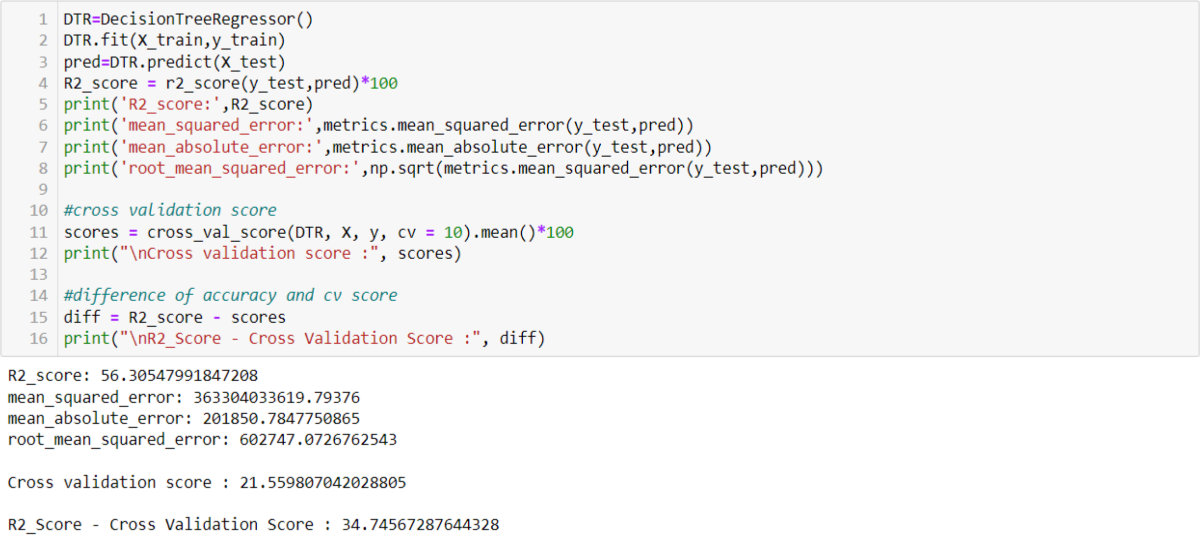
```
R2_score: 75.02585958468443
mean_squared_error: 207650889222.30786
mean_absolute_error: 198323.74514118215
root_mean_squared_error: 455687.2712972218

Cross validation score : 74.66439038794051

R2_Score - Cross Validation Score : 0.3614691967439114
```

- GradientBoostingRegressor is giving me 75.02% r2_score and the difference between r2_score and cross-validation score is 0.36.

4) DecisionTreeRegressor:



```python
DTR=DecisionTreeRegressor()
DTR.fit(X_train,y_train)
pred=DTR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(DTR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

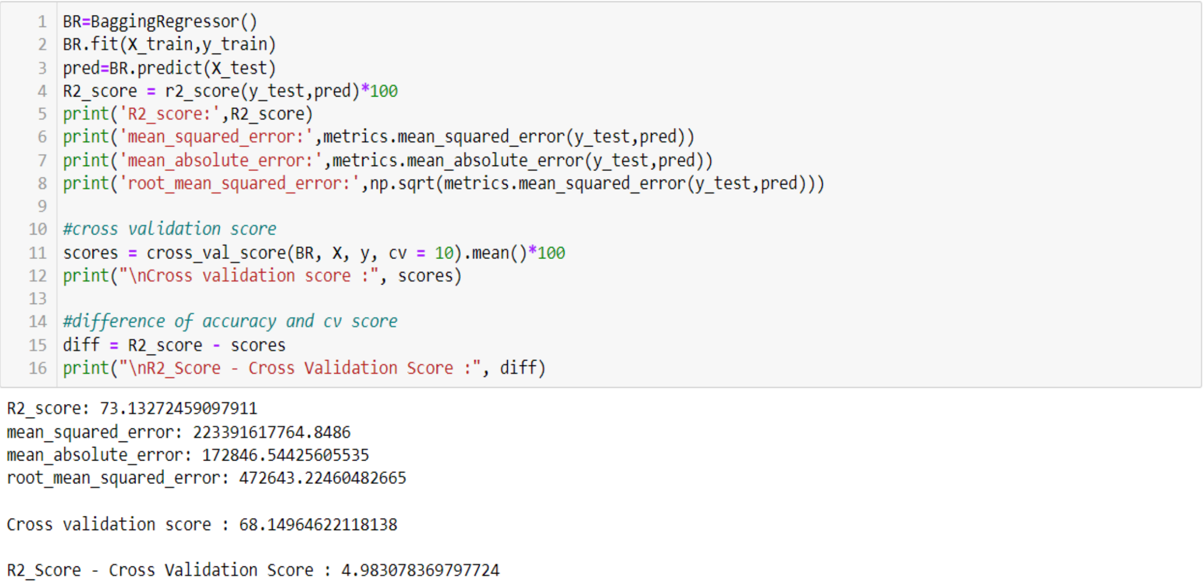
```
R2_score: 56.30547991847208
mean_squared_error: 363304033619.79376
mean_absolute_error: 201850.7847750865
root_mean_squared_error: 602747.0726762543

Cross validation score : 21.559807042028805

R2_Score - Cross Validation Score : 34.74567287644328
```

- DecisionTreeRegressor is giving me 56.30% r2_score and the difference between r2_score and cross-validation score is 34.

5) BaggingRegressor:



```python
BR=BaggingRegressor()
BR.fit(X_train,y_train)
pred=BR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(BR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

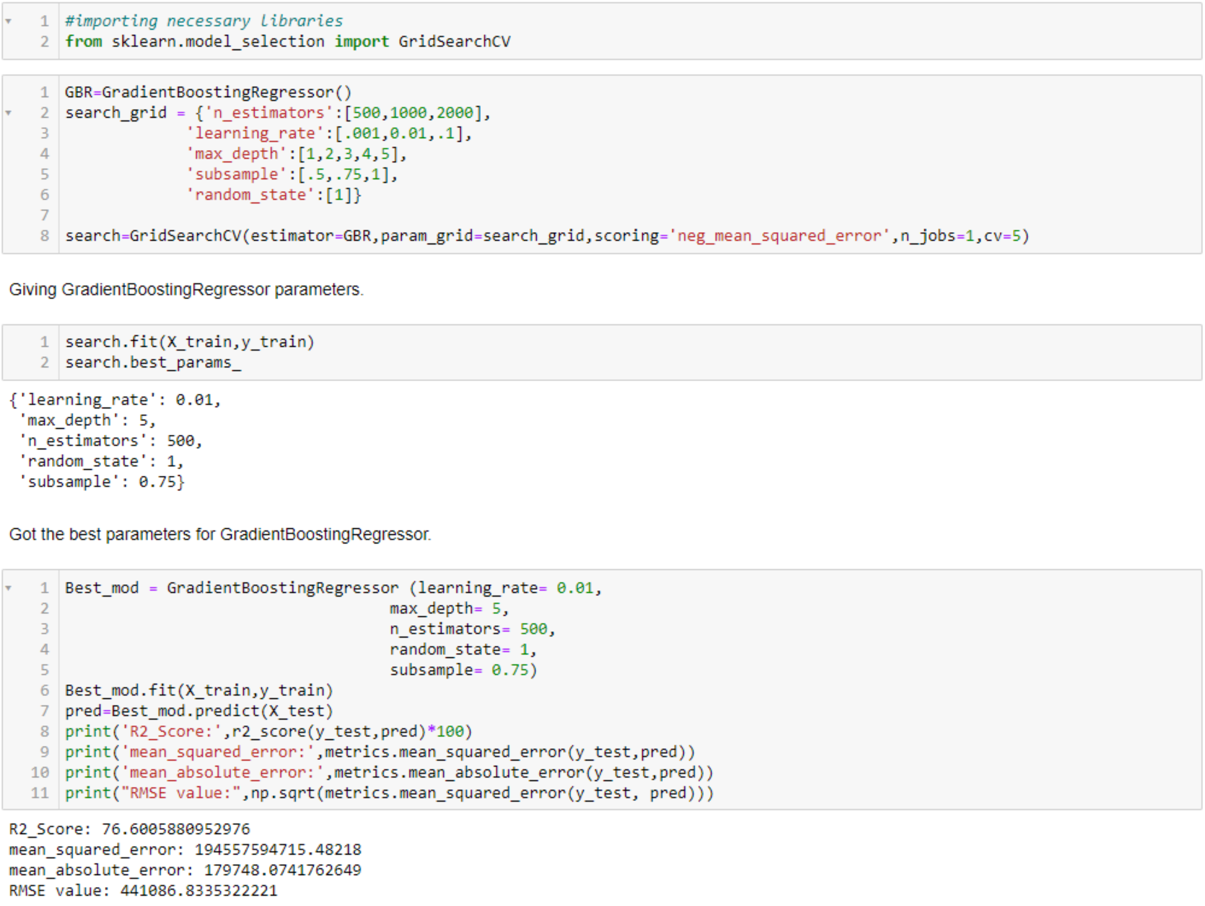
```
R2_score: 73.13272459097911
mean_squared_error: 223391617764.8486
mean_absolute_error: 172846.54425605535
root_mean_squared_error: 472643.22460482665

Cross validation score : 68.14964622118138

R2_Score - Cross Validation Score : 4.983078369797724
```

- BaggingRegressor is giving me 73.13% r2_score and the difference between r2_score and cross-validation score is 4.98.
- **By looking into the difference between model accuracy and cross-validation score I found Gradient Boosting Regressor as the best model with a 75.02% r2_score.**

2. Hyper Parameter Tuning:



```
#importing necessary libraries
from sklearn.model_selection import GridSearchCV
```

```
GBR=GradientBoostingRegressor()
search_grid = {'n_estimators':[500,1000,2000],
               'learning_rate':[.001,0.01,.1],
               'max_depth':[1,2,3,4,5],
               'subsample':[.5,.75,1],
               'random_state':[1]}

search=GridSearchCV(estimator=GBR,param_grid=search_grid,scoring='neg_mean_squared_error',n_jobs=1,cv=5)
```

Giving GradientBoostingRegressor parameters.

```
search.fit(X_train,y_train)
search.best_params_
```

```
{'learning_rate': 0.01,
 'max_depth': 5,
 'n_estimators': 500,
 'random_state': 1,
 'subsample': 0.75}
```

Got the best parameters for GradientBoostingRegressor.

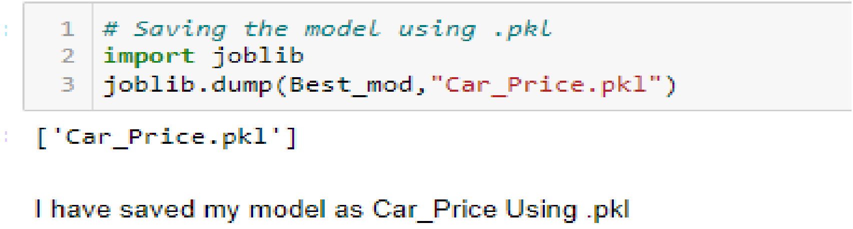
```
Best_mod = GradientBoostingRegressor (learning_rate= 0.01,
                                   max_depth= 5,
                                   n_estimators= 500,
                                   random_state= 1,
                                   subsample= 0.75)
Best_mod.fit(X_train,y_train)
pred=Best_mod.predict(X_test)
print('R2_Score:',r2_score(y_test,pred)*100)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print("RMSE value:",np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2_Score: 76.6005880952976
mean_squared_error: 194557594715.48218
mean_absolute_error: 179748.0741762649
RMSE value: 441086.8335322221
```

- **I have chosen all parameters of Gradient Boosting Regressor, after tunning the model with the best parameters I have increased my model accuracy from 75.02% to 76.60%.**

5. Saving the model and Predictions:

- I have saved my best model using .pkl as follows.

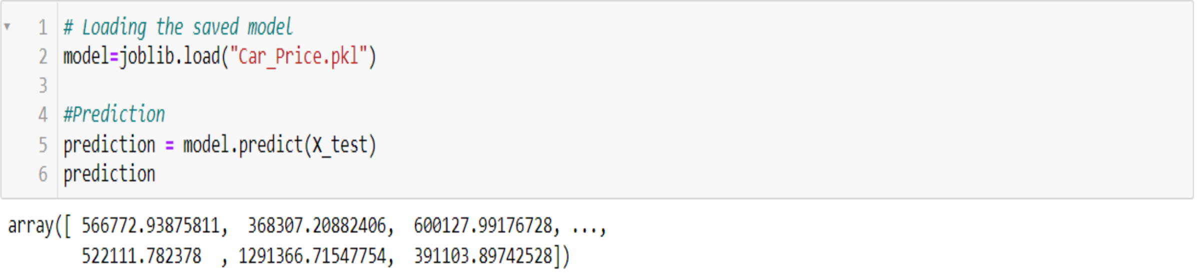


```
# Saving the model using .pkl
import joblib
joblib.dump(Best_mod,"Car_Price.pkl")
```

```
['Car_Price.pkl']
```

I have saved my model as Car_Price Using .pkl

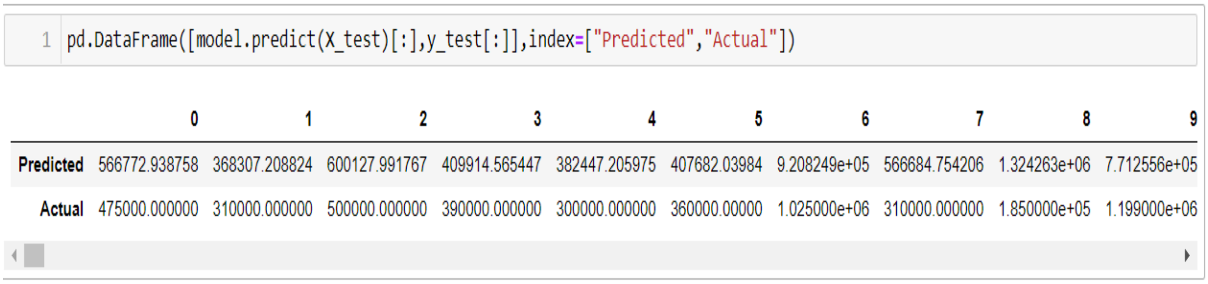
- Now loading my saved model and predicting the price values.



```
# Loading the saved model
model=joblib.load("Car_Price.pkl")

#Prediction
prediction = model.predict(X_test)
prediction
```

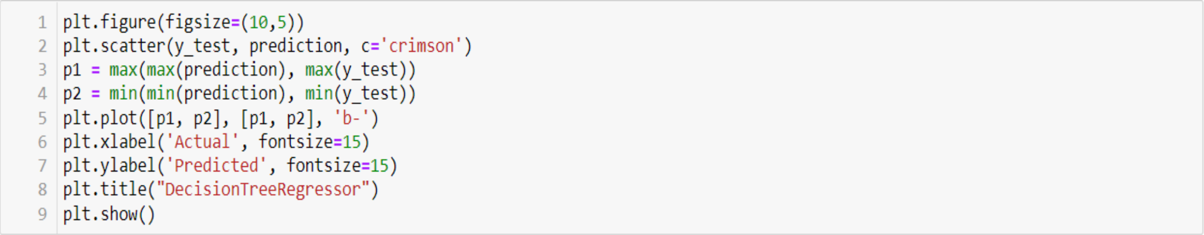
```
array([ 566772.93875811,  368307.20882406,  600127.99176728, ...,
        522111.782378  , 1291366.71547754,  391103.89742528])
```



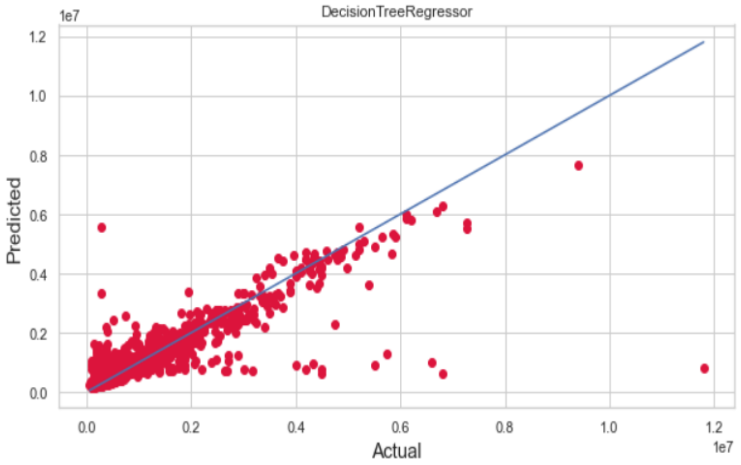
```
pd.DataFrame([model.predict(X_test)[:],y_test[:]],index=["Predicted","Actual"])
```

	0	1	2	3	4	5	6	7	8	9
Predicted	566772.938758	368307.208824	600127.991767	409914.565447	382447.205975	407682.03984	9.208249e+05	566684.754206	1.324263e+06	7.712556e+05
Actual	475000.000000	310000.000000	500000.000000	390000.000000	300000.000000	360000.00000	1.025000e+06	310000.000000	1.850000e+05	1.199000e+06

Above are the predicted values and the actual values.They are almost similar.



```
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.

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

3.6 Interpretation of the Results

- ✓ The dataset was scraped from the card echo website.
- ✓ The dataset was very challenging to handle it had 20 features with 10420 samples.
- ✓ Firstly, the datasets were having any null values, so I used the imputation method to replace the nan values.
- ✓ And there was a huge number of unnecessary entries in all the features so I have used feature extraction to get the required format of variables.
- ✓ And proper plotting for the proper type of features will help us to get a better insight into the data. I found both numerical columns and categorical columns in the dataset so I have chosen 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 to choose proper methods to deal with the outliers and skewness. If we ignore these outliers and skewness, we may end up with a bad model which has less accuracy.
- ✓ Then scaling the dataset has a good impact like it will help the model not to get biased. Since we have removed outliers and skewness from the dataset so we have to choose Standardisation.
- ✓ We have to use multiple models while building models using the dataset to get the best model out of it.
- ✓ And we have to use multiple metrics like me, mae, rmse, and r2_score which will help us to decide on the best model.
- ✓ I found Gradient boosting Regressor as the best model with a 74.9% r2_score. Also, I have improved the accuracy of the best model by running hyperparameter tuning.
- ✓ At last, I have predicted the used car price using the saved model. It was good!! that I was able to get the predictions near to actual values.

3. 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 find the correlation between the features. Thus, we can select the features which are correlated to each other and are independent. These feature sets were then given as an input to five algorithms and hyperparameter tuning 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 that the predicted and actual values were almost the same.

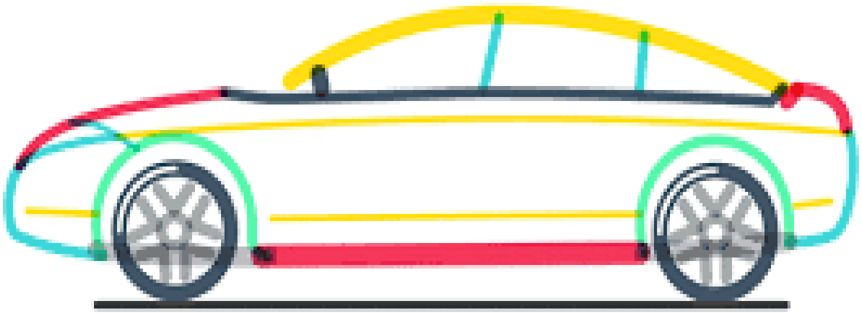
4.2 Learning Outcomes of the Study in respect of Data Science

I found that the dataset was quite interesting to handle as it contains all types of data in it and it was selfly scrapped frothy m 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 understand what data is trying to say. Data cleaning is one of the most important steps to removing 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 prices 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 prices. The 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, drawing back is scrapping the data as it is fluctuating process.
- ✓ Followed by more outliers and skewness these two will reduce our model accuracy.
- ✓ Also, we have tried our best to deal with outliers, skewness, and null values. So, it looks quite good that we have achieved an accuracy of 76.60% even after dealing with 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