PROJECT REPORT ON

"USED CAR PRICE PREDICTION"

Submitted By:

E.Sanjana

ACKNOWLEDGMENT

I would like to express my sincere thanks of gratitude to my SME as well as “Flip Robo Technologies” team for letting me work on “Used Car Price Prediction” project also huge thanks to my academic team “DataTrained”. 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.

Finally, I would like to thank my family and friends who have helped me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

INTRODUCTION

Predicting the price of used cars is an important and interesting problem. Predicting the resale value of a car is not a simple task. It is trite knowledge that the value of used cars depends on a number of factors. The most important ones are usually the age of the car, its make (model), the origin of the car (the original location of the manufacturer), its mileage (the number of kilometres it has run) and its horsepower (amount of power that an engine produces). Due to rising fuel prices, fuel economy is also of prime importance. Unfortunately, in practice, most people do not know exactly how much fuel their car consumes for each km driven. Other factors such as the type of fuel it uses, the interior style, the braking system, acceleration, engine displacement, the volume of its cylinders (measured in cc), its size, number of doors, paint colour, weight of the car, consumer reviews, prestigious awards won by the car manufacturer, its physical state, whether it is a sports car, whether it has cruise control, whether it is automatic or manual transmission, whether it belonged to an individual or a company and other options such as air conditioner, sound system, power steering, cosmic wheels, GPS navigator all may influence the price as well. Some special factors which buyers attach importance is the local of previous owners, whether the car had been involved in serious accidents. The look and feel of the car certainly contribute a lot to the price. As we can see, the price depends on a large number of factors. Unfortunately, information about all these factors are not always available and the buyer must make the decision to purchase at a certain price based on few factors only. In this work, we have considered only a small subset of the factors which are more important.

Business Problem Framing

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.

Business goal: The main aim of this project is to predict the price of used car based on various features. Machine Learning is a field of technology developing with immense abilities and applications in automating tasks. So, we will deploy an ML model for car selling price prediction and analysis. This kind of system becomes handy for many people. This model will provide the approximate selling price for the car based on different features like fuel type, transmission, price, weight, running in kms, length etc and this model will help the client to understand the price of used cars.

Conceptual Background of the Domain Problem

Car Price Prediction is really an interesting machine learning problem as there are many factors that influence the price of a car in the second-hand market. In many developed countries, it is common to lease a car rather than buying it outright. A lease is a binding contract between a buyer and a seller (or a third party – usually a bank, insurance firm or other financial institutions) in which the buyer must pay fixed installments for a pre-defined number of months/years to the seller/financer. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e., its expected resale value. Thus, it is of commercial interest to seller/financers to be able to predict the salvage value (residual value) of cars with accuracy. If the residual value is under-estimated by the seller/financer at the beginning, the installments will be higher for the clients who will certainly then opt for another seller/financer. If the residual value is over-estimated, the installments will be lower for the clients but then the seller/financer may have much difficulty at selling these high-priced used cars at this over-estimated residual value. Thus, we can see that estimating the price of used cars is of very high commercial importance as well.

Here we are trying to help the client works with small traders, who sell used cars to understand the price of the used cars by deploying machine learning models. These models would help the client/sellers to understand the used car market and accordingly they would be able to sell the used car in the market.

Review of Literature

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 require

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 values.

AnalyticalProblem Framing

Mathematical/ Analytical Modeling of the Problem

As a first step I have scrapped the required data from cardekho 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.

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 5655 rows and 17 columns including target. In this particular 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.

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 kms till the date
- Gear_transmission : Type of gear transmission used in car
- Seating_cap : Availability of number of seats in the car
- color : Car color
- 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 car in mm
- width : Width of 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 hours
- City_url : Url of the page of cars from a particular city
- Car_price : Price of the car

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 dropped Unnamed:0, unnamed:4,Max_power,engine displacement,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.

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.

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 to build the model we need some basic libraries as follows

```
#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

Data Analysis and Visualization

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.

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

Key Metrics for success in solving problem under consideration:

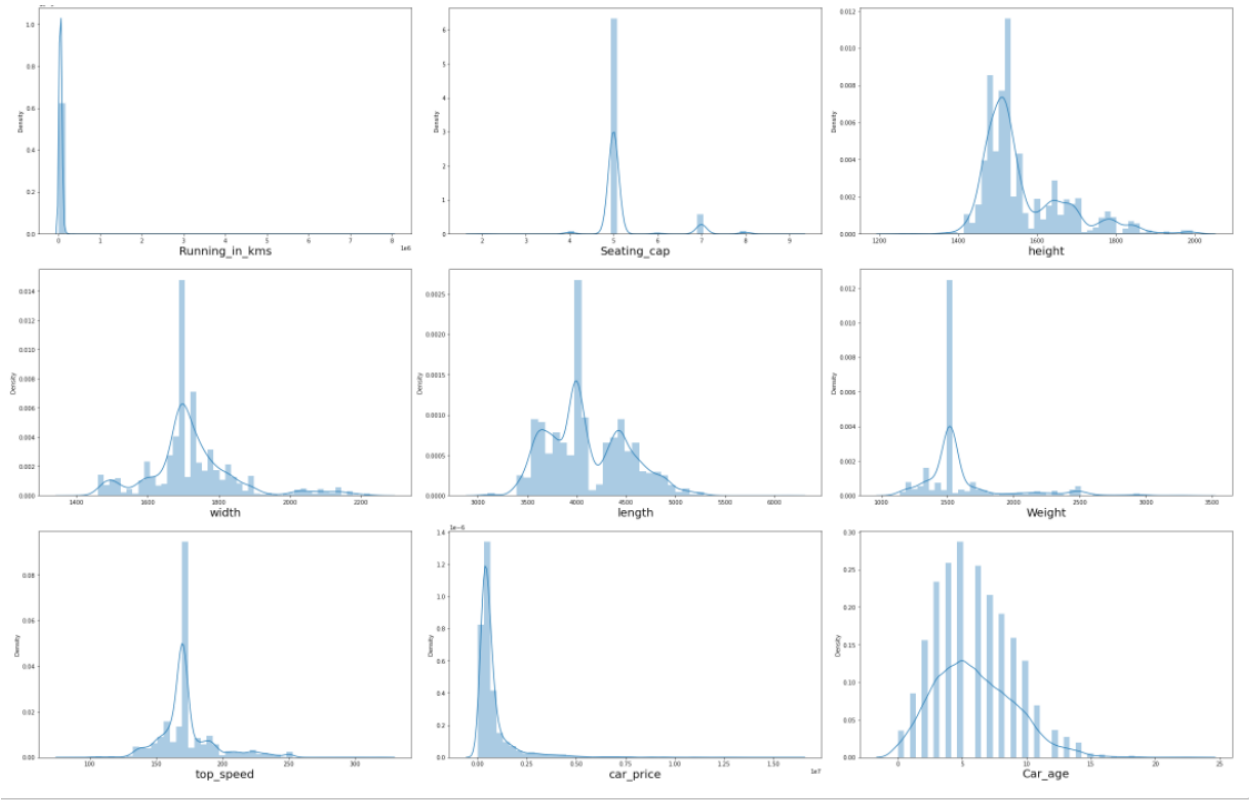
I have used the following metrics for evaluation:

- 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.

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

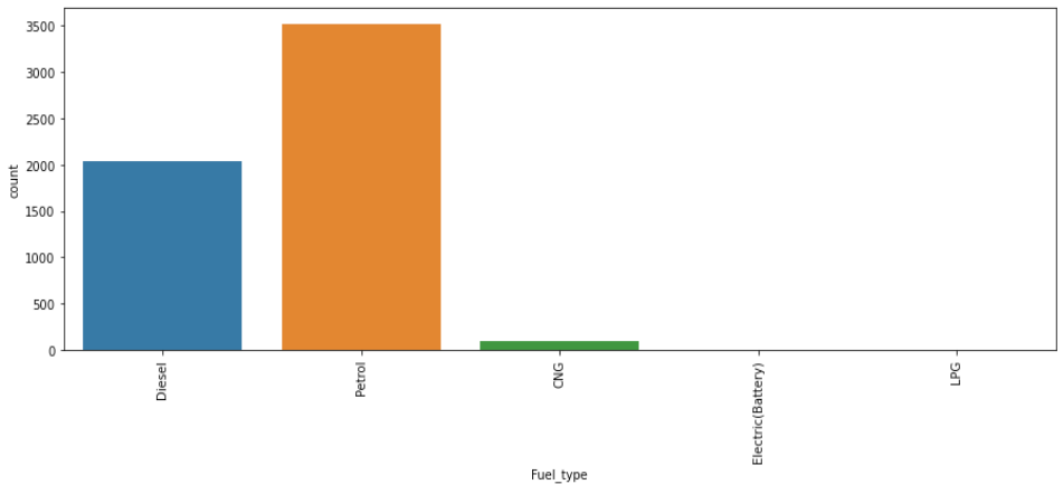
1) **Univariate Analysis for numerical columns**

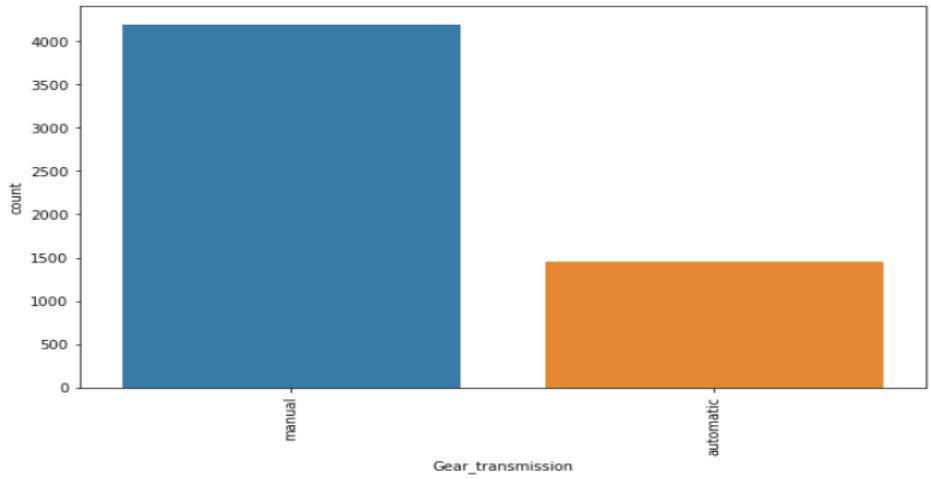


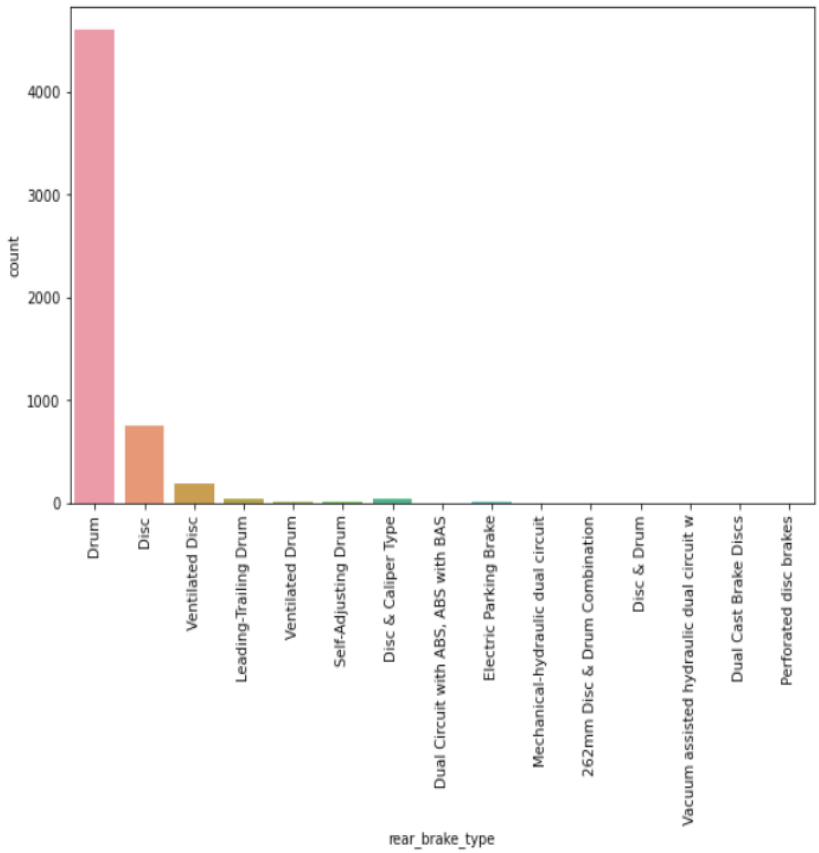
Observations:

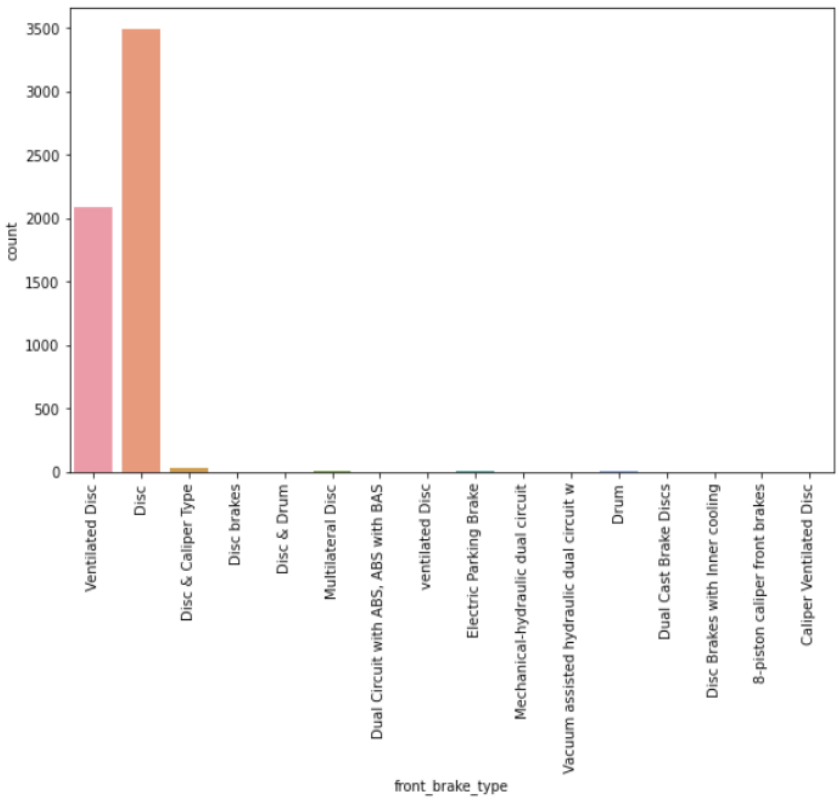
- ✓ We can clearly see that there is skewness in most of the columns so we have to treat them using suitable methods.

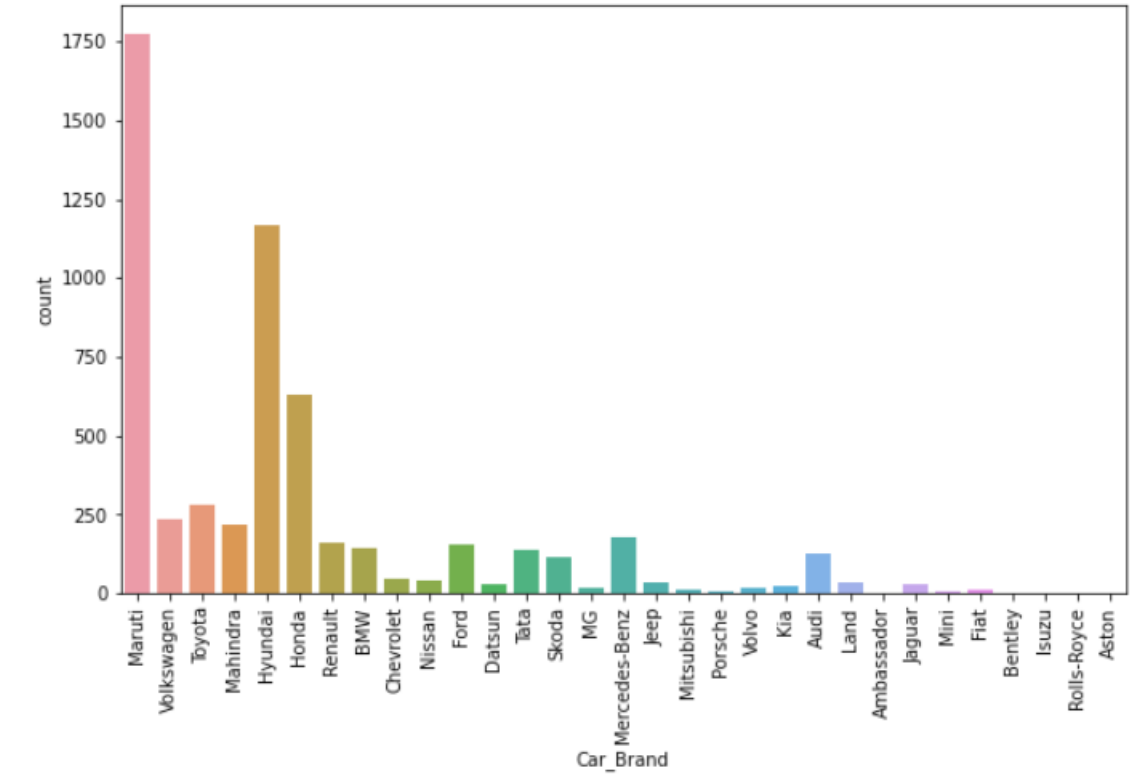
2) Univariate analysis for categorical column

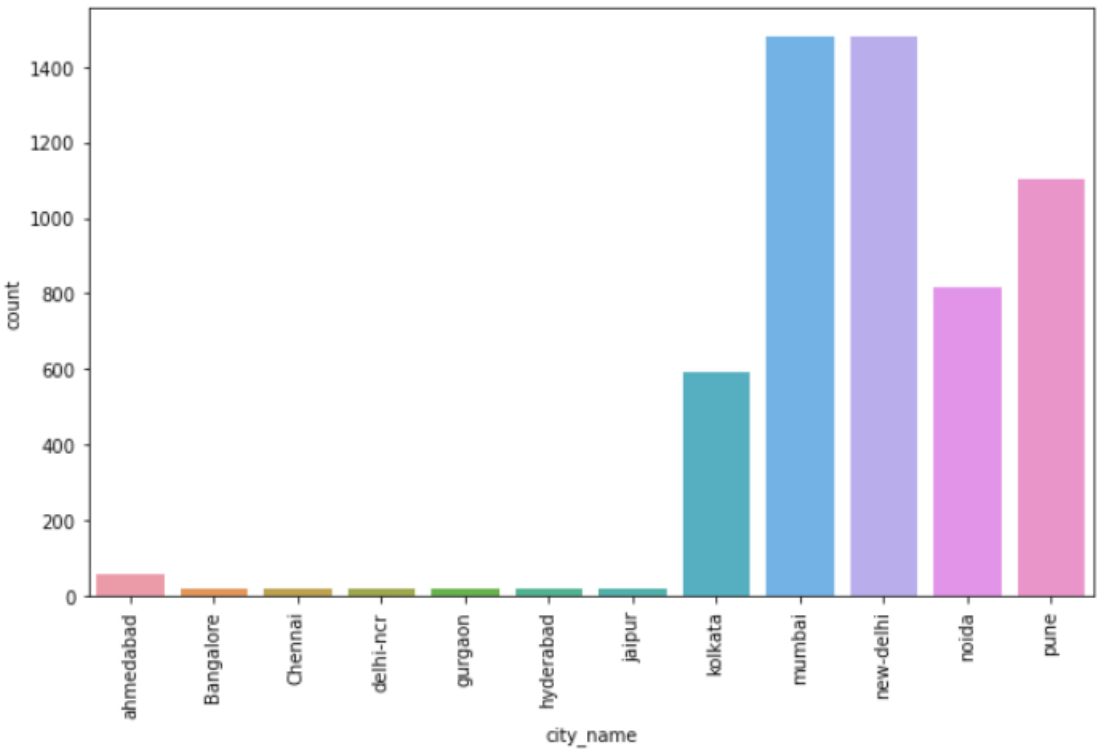



4000
3500
3000
2500
2000
1500
1000
500
0
count
manual
automatic
Gear_transmission


4000
3000
2000
1000
0
count
Drum
Disc
Ventilated Disc
Leading-Trailing Drum
Ventilated Drum
Self-Adjusting Drum
Disc & Caliper Type
Dual Circuit with ABS, ABS with BAS
Electric Parking Brake
Mechanical-hydraulic dual circuit
262mm Disc & Drum Combination
Disc & Drum
Vacuum assisted hydraulic dual circuit w
Dual Cast Brake Discs
Perforated disc brakes
rear_brake_type


3500
3000
2500
2000
1500
1000
500
0
count
Ventilated Disc
Disc
Disc & Caliper Type
Disc brakes
Disc & Drum
Multilateral Disc
Dual Circuit with ABS, ABS with BAS
ventilated Disc
Electric Parking Brake
Mechanical-hydraulic dual circuit
Vacuum assisted hydraulic dual circuit w
Drum
Dual Cast Brake Discs
Disc Brakes with Inner cooling
8-piston caliper front brakes
Caliper Ventilated Disc
front_brake_type

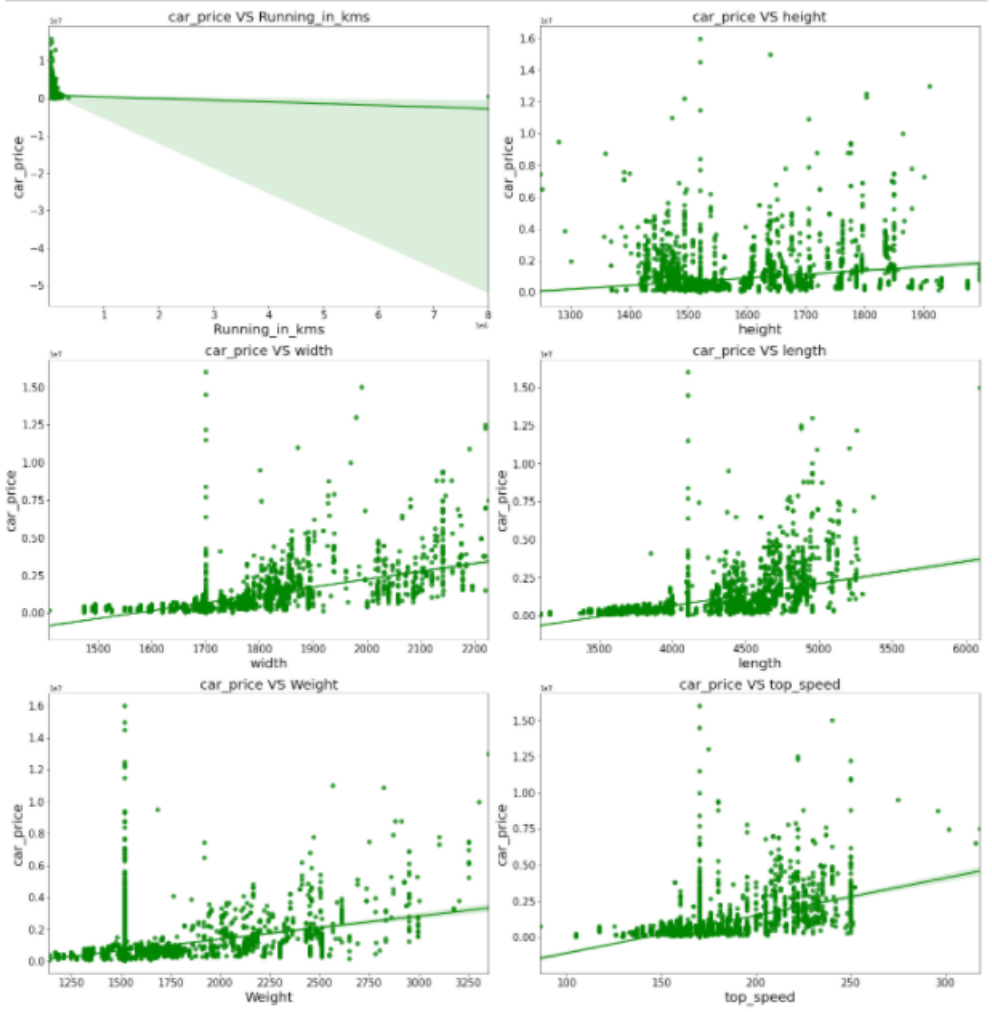

count
1750
1500
1250
1000
750
500
250
0
Maruti
Volkswagen
Toyota
Mahindra
Hyundai
Honda
Renault
BMW
Chevrolet
Nissan
Ford
Datsun
Tata
Skoda
MG
Mercedes-Benz
Jeep
Mitsubishi
Porsche
Volvo
Kia
Audi
Land
Ambassador
Jaguar
Mini
Fiat
Bentley
Isuzu
Rolls-Royce
Aston
Car_Brand

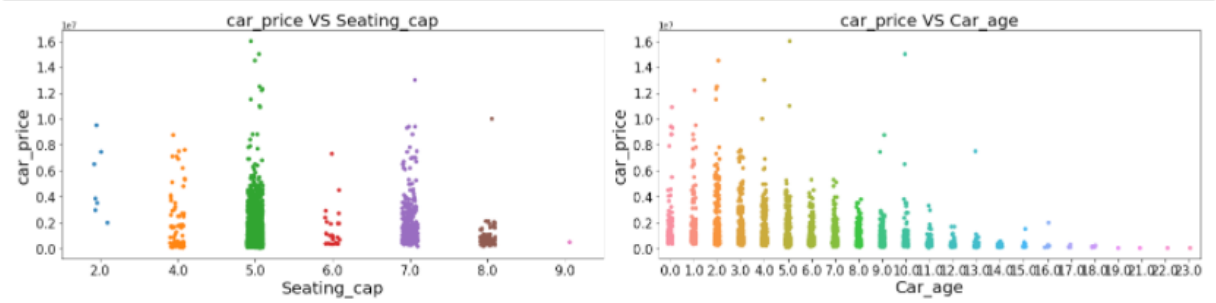

count
1400
1200
1000
800
600
400
200
0
ahmedabad
Bangalore
Chennai
delhi-ncr
gurgaon
hyderabad
jaipur
kolkata
mumbai
new-delhi
noida
pune
city_name

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 New-Delhi, mumbai and pune we can find maximum cars for sale. Since these are most populated places.

3) Bivariate analysis for numerical columns:

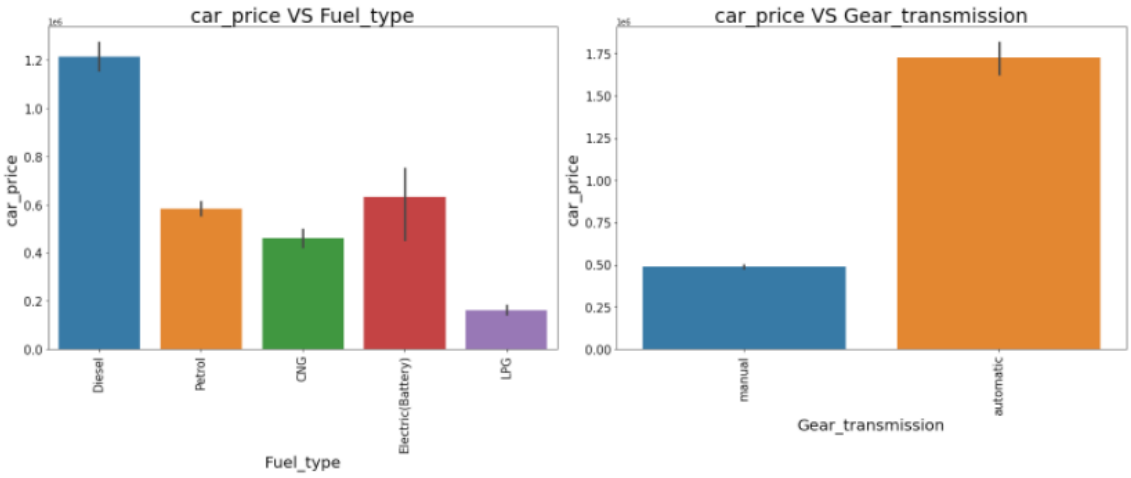


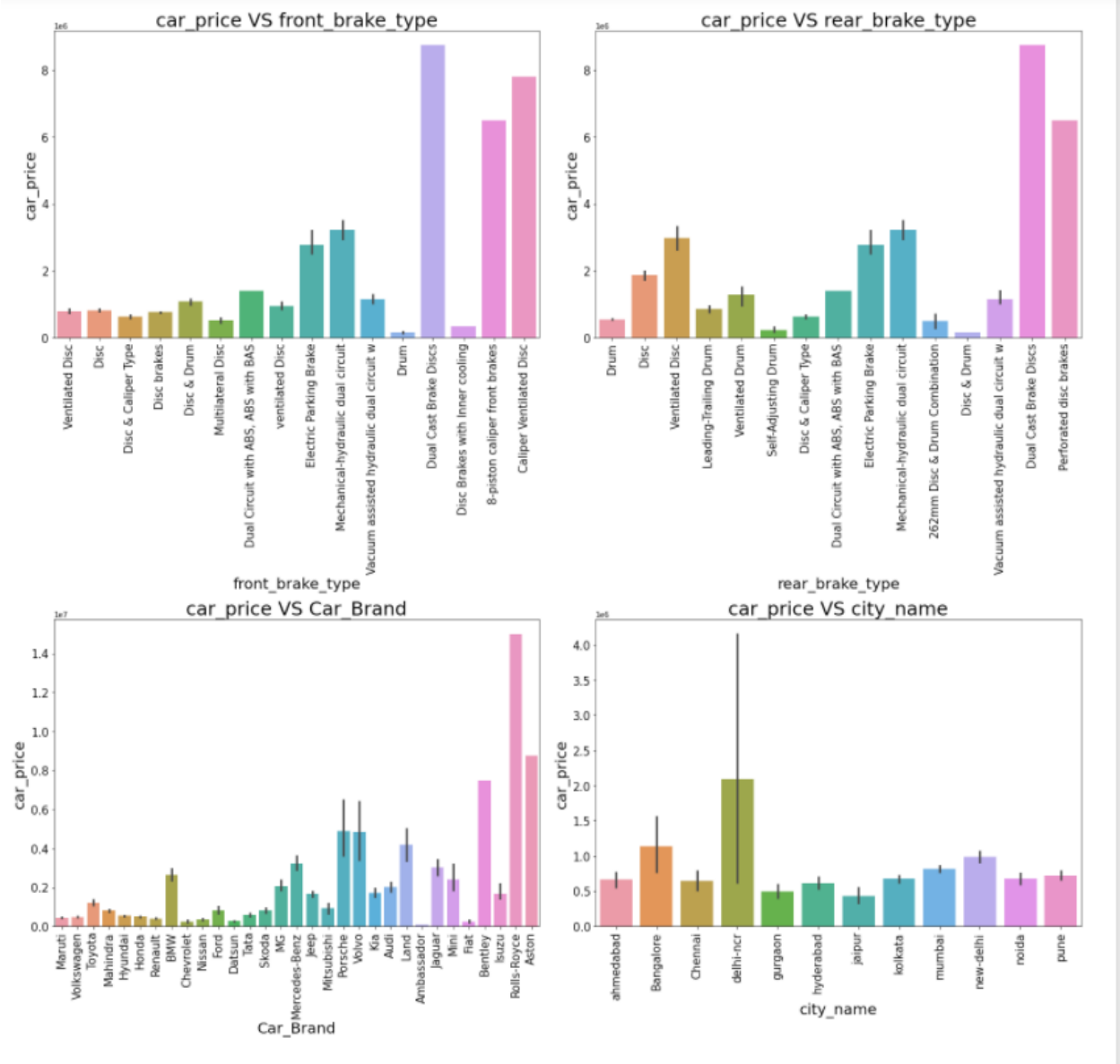


Observations:

- ✓ Maximum cars are having below 20k driven kms. And car price is high for less driven cars.
- ✓ 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.
- ✓ Cars with 5 and 7 seats are having highest price.
- ✓ As the age of the car increases the car price decreases.

4) Bivariate Analysis for categorical columns:





Observations:

- ✓ For Diesel and Electric cars the price is high compared to Petrol, LPG and CNG.
- ✓ Cars with automatic gear are costlier than manual gear cars.
- ✓ Cars with Dual cast break discs front break are costlier compared to other cars.
- ✓ Cars with Dual cast break discs rear break are costlier compared to other cars.
- ✓ Rolls Royce brand cars are having highest sale price.

- ✓ In Delhi-ncr, Bangalore, new delhi has the car prices are high as they are highly populated cities.

Run and Evaluate selected models

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: 92.9573500399289
mean_squared_error: 24458551607.866585
mean_absolute_error: 77588.56239182029
root_mean_squared_error: 156392.30034712894

Cross validation score : 80.47374098131851

R2_Score - Cross Validation Score : 12.483609058610384
```

- RandomForestRegressor has given me 92.96% r2_score and the difference between r2_score and cross validation score is 12.48%, but still we have to look into multiple models.

1) 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: 84.6486846034441
mean_squared_error: 53313872193.572556
mean_absolute_error: 83069.4404427249
root_mean_squared_error: 230897.9692279093

Cross validation score : 79.88731513757742

R2_Score - Cross Validation Score : 4.761369465866679
```

- XGBRegressor is giving me 84.64% r2_score and the difference between r2_score and cross validation score is 4.76%.

2) 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: 86.52989453999132
mean_squared_error: 46780582795.52797
mean_absolute_error: 105533.96417612548
root_mean_squared_error: 216288.19384221593

Cross validation score : 78.4270580311006

R2_Score - Cross Validation Score : 8.102836508890718
```

- GradientBoostingRegressor is giving me 86.52% r2_score and the difference between r2_score and cross validation score is 8.10%.

3) DecisionTreeRegressor:

```
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: 32.96950041982966
mean_squared_error: 232791483685.51382
mean_absolute_error: 113710.59345200255
root_mean_squared_error: 482484.6978770558

Cross validation score : 34.37136826892846

R2_Score - Cross Validation Score : -1.4018678490987995
```

- DecisionTreeRegressor is giving me 33% r2_score and the difference between r2_score and cross validation score is -1.40%.

4) BaggingRegressor:

```
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: 87.35905504068573
mean_squared_error: 43900975685.644844
mean_absolute_error: 88015.40916931554
root_mean_squared_error: 209525.59673138947

Cross validation score : 74.05388659445212

R2_Score - Cross Validation Score : 13.305168446233608
```

Bagging Regressor is giving me 87.40% r2_score.

- BaggingRegressor is giving me 87.35% r2_score and the difference between r2_score and cross validation score is 13.07%.

- **By looking into the difference of r2_score and cross validation score i found DecisionTreeRegressor as the best model and the difference between r2_score and cross validation score which is highly difference which is correct model to describe it.**

Hyper Parameter Tunning:

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

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

Giving DecisionTreeRegressor parameters.

```
GCV=GridSearchCV(DecisionTreeRegressor(),parameter,cv=10)
```

Running grid search CV for ExtraTreesRegressor.

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

```
GridSearchCV(cv=10, estimator=DecisionTreeRegressor(),
             param_grid={'criterion': ['squared_error', 'friedman_mse',
                                       'absolute_error', 'poisson'],
                         'max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
                                       13, 14, 15],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_split': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                                               11, 12, 13, 14, 15],
                         'splitter': ['best', 'random']})
```

Tunning the model using GCV.

```
GCV.best_params_
```

```
{'criterion': 'friedman_mse',
 'max_depth': 13,
 'max_features': 'auto',
 'min_samples_split': 4,
 'splitter': 'random'}
```

Got the best parameters for DecisionTreeRegressor.

```
Best_mod=DecisionTreeRegressor(criterion='friedman_mse',max_depth=15,max_features='auto',min_samples_split=4,splitter='random')
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: 69.83544620211455
mean_squared_error: 104759046662.37321
mean_absolute_error: 115348.0491877178
RMSE value: 323665.0223029563
```

- **I have choosed all parameters of DecisionTreeRegressor, after tunning the model with best parameters I got the 70% of it.**

Saving the model and Predictions:

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

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

- I have save the model as car_price.pkl
- Now loading my saved data and predicting the price values.

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

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

```
array([1211666.66666667,  298333.33333333,  371750.        , ...,
        427376.375     ,  574333.33333333,  529056.33333333])
```

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

	0	1	2	3	4	5	6	7	8	9	10	11
Predicted	1.211667e+06	298333.333333	371750.0	875125.0	827333.333333	556742.857143	495000.0	350100.0	145000.0	457500.0	2.019667e+06	509166.666667
Actual	1.435000e+06	345000.000000	382000.0	841000.0	850000.000000	715000.000000	495000.0	345000.0	125000.0	425000.0	1.950000e+06	525000.000000

```
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. Blue line is the actual line and red dots are the predicted values.

Interpretation of the Results

- The dataset was scrapped from cardekho website.
- The dataset was very challenging to handle it had 17 features with 5655 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 .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.

CONCLUSION

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.

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.

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 accuracy after tunning.
- 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