PROJECT REPORT -

Car Price Prediction

SUBMITTED BY:

Sumair Dhir

ACKNOWLEDGMENT

I would like to thank "Flip Robo" team, who has given me this opportunity to deal with this project as it helped me to improve my analyzation skills. And I want to express my huge gratitude to Ms. Khushboo Garg (SME Flip Robo) as she is the person who has helped me to get out of all the difficulties I faced while doing the project.

A big thanks to "Data trained" who is the reason behind this. Finally, my family who have been my backbone in every step of my life. And, thanks to other persons who has helped me directly or indirectly to complete the project.

Table of Contents

1.INTRODUCTION

1.1 Business Problem Framing

Car price prediction is quite interesting and popular problem. As per information 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 change in the price of a fuel. Different features like exterior colour, 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 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 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 its market value.

There are one of the biggest target groups 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 Literature Review

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 Analytical Modelling

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 perticular 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 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 tuned the best model and saved the best model. At last, I have predicted the car-price using saved model.

2.2 Data Sources

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 v ariable. In this dataset, I have object type of data which has been changed as per our analysis about the dataset. The information about features is as foll ows.

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
- Endine_disp : Engine displacement/engine CC
- Gear_transmission : Type of gear transmission used in car
- Milage_in_km/ltr : Overall milage of car in Km/ltr
- Seating_cap : Availability of number of seats in the car
- color : Car color
- Max_power : Maximum power of engine used in 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 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

2.3 Data Pre-processing

- 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, 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 useful 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 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 used: -

1. Processor — core i3
2. RAM — 8 GB
3. HDD — 500 GB

Softwares used: -

1.Anaconda

2. Windows 8.1

3. Python 3.8

Libraries required:-

To run the program and to 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 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.metrics import accuracy_score
- ✓ from sklearn.metrics import r2_score

- ✓ from sklearn.model_selection import train_test_split
- ✓ from sklearn.preprocessing import LabelEncoder
- ✓ from sklearn.preprocessing import StandardScaler
- ✓ from sklearn.ensemble import RandomForestRegressor
- ✓ from sklearn.tree import DecisionTreeRegressor
- ✓ from sklearn.svm import SVR
- ✓ from sklearn.linear_model import LinearRegression
- ✓ from sklearn.neighbors import KNeighborsRegressor as KNN
- ✓ from xgboost import XGBRegressor
- ✓ from sklearn.metrics import classification_report
- ✓ from sklearn.ensemble import GradientBoostingRegressor as GBR
- ✓ from sklearn.model_selection import cross_val_score as cvs
- ✓ from sklearn import metrics

With these libraries we can go ahead with our model building.

3.Data Analysis and Visualization

3.1 Identification of possible problem-solving 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 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
- SVR
- GradientBoostingRegressor
- DecisionTreeRegressor
- Linear Regression
- KNN Regressor

3.3 Key Metrics used

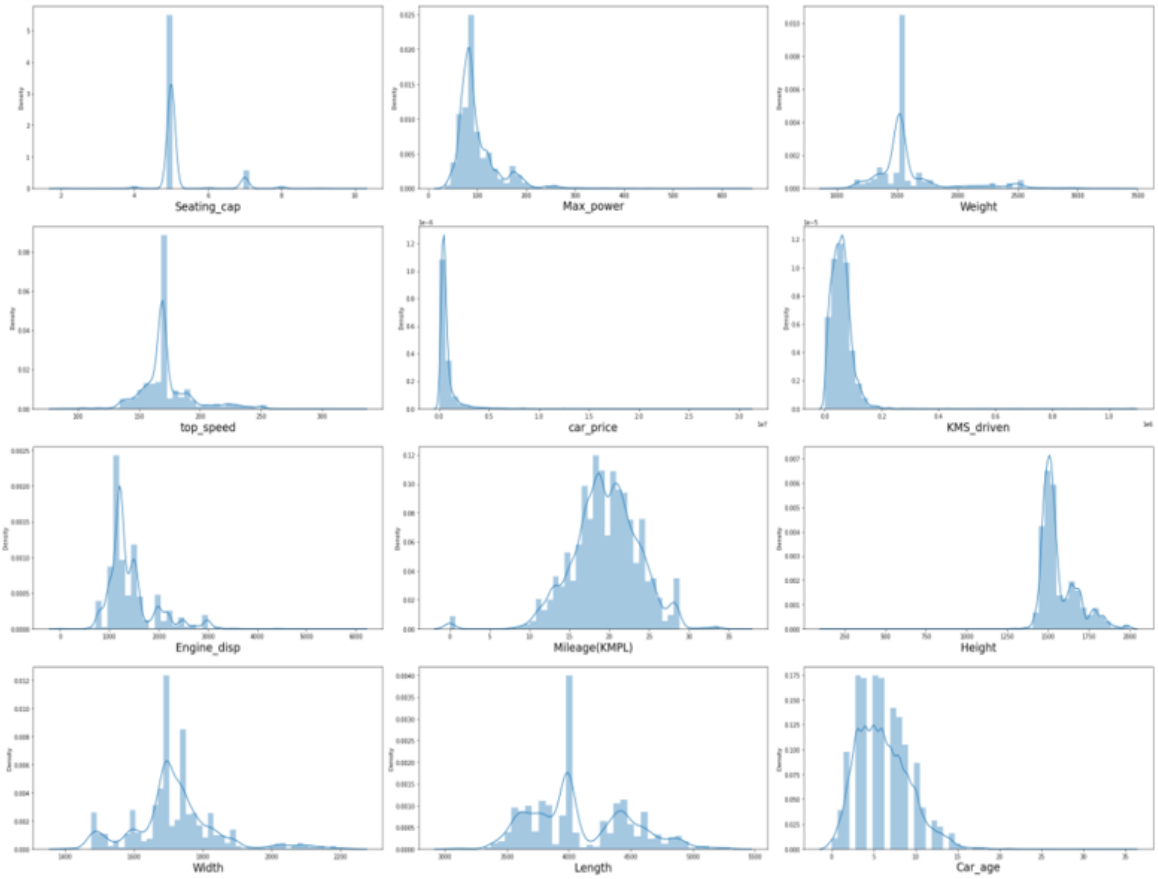
I have used the following metrics for evaluation:

- 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.

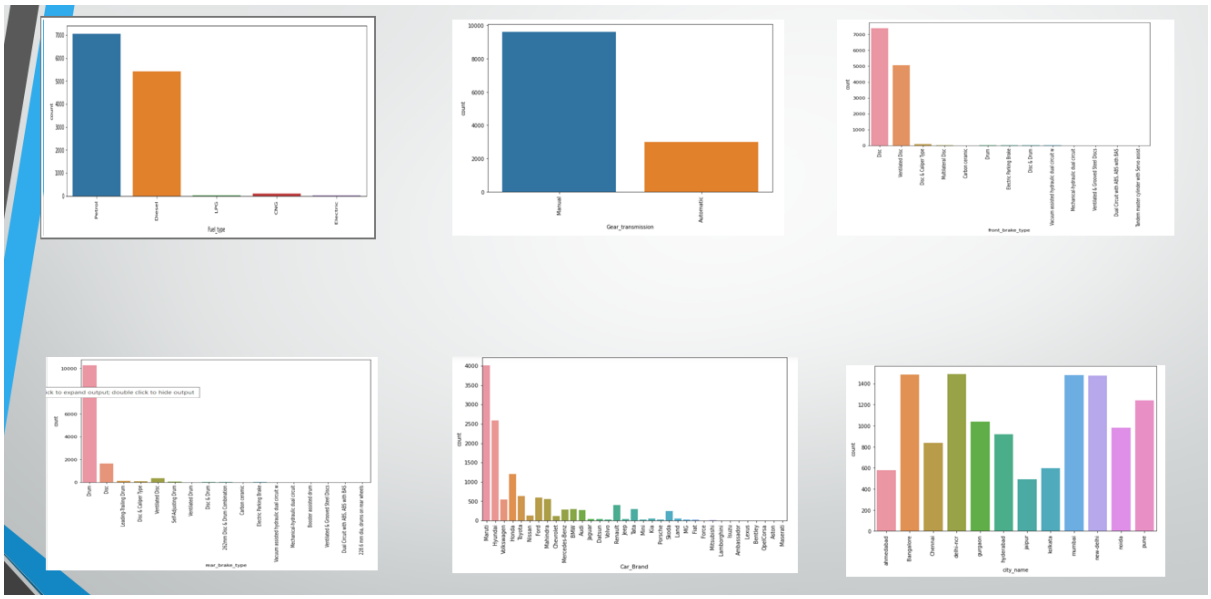
1. Univariate Analysis for numerical columns:



Observations:

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

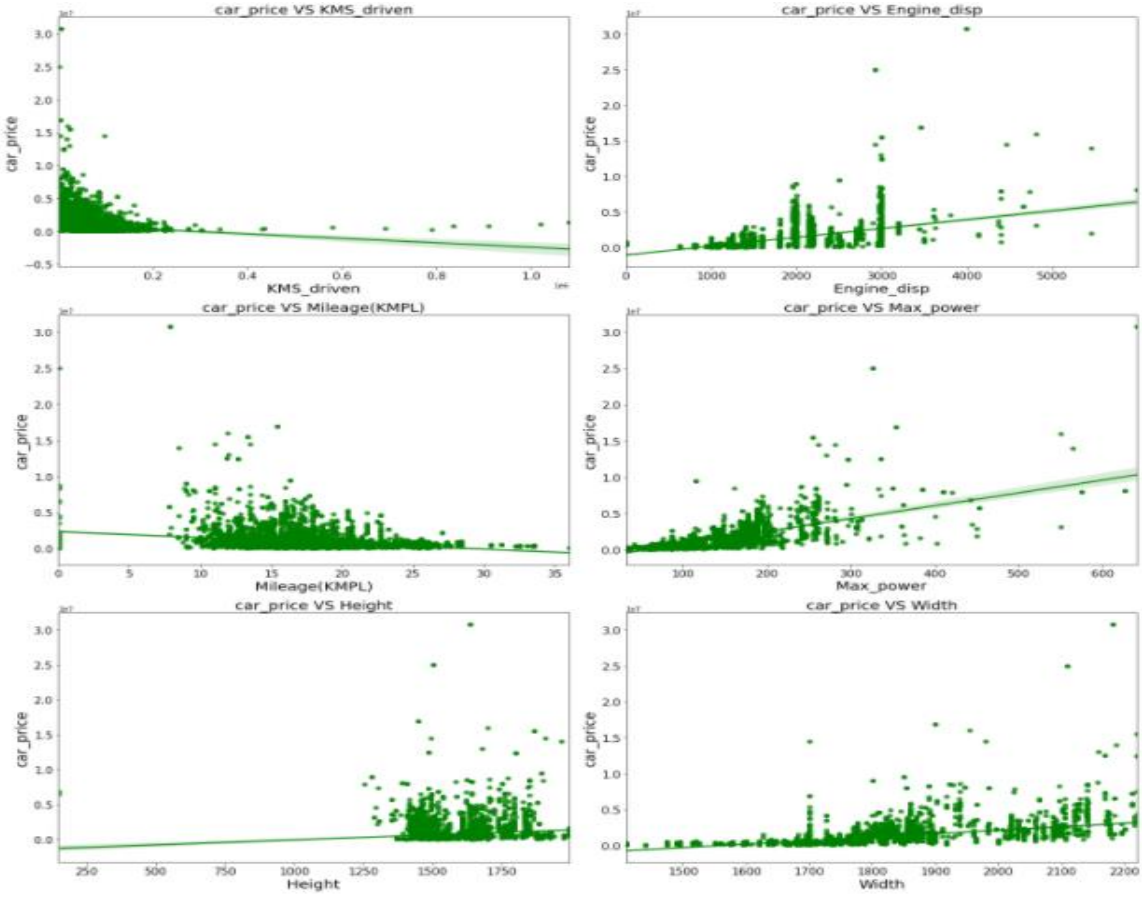
2. Univariate analysis for categorical column:

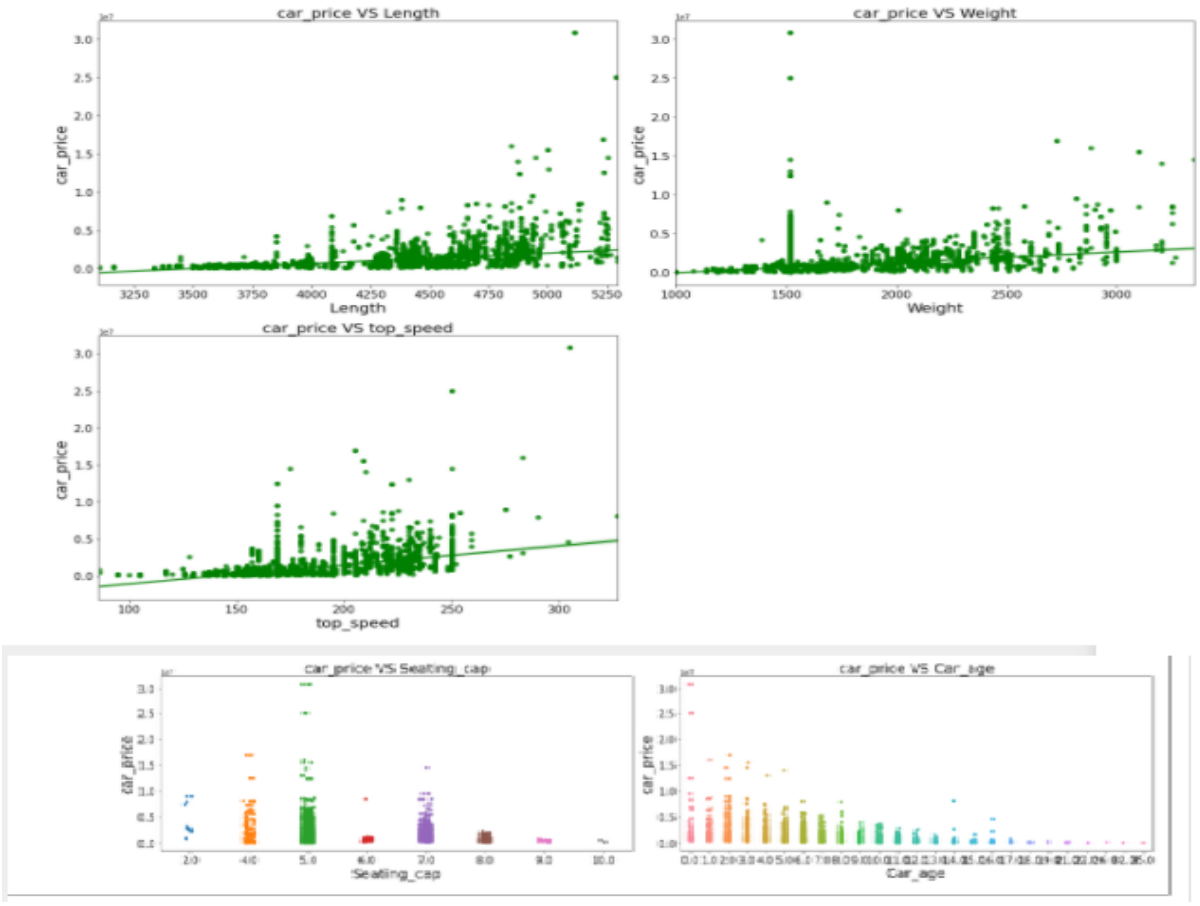


Observations:

- Maximum cars are petrol driven and 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.

3. Bivariate analysis for numerical columns:

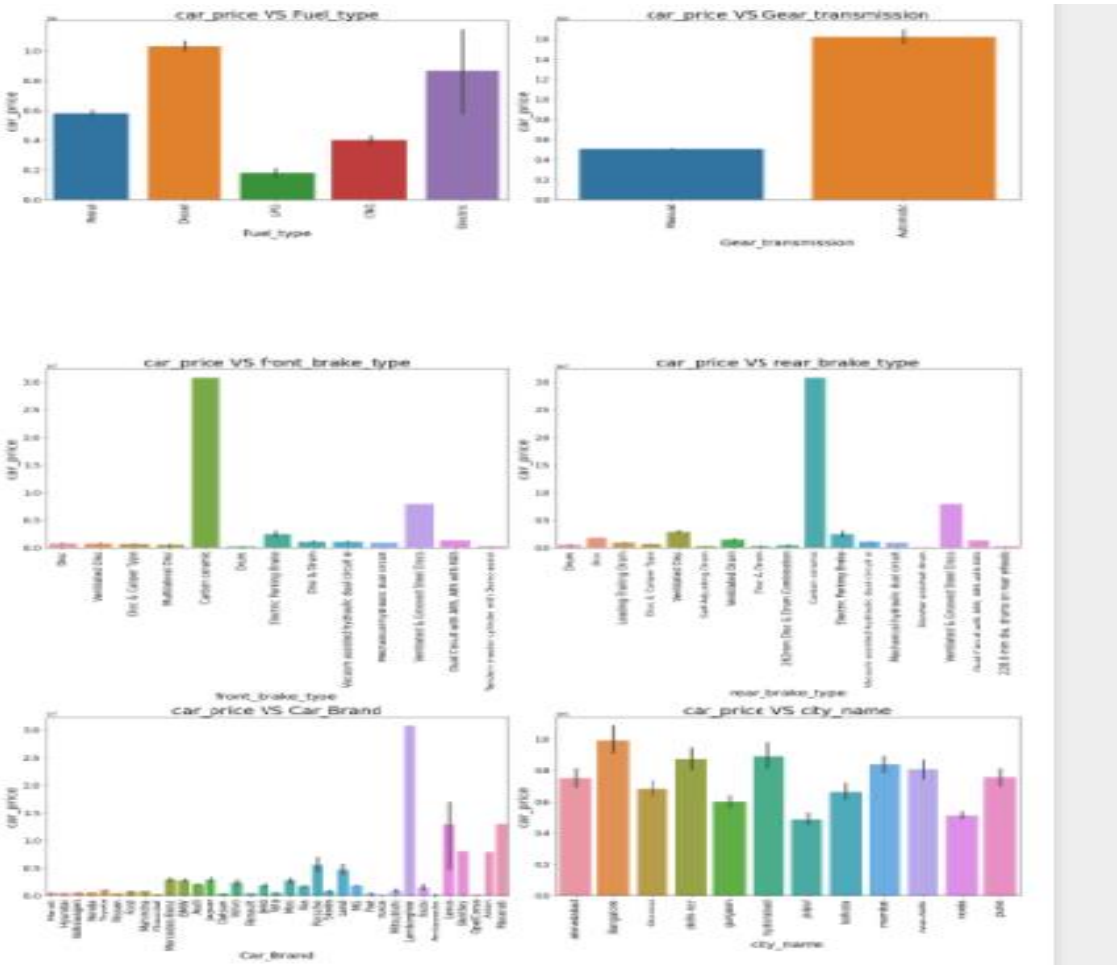

car_price VS KMS_driven
car_price
KMS_driven
car_price VS Engine_disp
car_price
Engine_disp
car_price VS Mileage(KMPL)
car_price
Mileage(KMPL)
car_price VS Max_power
car_price
Max_power
car_price VS Height
car_price
Height
car_price VS Width
car_price
Width



Observations:

- Most of the cars are less than 20k driven kms and car price is high for less driven cars.
- Maximum cars have 1000-3000 Engine_disp. and car price is high for 3000 Engine_disp.
- Majority of cars have mileage of 10-25 kms however, it 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.
- Cars with 4 and 5 seats are having highest price.
- As the age of the car increases the car price decreases.

4. Bivariate Analysis for categorical columns:



Observations:

- Diesel and Electric cars have high price compared to Petrol, LPG and CNG.
- Automatic gear cars are costlier than manual gear cars.
- Cars having carbon ceramic front break are costlier compared to other cars.
- Cars having carbon Ceramic rear break are costlier compared to other cars.
- Lamborghini brand cars are having highest sale price.
- In Bangalore, Hyderabad and Delhi-Ncr, the car prices are high as they are highly populated cities

3.5 Evaluation of selected models

1. Model Building:

1) RandomForestRegressor

```
RandomForestRegressor()  Results :

R2_score: 96.60108430654607
mean_squared_error: 8871009135.413982
mean_absolute_error: 50855.39521846707
root_mean_squared_error: 94186.03471541831

Cross validation score : 92.85744434721555

R2_Score - Cross Validation Score : 3.7436399593305225
```

- RandomForestRegressor has given me 96.60% r2_score and the difference between r2_score and cross validation score is 3.74.

2) Decision Tree Regressor

```
DecisionTreeRegressor()  Results :

R2_score: 92.38574678926244
mean_squared_error: 19872840600.870827
mean_absolute_error: 63509.86937590711
root_mean_squared_error: 140971.0629912069

Cross validation score : 88.72775566102244

R2_Score - Cross Validation Score : 3.657991128239999
```

- Decision Tree Regressor is giving me 92.38% r2_score and the difference between r2_score and cross validation score is 3.65%.

3) SVR

```
SVR()  Results :

R2_score: -6.013119700186476
mean_squared_error: 276689226256.8672
mean_absolute_error: 279724.72377090313
root_mean_squared_error: 526012.5723372657

Cross validation score : -7.618242030879903

R2_Score - Cross Validation Score : 1.605122330693427
```

- SVR is giving me -6.01% r2_score and the difference between r2_score and cross validation score is 1.60%.

4) Linear Regression

```
LinearRegression()  Results :

R2_score: 65.37792791225606
mean_squared_error: 90361969956.72887
mean_absolute_error: 179646.06388251757
root_mean_squared_error: 300602.677893476

Cross validation score : 59.86569456127147

R2_Score - Cross Validation Score : 5.512233350984587
```

- Linear Regression is giving me 65.37% r2_score and the difference between r2_score and cross validation score is 5.51%.

5) KNNRegressor

```
KNeighborsRegressor()  Results :

R2_score: 89.65922005059254
mean_squared_error: 26988946379.332367
mean_absolute_error: 86504.08127721335
root_mean_squared_error: 164283.12871178333

Cross validation score : 84.64325187065668

R2_Score - Cross Validation Score : 5.015968179935854
```

- KNN Regressor is giving me 89.65% r2_score and the difference between r2_score and cross validation score is 5.01%.

6) XGB Regressor

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
             gamma=0, gpu_id=-1, importance_type=None,
             interaction_constraints='', learning_rate=0.300000012,
             max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
             monotone_constraints='()', n_estimators=100, n_jobs=4,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method='exact',
             validate_parameters=1, verbosity=None)  Results :

R2_score: 96.88266001914756
mean_squared_error: 8136109848.676971
mean_absolute_error: 49944.77463035196
root_mean_squared_error: 90200.38718695709

Cross validation score : 93.3770758825645

R2_Score - Cross Validation Score : 3.5055841365830673
```

- XGB Regressor is giving me 96.88% r2_score and the difference between r2_score and cross validation score is 3.50%.

7) Gradient Boosting Regressor

```
GradientBoostingRegressor()  Results :

R2_score: 94.93679320038765
mean_squared_error: 13214730174.201113
mean_absolute_error: 71216.56262529005
root_mean_squared_error: 114955.33991164183

Cross validation score : 90.200796401679

R2_Score - Cross Validation Score : 4.735996798708655
```

- Gradient Boosting Regressor is giving me 94.93% r2_score and the difference between r2_score and cross validation score is 4.73%.

XGB Regressor was selected as best model after looking into the R2_score and cross validation score.

2. Hyper Parameter Tunning:

In [107]:
```
#importing necessary libraries
from sklearn.model_selection import GridSearchCV
```

In [108]:
```
params = { 'max_depth': [3,6,10],
           'learning_rate': [0.01, 0.05, 0.1],
           'n_estimators': [100, 500, 1000],
           'colsample_bytree': [0.3, 0.7]}
```

In [111]:
```
import xgboost
```

In [112]:
```
xgbr = xgboost.XGBRegressor(seed = 20)
clf = GridSearchCV(estimator=xgbr,
                   param_grid=params,
                   scoring='neg_mean_squared_error',
                   verbose=1)
```

In [114]:
```
clf.fit(X_train, y_train)
print("Best parameters:", clf.best_params_)
print("Lowest RMSE: ", (-clf.best_score_)**(1/2.0))
```

```
Fitting 5 folds for each of 54 candidates, totalling 270 fits
Best parameters: {'colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 1000}
Lowest RMSE:  111090.44540502375
```

In [115]:
```
Best_mod=XGBRegressor(colsample_bytree=0.7,learning_rate=0.1,max_depth=3,n_estimators=1000)
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: 97.16024210704445
mean_squared_error: 7411633733.454948
mean_absolute_error: 52582.854951696296
RMSE value: 86090.84581681695
```

After Hyper tuning, R2_Score increased from 96.88 to 97.16 and also RMSE got reduced which is good.

I have hypertuned the XGBRegressor and was able to increase the accuracy to 97.16%.

3. Saving the model and Predictions:

Best model was saved using .obj as follows.

```
In [117]: # Saving the model using .pkl
          import joblib
          joblib.dump(Best_mod,"Price.obj")

Out[117]: ['Price.obj']
```

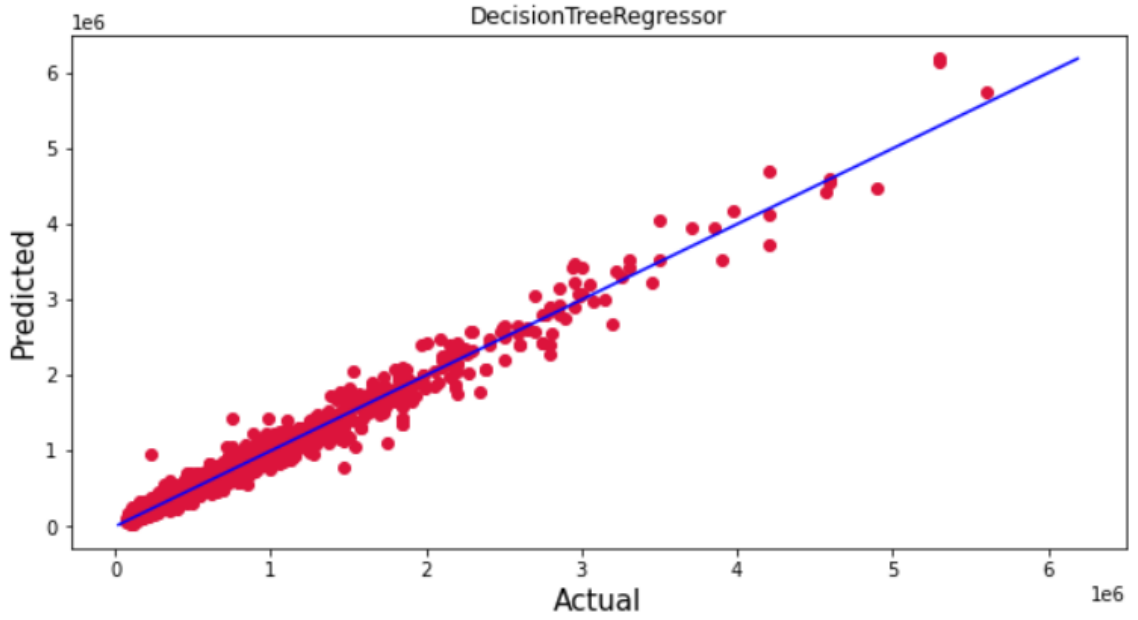
- In last, saved model was used to predict the price values.

Predictions:

```
In [118]: # Loading the saved model
          model=joblib.load("Price.obj")

          #Prediction
          prediction = model.predict(X_test)
          prediction

Out[118]: array([ 377009.34, 1603497.1 ,  426911.53, ...,  582244.9 ,  195468.89,
                  365565.66], dtype=float32)

In [119]: pd.DataFrame([model.predict(X_test)[:],y_test[:]],index=["Predicted","Actual"])

Out[119]:
```

	0	1	2	3	4	5	6	7	8	9	...	3435
Predicted	377009.34375	1603497.125	426911.53125	386714.03125	551353.8125	439637.96875	598355.625	598355.625	1568343.5	688819.8125	...	930666.625
Actual	379000.00000	1650000.000	465000.00000	435000.00000	550000.0000	450000.00000	550000.000	550000.000	1500000.0	643000.0000	...	725000.000

```
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

3.6 Result Interpretation

- Scaling the dataset has a good impact as it will help the model not to get baised. Since outliers and skewness were removed from the dataset so I have to choose Standardisation.
- Multiple models were used 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 XGBRegressor as the best model with 97.16% r2_score. Also, I have improved the accuracy of the best model by running hyper parameter tunning.
- Finally, 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, I have used Machine learning algorithms to predict the second-hand car price. I have mentioned the step-by-step procedure to analyse the dataset and finding the correlation between the independent variables and target variable. Hence, the features which are correlated to each other and are independent in nature are selected. With the help of visualization and its graphical representation, I was able to understand what data is trying to say. Data cleaning is one of the most important steps to remove unrealistic and unnecessary values. The data was done divided into training and testing set which were then given as an input to various algorithms and then, the best model was selected, and hyper parameter tunning was done to improve its accuracy. Hence, we calculated the performance of each model using different performance metrics and compared them based on these metrics. Finally, the best model was saved, and the car price was predicted. And the predicted and actual values were almost same.

4.2 Learning Outcomes of the Study in respect of Data Science

The dataset was quite interesting to handle as it contains all types of data in it, and it was scrapped by myself from cardekho.com 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 second-hand car price research.

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

4.3 Limitations of this work and Scope for Future Work

- Scrapping the data is fluctuating process which is one of the limitations.
- Greater number of outliers and skewness will reduce the model accuracy.
- I have tried best to deal with outliers, skewness, and null values. Hence, I was able to achieve an accuracy of 97.16% even after dealing with all these drawbacks.
- This project only covers the chosen algorithm, so it will not cover all Regression algorithms, starting from the basic ensembling techniques to the advanced ones.