FLIP ROBO

PROJECT REPORT

ON

Used-Car Price Prediction Project

SUBMITTED BY

V Samyukta

ACKNOWLEDGMENT

It gives me immense pleasure to deliver this report. Working on this project was a great learning experience that helped me attain in-depth knowledge on data analysis process.

Flip Robo Technologies (Bangalore) provided all of the necessary information and datasets, required for the completion of the project.

I express my gratitude to my SME, Ms. Khushboo Garg, for providing the dataset and directions for carrying out the project report procedure.

My heartfelt gratitude to DataTrained institute and FlipRobo company for providing me this internship opportunity. Last but not least to my sincere thanks to my family and all those who helped me directly or indirectly in completion this project.

CONTENTS

1.INTRODUCTION

- **Business Problem Framing:**

Deciding whether a used car is worth the posted price when you see listings online can be difficult. Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a used car appropriately [2-3]. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.

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

- **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.

- **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 every year. 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.

- **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

- **Mathematical / Analytical Modelling of the Problem:**
  As a first step I have scrapped the required data from 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 no null values in the dataset.
  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 checked for outliers and skewness in the dataset. I removed outliers using IQR method and skewness is not prominent in the dataset. 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.

- **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 3210 rows and 10 columns including target. In this particular dataset I have object type of data which has been changed as per our analysis about the d ataset. The information about features is as follows.

Features Information:

- Car_Brand : Name of the car with Year
- Make_Year : Year of manufracture
- Fuel_Type : Type of fuel used for car engine
- Kms_Driven : Car running in kms till the date
- Engine_Displacement: Engine displacement/engine CC
- Transmission : Type of gear transmission used in car
- Car_price : Price of the car
- Registration_Year : Year of car registration
- No_of_Owners : Number of owners of the car
- Insurance_Type : Type of insurance available to the car

- **Data Pre-processing 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 have also dropped Unnamed:0, and Resigstration_Year 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 have extracted useful 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.

- **Hardware and Software Requirements and Tools Used:**

  Hardware technology being Used:-
  - CPU: HP Pavilion
  - Chip: intel core13 8th Gen
  - RAM: 8 GB

  Software Technology being Used:-

  - Programming language: Python
  - Distribution: Anaconda Navigator
  - Browser based language shell: Jupyter Notebook

  Libraries/Packages Used:-

  Pandas, NumPy, matplotlib, seaborn, scikit-learn and pandas_profiling.

3.DATA ANALYSIS AND VISUALIZATION

- **Identification of possible problem-solving approaches (methods):**

  - To tackle the problem, I employed both statistical and analytical methodologies, which mostly included data pre-processing and EDA to examine the connection of independent and dependent characteristics. In addition, before feeding the input data into the machine learning models, I made sure that it was cleaned and scaled. We need to anticipate the car price of the used cars for this project, which implies o ur goal column is continuous, making this a regression challenge. I evaluated the prediction using a variety of regression methods. After a series of assessments, I determined that XGB Regressor is the best method for our final model since it has the best r2-score and the smallest difference in r2-score and CV-score of all the algorithms tested. Other regression methods are similarly accurate.
  - I used K-Fold cross validation to gain high performance and accuracy. Then hyper parameter tweaked the final model.
  - Once I had my desired final model, I made sure to save it before loading the testing data and beginning to do data pre-processing as the training dataset and retrieving the anticipated selling price values from the Regression Machine Learning Model.

- **Testing of Identified Approaches (Algorithms):**

Since Car_price is the target and is a continuous column so given problem is 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 XGB Regressor 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 to go through cross validation. Below is the list of regression algorithms I have used in my project.

- RandomForestRegressor
- XGBRegressor
- ExtraTreesRegressor
- GradientBoostingRegressor
- DecisionTreeRegressor

- **Key Metrics for success in solving problem under consideration:**

  r2 score, cross val score, MAE, MSE, and RMSE were the main metrics employed in this study. We used Hyperparameter Tuning to identify the optimal parameters and to improve our results, and we'll be utilising the GridSearchCV technique to do it.

  - Cross Validation:

  Cross-validation aids in determining the model's overfitting and underfitting. The model is constructed to run on several subsets of the dataset in cross validation, resulting in numerous measurements of the model. If we fold the data five times, it

will be separated into five parts, each representing 20% of the whole dataset. During the Cross-validation, the first part (20%) of the 5 parts will be left out as a holdout set for validation, while the rest of the data will be utilised for training. We'll acquire the initial estimate of the dataset's model quality this way.

Further rounds are produced in the same way for the second 20% of the dataset, which is kept as a holdout set while the remaining four portions are utilised for training data during the process. We'll acquire the second estimate of the dataset's model quality this way. During the cross-validation procedure, these stages are repeated to obtain the remaining estimate of model quality.

- R2 Score:

It is a statistical metric that indicates the regression model's quality of fit. The optimal r-square value is 1. The closer the r-square value is to 1, the better the model fits.

- Mean Squared Error:

The average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated is measured by the MSE of an estimator (of a process for estimating an unobserved variable). MSE is a risk function that represents the squared error loss's anticipated value. The Root Mean Squared Error is abbreviated as RMSE.

- Mean Absolute Error:

MAE is a statistic that assesses the average magnitude of mistakes in a set of forecasts without taking into account their direction. It's the average of the absolute differences between forecast and actual observation over the test sample, where all individual deviations are given equal weight.

- Hyperparameter Tunning:

There is a list of several machine learning models available. They're all distinct in some manner, yet the only thing that distinguishes them is the model's input parameters. Hyperparameters are the name given to these input parameters. These hyperparameters will establish the model's architecture, and the greatest thing is that you get to choose the ones you want for your model. Because the list of hyperparameters for each model differs, you must choose from a distinct list for each model.

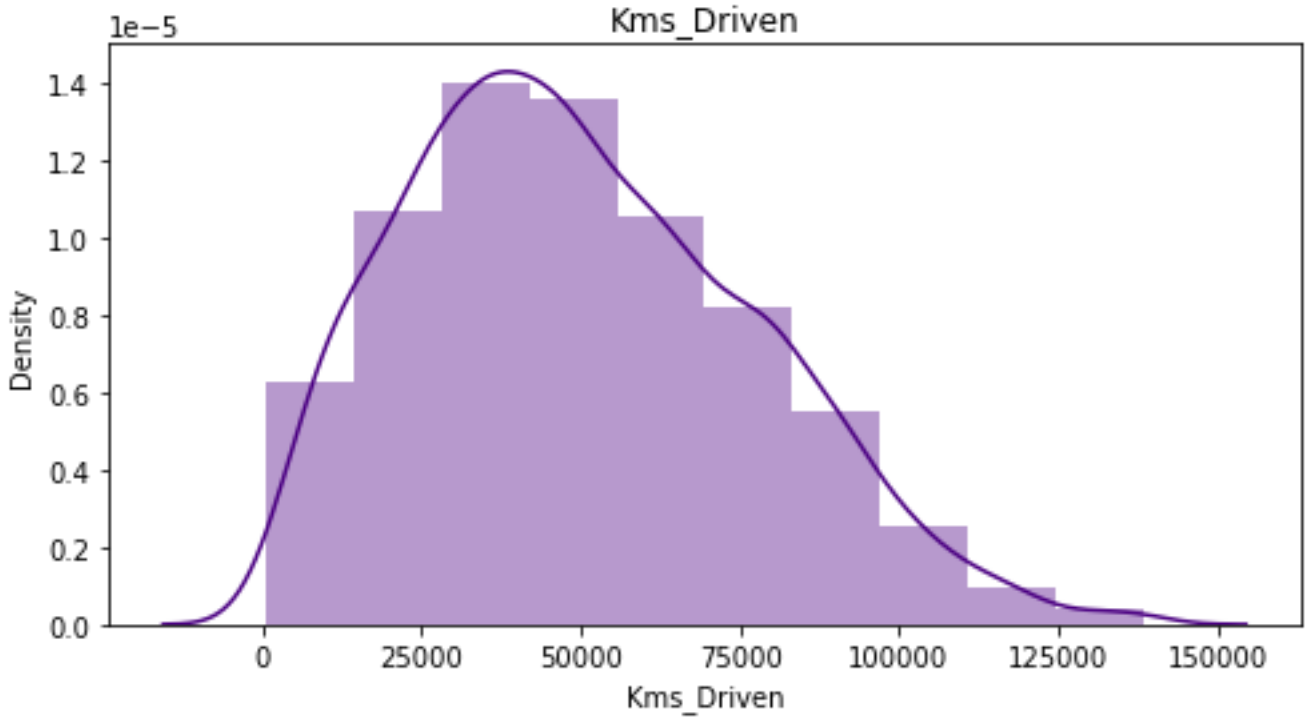
We are unaware of the ideal hyperparameter settings that would produce the best model output. So we instruct the model to automatically explore and choose the best model architecture. Hyperparameter tuning is the term for the method of selecting hyperparameters. GridSearchCV may be used to tune the system.

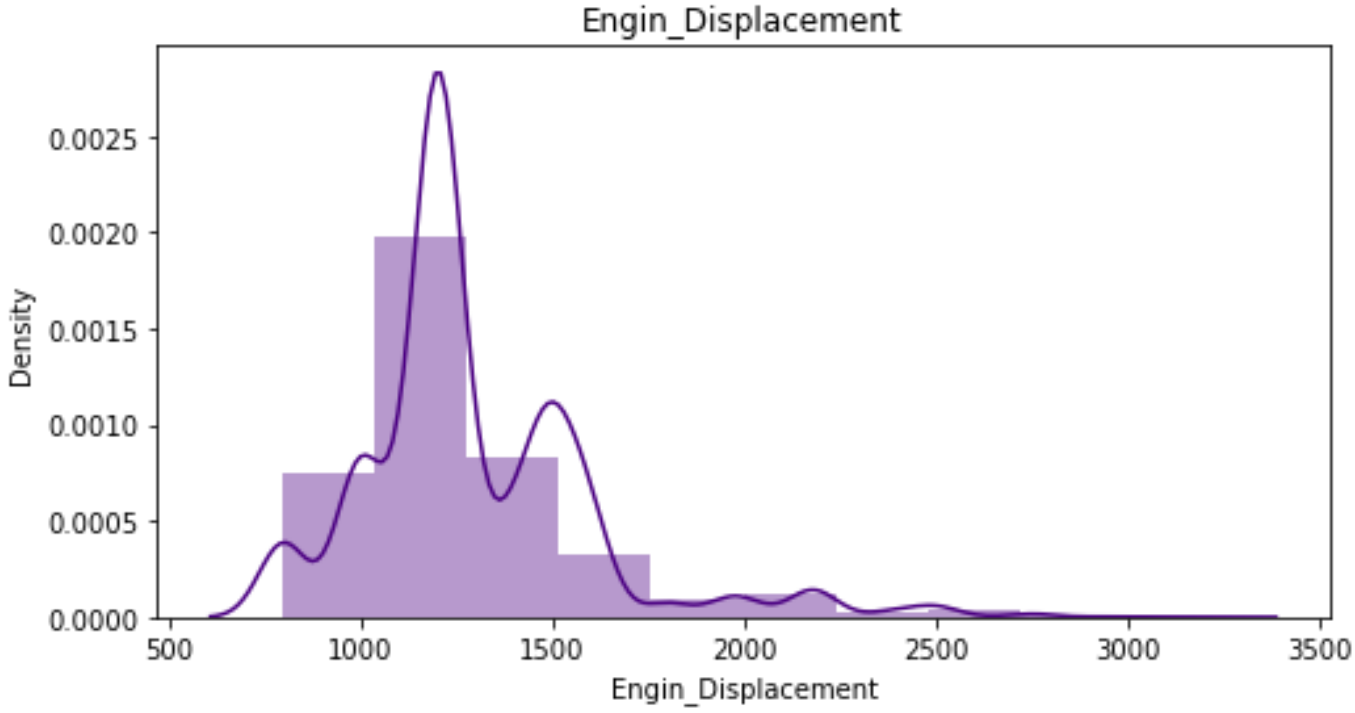
GridSearchCV is a model selection function in the Scikit-learn (or SK-learn) package. It is vital to remember that the Scikit-learn library must be installed on the PC. This function aids in fitting your estimator (model) to your training set by looping over specified hyperparameters. Finally, we may choose the optimal settings from the hyperparameters presented.

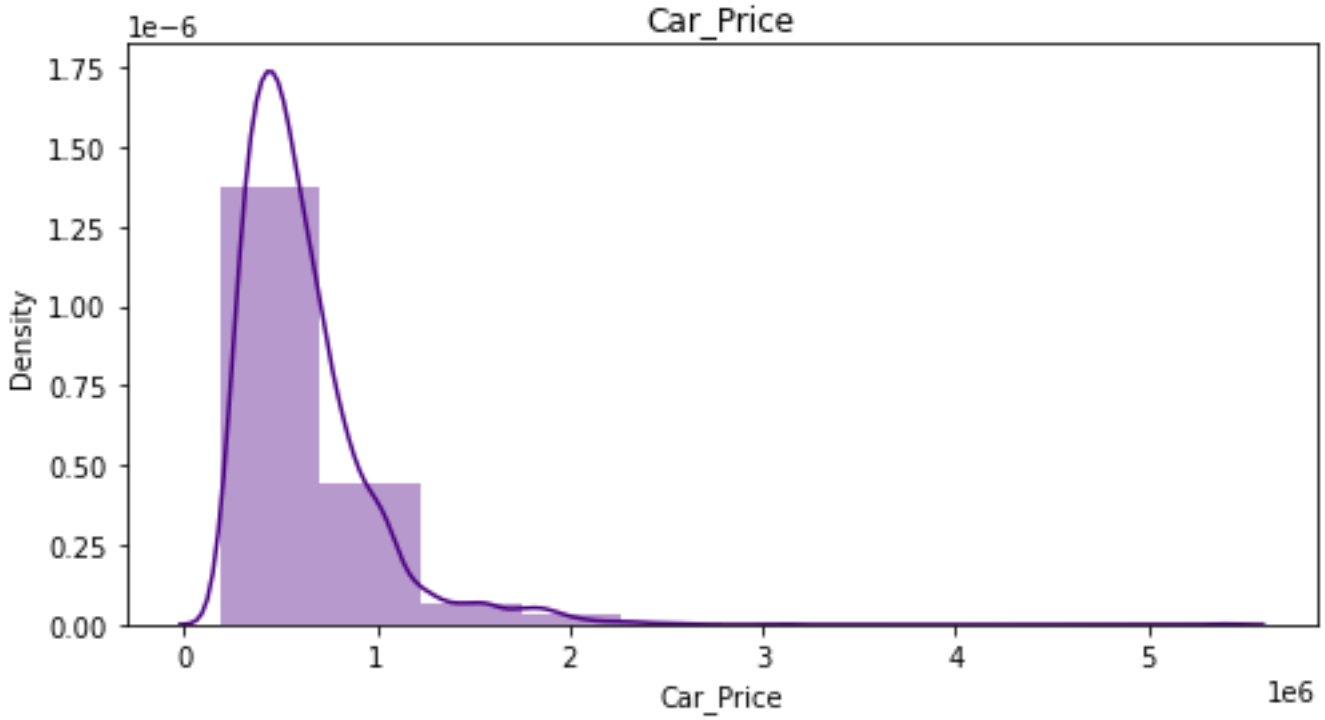
- **Visualization:**

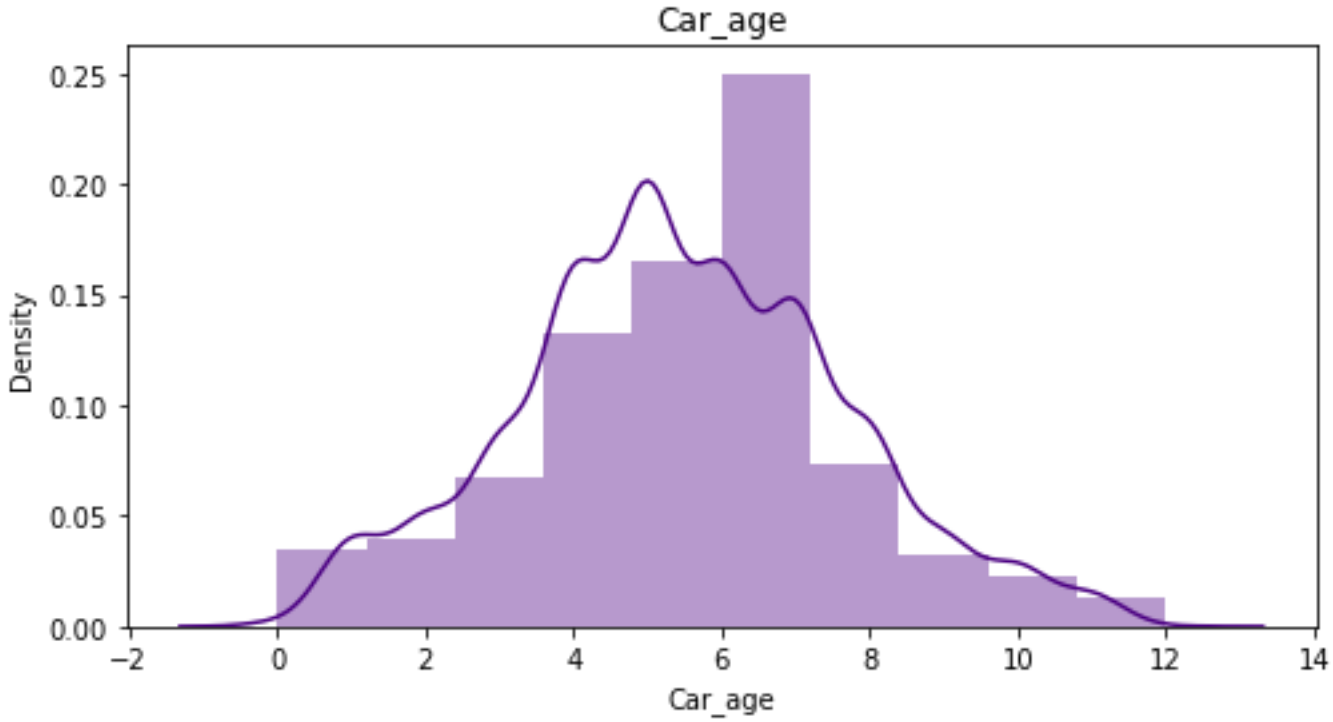
  I have used bar plots to see the relation of categorical feature and I have used dist plots for numerical features.

  - **<u>Visualization of numerical features with target:</u>**





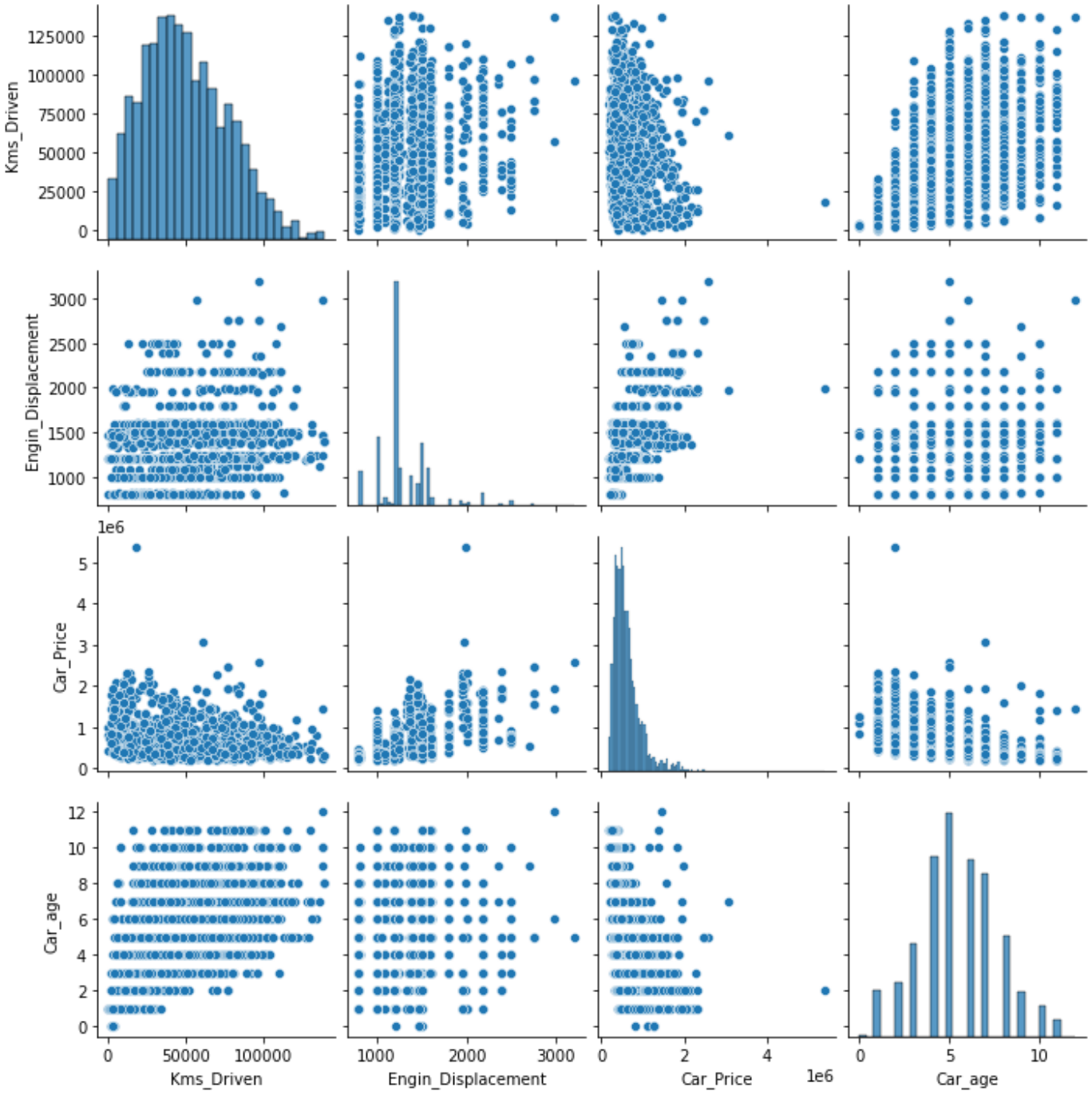




Observations:

- Engine_Displacement is having wide spread of data between 500-3500 and data is right skewed.
  Most of the cars have engine displacement between 1000-1300.
- and car_price is also right skewed. however, car_price skewness will not be removed as it is my target column.
- Most of used car are 7 year old.

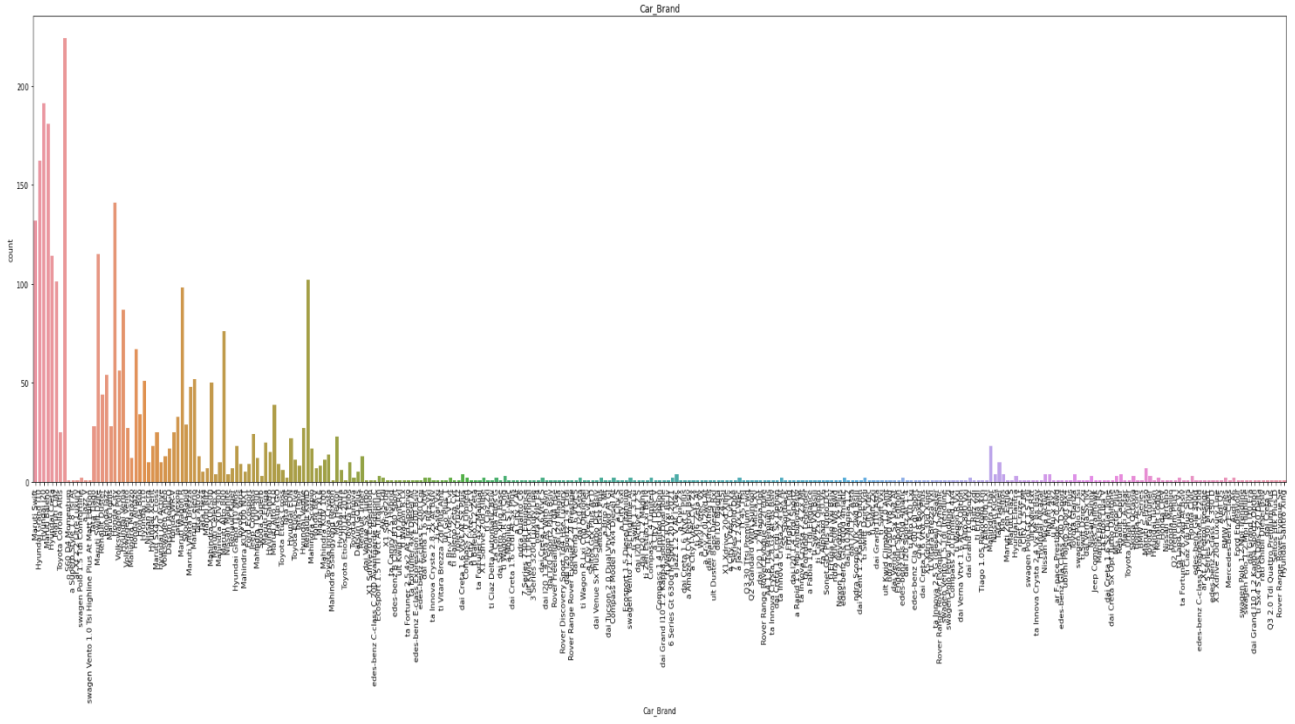
- Kms_Driven has wide spread of data between 20000-150000. Most of the car are driven between
  25000 – 75000 kms.
  PairPlot

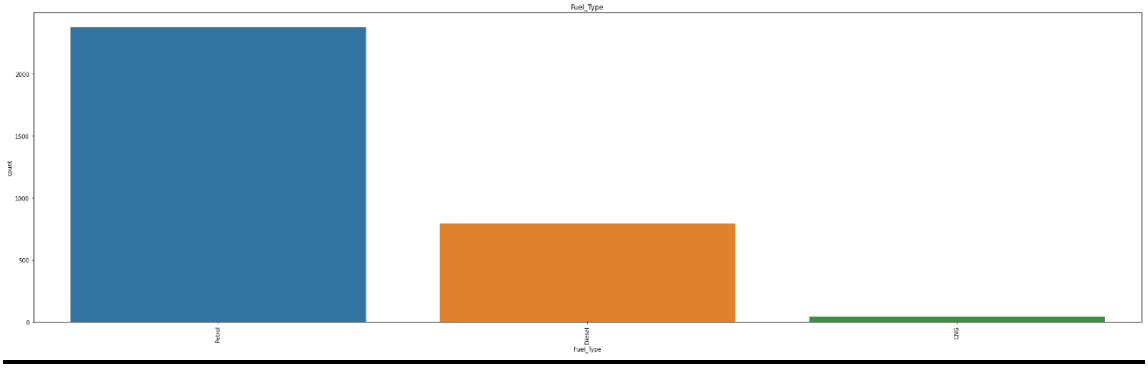


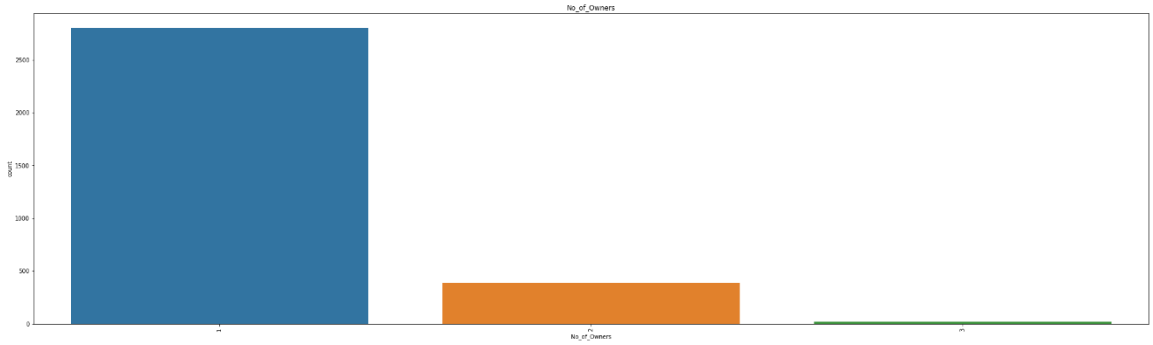
Observations:

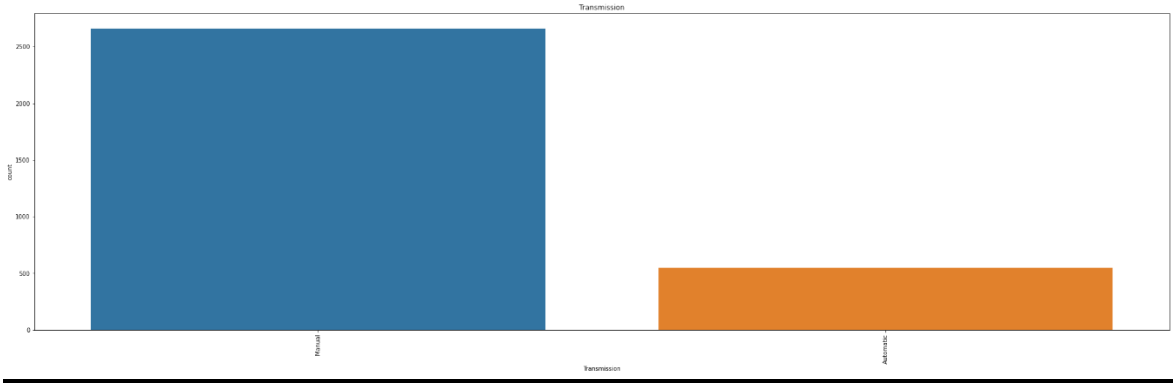
Pairplot doesn't show any linear relationship among columns

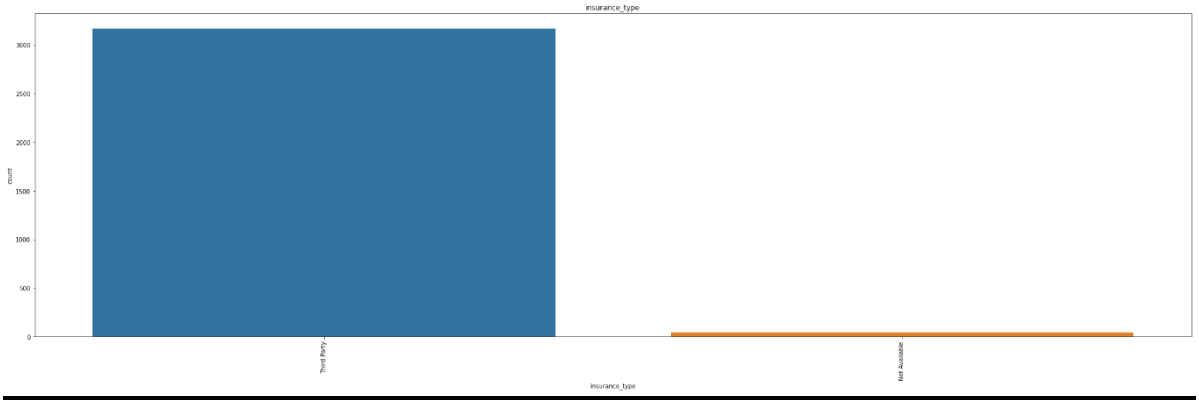
2. Visualization of categorical features with target:

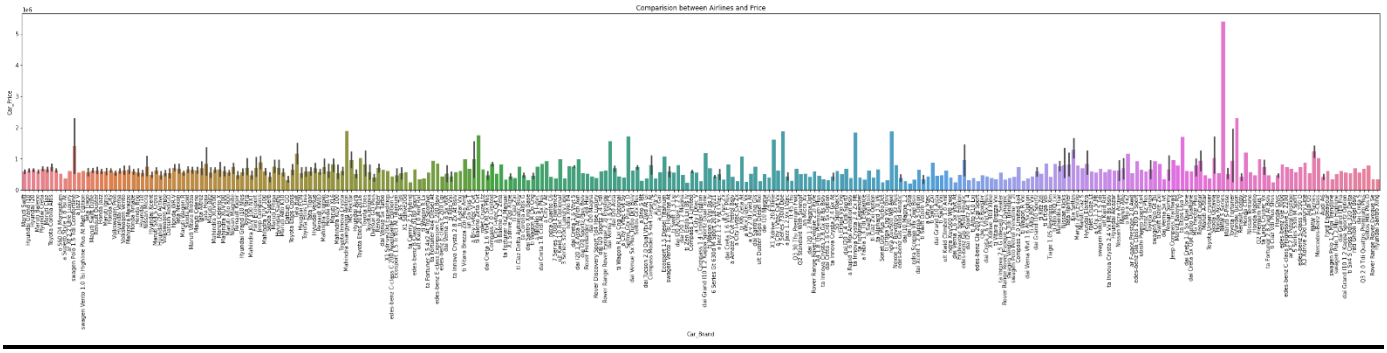








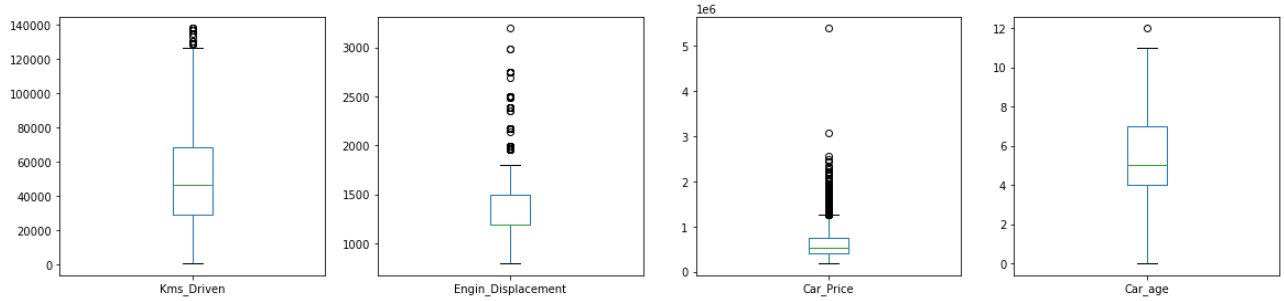




Observations:

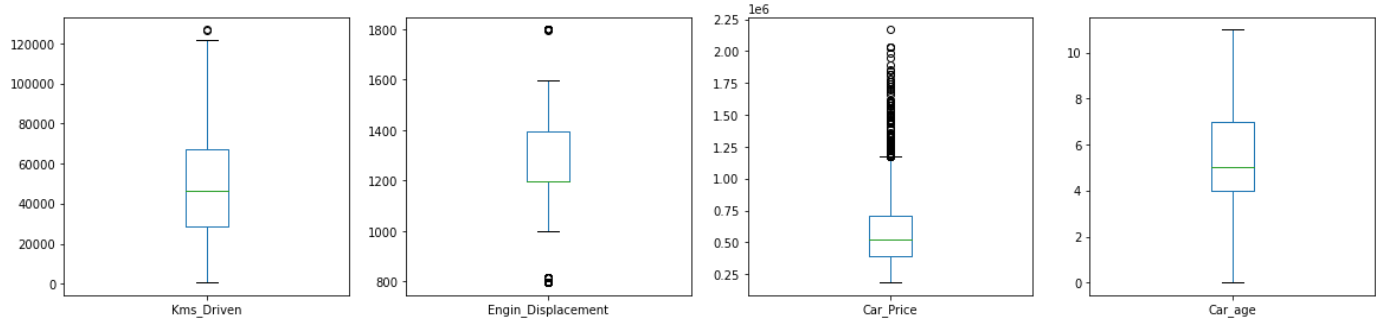
- Most of the used cars run on petrol and very few on CNG
- Most of the used cars belong to single owner
- Most of the used cars have manual transmission.
- Almost all used cars have third party insurance.
- BMW 5 series has highest price.

Boxplot

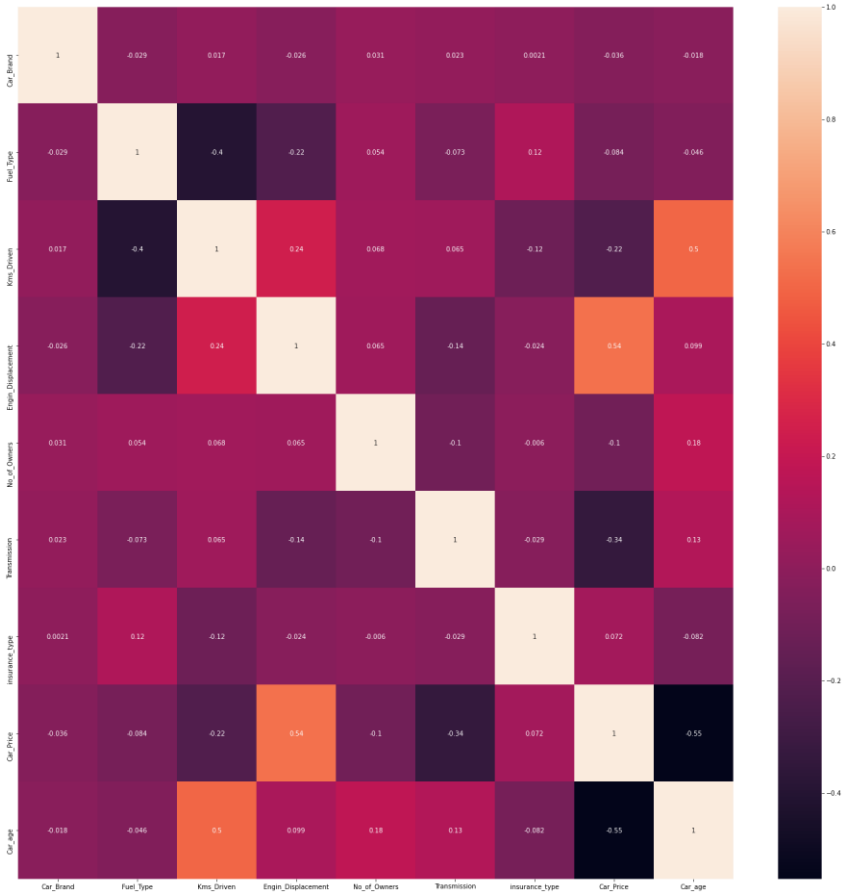


Observations:

outliers are present in Kms_Driven, Engin_Displacement, Car_age and Car_Price. however, outliers of Car_Price will not be romoved as it is our target column.



Boxplot after removing outliers.



Heat map shows no multicollinearity among attributes.

- **Run and evaluate selected models**

1.Model building:

i) RandomForestRegressor:

```
RFR=RandomForestRegressor()
RFR.fit(x_train,y_train)
pred=RFR.predict(x_test)
print('R2_score:',r2_score(y_test,pred))
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)))
```

```
R2_score: 0.9046448280273016
mean_squared_error: 0.1025831096003571
mean_absolute_error: 0.22754034217907676
root_mean_squared_error: 0.3202859809613232
```

ii) ExtraTreeRegressor:

```
ETR=ExtraTreesRegressor()
ETR.fit(x_train,y_train)
pred=ETR.predict(x_test)
print('R2_score:',r2_score(y_test,pred))
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)))
```

```
R2_score: 0.8916713587136891
mean_squared_error: 0.11653997000931565
mean_absolute_error: 0.22027435551942032
root_mean_squared_error: 0.3413795102365045
```

- Random forest regressor has given 90.4% accuracy.
- Extra tree regressor has given 89.1% accuracy.

iii) Gradient Boosting:

```
GBR=GradientBoostingRegressor()
GBR.fit(x_train,y_train)
pred=GBR.predict(x_test)
print('R2_score:',r2_score(y_test,pred))
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)))
```

```
R2_score: 0.8637395944840895
mean_squared_error: 0.14658896653481845
mean_absolute_error: 0.2769177137656511
root_mean_squared_error: 0.3828693857372491
```

iv) DecisionTreeRegressor:

```
DTR=DecisionTreeRegressor()
DTR.fit(x_train,y_train)
pred=DTR.predict(x_test)
print('R2_score:',r2_score(y_test,pred))
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)))
```

```
R2_score: 0.825253299396336
mean_squared_error: 0.18799252908336161
mean_absolute_error: 0.26682950079684614
root_mean_squared_error: 0.4335810524957953
```

- Gradient bossting has given 86.3% accuracy.
- Decision tree regressor has given 82.5% accuracy.

v) XGB Regressor:

```
XGB=XGBRegressor()
XGB.fit(x_train,y_train)
pred=XGB.predict(x_test)
print('R2_score:',r2_score(y_test,pred))
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)))
```

```
R2_score: 0.8997605237546215
mean_squared_error: 0.1078376449355699
mean_absolute_error: 0.2309362790060536
root_mean_squared_error: 0.3283864262352661 5
```

- Regressor has given 89.9% accuracy.

  - By looking into the difference of model accuracy and cross validation score I found XGB Regressor as the best model.

2. Hyper Parameter Tunning:

```
In [86]: #importing necessary libraries
         from sklearn.model_selection import GridSearchCV

In [87]: # parameter of XGBRegressor
         parameter = {'gamma':np.arange(0,0.5,0.1),
                      'n_estimators':[10,100,1000],
                      'max_depth': [2,4,6,8,10],
                      'n_jobs':[-2,-1,1,2]}

In [88]: GCV=GridSearchCV(XGBRegressor(verbosity=0),parameter,cv=5)

In [89]: GCV.fit(x_train,y_train)

Out[89]: GridSearchCV(cv=5,
                      estimator=XGBRegressor(base_score=None, booster=None,
                                             callbacks=None, colsample_bylevel=None,
                                             colsample_bynode=None,
                                             colsample_bytree=None,
                                             early_stopping_rounds=None,
                                             enable_categorical=False, eval_metric=None,
                                             gamma=None, gpu_id=None, grow_policy=None,
                                             importance_type=None,
                                             interaction_constraints=None,
                                             learning_rate=None, max_bin=None,
                                             max_cat...le_slep=None,
                                             max_depth=None, max_leaves=None,
                                             min_child_weight=None, missing=nan,
                                             monotone_constraints=None, n_estimators=100,
                                             n_jobs=None, num_parallel_tree=None,
                                             predictor=None, random_state=None,
                                             reg_alpha=None, reg_lambda=None, ...),
                      param_grid={'gamma': array([0. , 0.1, 0.2, 0.3, 0.4]),
                                  'max_depth': [2, 4, 6, 8, 10],
                                  'n_estimators': [10, 100, 1000],
                                  'n_jobs': [-2, -1, 1, 2]})

In [90]: GCV.best_params_

Out[90]: {'gamma': 0.2, 'max_depth': 8, 'n_estimators': 100, 'n_jobs': -2}
```

```
]: GCV.best_params_

]: {'gamma': 0.2, 'max_depth': 8, 'n_estimators': 100, 'n_jobs': -2}

]: Best_mod=XGBRegressor(gamma=0.2,max_depth=8,n_estimators=100,n_jobs=-2)
   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: 90.1869737579966
   mean_squared_error: 0.10556855235737327
   mean_absolute_error: 0.2364189983145356
   RMSE value: 0.32491314586728137
```

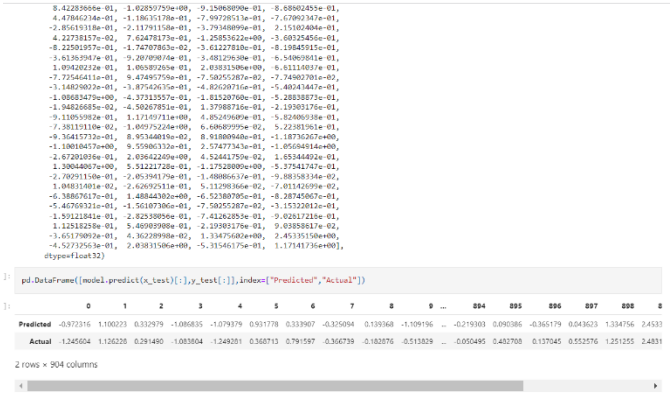
- I have choosed all parameters of XGB Regressor, after tunning the model with best parameters I have incresed my model accuracy from 89.9% to 90.18%. Also mse and rmse values has reduced which means error has reduced.

3. Saving the model and Predicting SalePrice for test data:

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

]: ['Used_cars_Price.pkl']
```

- I have saved my best model using .pkl as follows**.**
- Now loading my saved model and predicting the test values.



```
 8.42283666e-01, -1.02859759e+00, -9.15068090e-01, -8.68802455e-01,
 4.47846234e-01, -1.18635170e-01, -7.69728513e-01, -7.67902347e-01,
-2.85619318e-01, -2.11791158e-01, -3.79348099e-01,  2.15102404e-01,
 4.22738157e-02,  7.62478173e-01, -1.25853622e+00, -3.60325456e-01,
-8.22501957e-01, -1.74707863e-02, -3.61227810e-01, -8.19645815e-01,
-3.61363947e-01, -9.28709074e-01, -3.48129639e-01,  6.54063041e-01,
 1.09420232e-01,  1.06589265e-01,  2.03831508e+00, -6.61114037e-01,
-7.72546411e-01,  9.47495756e-01, -7.50255287e-02, -7.74902701e-02,
-1.14029022e-01, -1.87542635e-01, -4.82620716e-01, -5.40343447e-01,
-1.08683479e+00, -4.37313557e-01, -1.81520760e-01, -5.28838873e-01,
-1.94826685e-02, -4.50267851e-01,  1.37988716e-01, -2.19303176e-01,
-9.11059382e-01,  1.17169771e+00,  4.85249497e-01, -5.82406338e-01,
-7.38119110e-02,  1.06975224e+00,  6.68689995e-02,  5.22381961e-01,
-9.36415732e-01,  8.95344019e-02,  8.91800940e-01, -1.18736267e+00,
-1.10018457e+00,  9.55906332e-01,  2.57477343e-01, -1.05694914e+00,
-2.67201036e-01,  2.03642240e+00,  4.52441759e-02,  1.65344402e-01,
 1.30044067e+00,  5.51221728e-01, -1.17528009e+00, -5.37541747e-01,
-2.70291150e-01, -2.05394179e-01, -1.48086637e-01, -9.88358334e-02,
 1.04831401e-01, -2.62692511e-01,  5.11298366e-02, -7.01142099e-02,
-6.38867617e-01,  1.48844302e+00, -6.52380705e-01, -8.28745067e-01,
-5.46769321e-01, -1.56107306e-01, -7.50255287e-02, -3.15522012e-01,
-1.59121841e-01,  2.02538056e-01, -7.41262855e-01, -9.02617218e-01,
 1.12518234e-01,  5.46903908e-01, -2.10985175e-01,  9.03858817e-02,
-3.65179092e-01,  4.36228998e-02,  1.33475602e+00,  2.45135150e+00,
-4.52732563e-01,  2.03831506e+00, -5.31546175e-01,  1.17141736e+00]],
dtype=float32)
```

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

	0	1	2	3	4	5	6	7	8	9	...	894	895	896	897	898	8
Predicted	-0.972316	1.100223	0.332979	-1.086835	-1.079379	0.931778	0.333907	-0.325094	0.139368	-1.109196	...	-0.219303	0.090386	-0.365179	0.043623	1.334756	2.4533
Actual	-1.245604	1.126228	0.291490	-1.083804	-1.249201	0.368713	0.791597	-0.366739	-0.182876	-0.513829	...	-0.050495	0.402708	0.137045	0.552576	1.251255	2.4831

2 rows × 904 columns

- Plotting Actual vs Predicted values.

Predicting Used cars Price for test dataset:

- I have predicted the SalePrice to save model. I have also saved my predictions for further analysis.

- **Interpretation of the Results:**

- The dataset was scrapped from cardekho website.
- The dataset has 10 features with 3012 samples.
- The dataset has no null values.
- plotting features helped us to get better insight on the data. I found both numerical columns and categorical columns in the dataset so I have chosen dist plot, and bar plot to see the relation between target and features.
- There huge number of outliers are present in the data so we have chosen proper methods to deal with the outliers. If we ignore outliers, 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 XGB Regressor as the best model with 90.18% r2_score. Also, I have improved the accuracy of the best model by running hyper parameter tunning.

4.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 analyse 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.

- **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 drawback is scrapping the data as it is fluctuating process.
- Followed by a greater number of outliers and skewness these two will reduce our model accuracy.
- Also, we have tried best to deal with outliers, skewness and null values. So it looks quite good that we have achieved a accuracy of 90.18% even after dealing all these drawbacks.
- Also, this study will not cover all Regression algorithms instead, it is focused on the chosen algorithm, starting from the basic ensembling techniques to the advanced ones.

5.REFERENCE

www.cardekho.com