Used Car Price prediction

A Step-by-Step guide to Build a Machine Learning Model to predict Car Prices

Submitted By:

Razni Nazeem

08.08.2022

FlipROBO Technologies

ACKNOWLEDGEMENT

I would like to express my special gratitude to the “Flip Robo” team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzing skills. Also, I want to express my huge gratitude to Ms. Khushboo Garg (SME FlipRobo), who has helped me overcome difficulties within this project and others.

I would also like to thank various websites like stackoverflow, Kaggle, medium and towardsDatascience for helping me resolve any issues I face during my project.

A huge thanks to my academic team “Data trained” who has helped me grow from a non-Coder to what I am Now. Lastly, I would like to extend my Heartfelt thanks to my Husband and kids because without their support this project would not have been successful. And thank you to many other persons who have helped me directly or indirectly to complete the project.

INTRODUCTION

Business Problem Framing:

In this Article, I will be guiding you to the step-by-step procedure in building a Machine Learning model in Python using popular machine learning libraries NumPy, Pandas & scikit-learn to predict used Car prices in India.

As per Times of India, Small-town India is cashing in on its luxury cars taking advantage of the soaring demand and a resultant rise in prices of pre-used luxe vehicles. In the first six months of Calendar 2022, Tier 2 luxury car listings were up 45% compared to 40% for Tier 1 listings on the OLX Platform. This project was scraped on 24th July,2022 from Cars24.com website for Used cars. I have taken 11 major cities in India including New Delhi, Chennai, Bangalore and the dataset includes 21 explanatory features and 9176 entries of Used cars. The aim is to predict the used car Prices for our client with respect to their budgets and priorities. By analyzing previous market trends and price ranges, and upcoming developments future prices will be predicted.

Conceptual Background of the Domain Problem:

To help machines understand like humans do and to strengthen AI, machine Learning is required. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, improving their marketing strategies and focusing on changing trends in used car sales and purchases.

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.

Why is used Car price prediction important?

Used Car Price prediction is important to drive Economic efficiency. As earlier, Used cars were bought and sold mostly through a small trader or family in the Locality. But as Internet risen substantially, Websites and Applications for used Cars have grown tremendously. It is very Important to build a model for used Car price trend so that small traders can get a hold of the current market situations. Traders can investigate the market for which Brands and Models the price goes up and for which its coming down. In that way, Small traders can grow their Business and hence the Country also grows. Therefore, the used Car Price prediction model is very essential in filling the information gap and improving Economic efficiency.

Review of Literature

In the Literature, few researchers applied various machine Learning techniques to predict the car cost as per the given requirements. It is inferred that establishing a simple linear mathematical relationship for these time-series data is found not viable for forecasting. Hence it became imperative to establish a non-linear model which can well fit the data characteristic to analyze and forecast future trends. This paper aims to build a model to predict used cars' reasonable prices based on multiple aspects, including vehicle mileage, year of manufacturing, fuel consumption, transmission, fuel type, and Models.

Therefore, in this project report, we present various important features to use while predicting used car prices with good accuracy. We can use regression models, using various features to have lower Residual Sum of Squares error. While using features in a regression model some feature engineering is required for better prediction.

Before the actual start of model-building, this project visualized the data to understand the dataset better. To evaluate the performance of each regression, R-square was calculated. Both the seller and the buyer should have a fair deal. This paper presents a system that has been implemented to predict a fair price for any pre-owned car.

Motivation for the Problem Undertaken:

I must model the price of Used Cars with the scraped independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on those Cars that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market. 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. There is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features.

LIBRARIES IMPORTED

There are 3 sets of libraries used.

- Basic Libraries for Data Analysis and Visualization
- Libraries for Data Cleaning and Feature Engineering (Data preprocessing)
- Libraries for Building the ML Models

Basic Libraries used are:

```
#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')
```

Libraries for Data Cleaning and Feature Engineering (Data preprocessing):

```
#for Outliers removal, z-score is used
from scipy.stats import zscore

#for Skewness removal, Poer transformer is used
from sklearn.preprocessing import PowerTransformer

#for rncoding Ordinal encoder is used
from sklearn.preprocessing import OrdinalEncoder

#for normalizing, standard scaler is used
from sklearn.preprocessing import StandardScaler

#for checking multicollinearity, VIF Factor is used
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

Libraries for Building the ML Models:

```
#Importing regression libraries

from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.linear_model import ElasticNet
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
from xgboost import XGBRegressor

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

These are the libraries used in my Jupyter notebook for Prediction.

Analytical Problem Framing

1. Mathematical/Analytical Modeling of the Problem

After Importing basic libraries, I have loaded my dataset and opened it.

	City	Car Full Name	Car Name	Brand	Model	Variant	Year	Transmission	Max power BHP	Max power RPM	Max tourque NM	Max tourque RPM	Fuel Type	Fuel tank capacity	Cylinders	Seating capacity
0	Gurgaon	2014 MARUTI WAGON R 1.0 VXI MANUAL	Maruti Wagon R 1.0	MARUTI	WAGON R 1.0	VXI	2014	MANUAL	NaN	NaN	NaN	NaN	Petrol	NaN	NaN	NaN
1	Gurgaon	2010 HONDA CITY S MT PETROL MANUAL	Honda City	HONDA	CITY	S MT PETROL	2010	MANUAL	118	NaN	146	NaN	Petrol	42	4 , Inline	5
2	Gurgaon	2019 MG HECTOR SHARP DCT PETROL AUTOMATIC	MG HECTOR	MG	HECTOR	SHARP DCT PETROL	2019	AUTOMATIC	141	NaN	250	NaN	Petrol	60	4 , Inline	5
3	Gurgaon	2019 MG HECTOR SHARP 2.0 DIESEL MANUAL	MG HECTOR	MG	HECTOR	SHARP 2.0 DIESEL	2019	MANUAL	169	NaN	350	NaN	Diesel	60	4 , Inline	5
4	Gurgaon	2013 HYUNDAI I20 ERA 1.4 CRDI MANUAL	Hyundai i20	HYUNDAI	I20	ERA 1.4 CRDI	2013	MANUAL	90	NaN	220	NaN	Diesel	45	4 , Inline	5

We have 9176 rows and 21 columns including our Target “Price” for the dataset.

By looking into the Target column “Price” which is continuous, I came to know that it is a Regression problem. I analyzed each column’s information and unique values. I saw a huge number of missing values in certain columns and more than 60% of the data in certain columns as 0. So, I dropped those columns to avoid high bias and variance.

I have Included complete life cycle of data science.

1. Data Cleaning
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Building
5. Model Evaluation
6. Selecting the best model

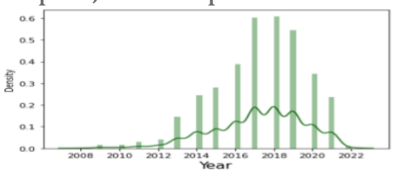
Data Cleaning

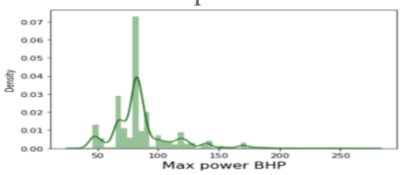
I used an Imputation technique to replace NaN values in the train and test dataset. Further I

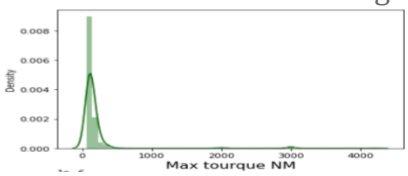
```
df['Max power BHP'] = df['Max power BHP'].str.replace('73 bhp @ 6000 rpm','73')
df['Max power BHP']= df['Max power BHP'].astype('float')
```

Exploratory Data Analysis

Afterwards, I have analyzed the dataset using plots and other visualization techniques like barplot, distplot etc. I also plotted features with the Target.







Data Pre-Processing

Then I have removed Outliers, skewness and found correlation between variables and removed multicollinearity, and normalized the dataset using Standard scaler.

```
from sklearn.preprocessing import StandardScaler

scaler=StandardScaler()
x = pd.DataFrame(scaler.fit_transform(x), columns=x.columns)
```

Model Building

I have created 7 models and used hyperparameter tuning to find the best score. Regplot has been plotted for each model for their actual and predicted values.

```
pred_knn = knn.predict(x_test)

print('The r2 score is:', r2_score(y_test, pred_knn))
print('The mean absolute error', mean_absolute_error(y_test, pred_knn))
print('The mean squared error', mean_squared_error(y_test, pred_knn))
print('root_mean_squared_error:',np.sqrt(mean_squared_error(y_test,pred_knn)))
```

```
The r2 score is: 0.9109521824868332
The mean absolute error 54550.53504074506
The mean squared error 8448656758.045605
```

Model Evaluation

Cross Validation score has also been computed and then found the best model and saved for future predictions.

```
cv = cross_val_score(XGB_H, x,y,cv=5)
print('The cross validation score', cv.mean())
```

```
The cross validation score 0.9752155134770613
```

Selecting the best model

Using the saved model, I have predicted car prices of the dataset and stored it in a csv file. Using my model, anyone can find the Used car prices provided the features are similar as in my model.

```
# Saving the model using .pkl

import joblib
joblib.dump(Final_model,"Used_Car_Price.pkl")
```

```
['Used_Car_Price.pkl']
```

2. Data Sources and their formats:

Data was collected by Web Scraping the Website - Cars24.com using Selenium. Cars24.com is a next generation ecommerce platform for pre-Owned Cars. They have used Cars in all cities In India. Moreover, cars24 is a Worldwide website which is operating in almost all countries. The website provides Inspection checklist and Finance option for those who opt for loan payment. I have scraped 21 variables and 9176 rows which contains all the ads posted in 11 Major cities in India.

In My dataset, I have object, float, and integer types of data.

3. Data Processing done

* Imported necessary Libraries and loaded the dataset.

* Statistical Analysis done like Shape, info, nunique, value_counts.

* Dropped columns with more than 60% of NaN values (Max power RPM, Max tourque RPM).

* Imputation Technique to replace other NaN values.

* Dropped column- History which had 1 value.

Checked Correlation between Variables and Feature

* Outliers removed, Encoded categorical columns.

* Skewness removed, removed columns with Multicollinearity issue.

* Using Standard Scaler, standardized the data.

4. Data Inputs- Logic- Output Relationships

* The relation between Categorical columns with Target has been found using Boxenplot.

* The relation between Numerical columns with Target has been found by using Scatterplot.

* Various columns have a linear relationship with the Target. While Some columns do not have any specific Pattern.

5. Hardware and Software Requirements and Tools Used

* Hardware: **Processor**- Core i5 and above, **RAM**- 8GB or above, **SSD**- 250 or above

* Software: Anaconda

* Libraries Used mentioned before.

MODEL DEVELOPMENT AND EVALUATION

1. Identification of possible problem-solving approaches

I have checked the information of the dataset regarding null values and datatypes.
I have 1 float type, 4 INT type and 16 Object type data. I then checked the missing data.
There is a lot of missing data.

I have dropped those columns with more than 60% missing values.

```
#Droping unnecessary columns which have more null values
df = df.drop(["Max power RPM"],axis=1)
df = df.drop(["Max tourque RPM"],axis=1)
```

I used an imputation technique to replace NaN values.

```
for col in ["Max power BHP"]:
    df[col] = df[col].fillna(df[col].median())
```

I used the Z-score method to remove Outliers.

```
from scipy.stats import zscore

z=np.abs(zscore(out_cols))
df_new=df[(z<3).all(axis=1)]
df_new
```

I used a Power transformer to remove Skewness.

```
from sklearn.preprocessing import PowerTransformer
scaler = PowerTransformer(method='yeo-johnson')
x[skew_cols] = scaler.fit_transform(x[skew_cols].values)
```

I used Ordinal Encoder to convert categorical columns to numerical.

```
from sklearn.preprocessing import OrdinalEncoder
enc=OrdinalEncoder()

for i in cat_cols:
    df_new[i]=enc.fit_transform(df_new[i].values.reshape(-1,1))
```

I used Pearson Correlation to find the correlation between variables.

```
#Plotting the Heatmap of Correlation

corr = df_new.corr()
plt.figure(figsize=(20,10), facecolor='white')
sns.heatmap(corr, annot=True,cmap='rocket',fmt='.2f')
plt.show()
```

I used Standard Scaler to scale and normalize the data.

```
scaler=StandardScaler()
x = pd.DataFrame(scaler.fit_transform(x), columns=x.columns)
```

I used various Machine Learning algorithms to create models to predict Car Price.

```
The r2 score is: 0.954545964799164
The mean absolute error 40378.30493983436
The mean squared error 4312576685.253435
```

2. Testing of Identified Approaches (Algorithms)

Since our Target is Price, which is continuous, I have a Regression Problem. I have used 7 different algorithms to build the models and found the R2 score and CV Score of each one of them. I have finally decided to select the model which has the highest r2 and CV Score and least MSE, RMSE and MAE and that model is Random Forest Regressor model.

I have Used: Linear Regression, KNN Regressor, Random Forest Regressor, XGB Regressor and Gradient Boosting Regressor.

3. Run and evaluate selected models

A. Linear Regression:

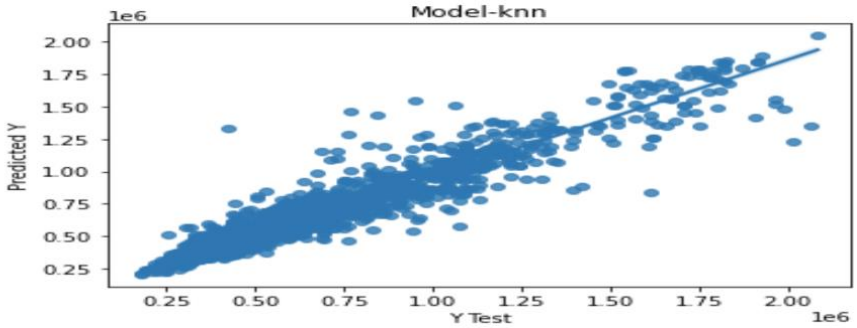
At first, I found the best Random state for which I got the best score and performed a train-test-split to fit the model. My score for Linear regression model is 73.33% and CV Score of 69.68%. I have tuned with the best parameters, but score remained the same.

```
pred = Rd.predict(x_test)
print('The r2 score is:', r2_score(y_test, pred))
print('The mean absolute error', mean_absolute_error(y_test, pred))
print('The mean squared error', mean_squared_error(y_test, pred))
cv = cross_val_score(Rd, x,y,cv=5)
print('The cross validation score', cv.mean())
```

```
The r2 score is: 0.7333770327426984
The mean absolute error 107775.91304121495
The mean squared error 24377502674.3793
```

B. KNN Regressor:

I found the best random state which yields the best score and then fit the model. R2 score for KNN Regressor was 91.09%. I have tuned with different parameters and the score has improved to 95.45% and CV Score of 94.64%. The regplot of actual and predicted values using KNN is:



C. Ridge Regression:

The R2 score of Ridge regressor was 73.37% and CV Score of 69.68%. Same as Linear regression.

D. Random Forest Regressor (Best Model):

I first found the best random state and fit the model. I got a score of 98.94% and a CV Score of 97.04%. Hence, I selected the random forest as the best model and saved the model. While performing the fitting of the final model, my score was improved, and it became **98.98%**. Hyper parameter tuning of the model did not increase my score.

```
pred_rdf = rdf.predict(x_test)

print('The r2 score is:', r2_score(y_test, pred_rdf))
print('The mean absolute error', mean_absolute_error(y_test, pred_rdf))
print('The mean squared error', mean_squared_error(y_test, pred_rdf))
print('root_mean_squared_error:',np.sqrt(mean_squared_error(y_test,pred_rdf)))
```

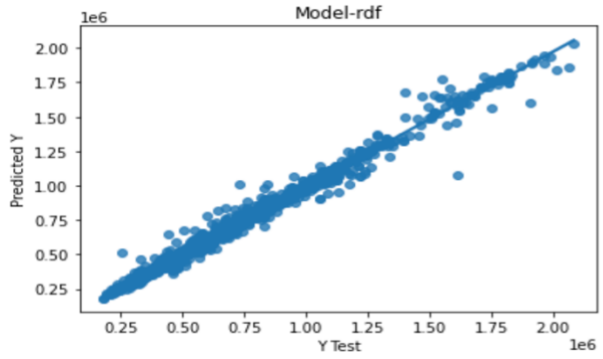
```
The r2 score is: 0.9894158276247104
The mean absolute error 15711.477481567714
The mean squared error 1004202483.1612341
```

The CV Score:

```
cv = cross_val_score(rdf, x,y,cv=5)
print('The cross validation score', cv.mean())
```

```
The cross validation score 0.9704991901546685
```

The regplot of actual VS Predicted for Random Forest model shows it's a good model.



I saved the Final Model using joblib:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state=max_RS)
pred_rdf = Final_model.predict(x_test)
print('The r2 score is:', r2_score(y_test, pred_rdf))
print('The mean absolute error', mean_absolute_error(y_test, pred_rdf))
print('The mean squared error', mean_squared_error(y_test, pred_rdf))
print('root_mean_squared_error:',np.sqrt(mean_squared_error(y_test,pred_rdf)))
```

```
The r2 score is: 0.9898587881955153
```

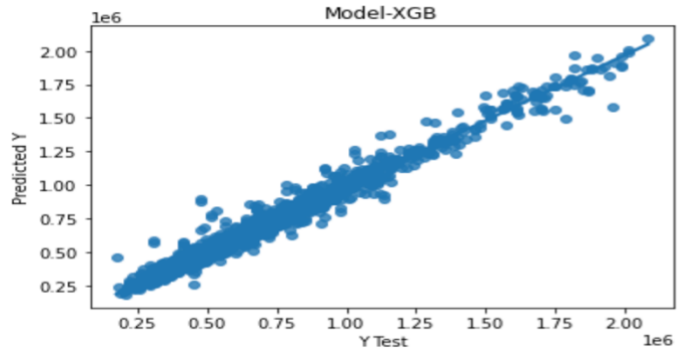
While saving the Final model I got accuracy of 98.98%. Hence, my model is saved for further predictions.

E. XGB Regressor:

For XGB Model, the r2 score was initially 97.76% NS after Hyper parameter tuning score Improved to 98.04%. The CV Score for the XGB hyper parameter tuned model was 97.52%. The regplot of XGB model is also very good and linear.

```
XGB_H=XGBRegressor(learning_rate=0.1,max_depth=7,n_estimators=300,
                   reg_alpha=0.5,reg_lambda=1,gamma=0.05)
XGB_H.fit(x_train,y_train)
xgbpred=XGB_H.predict(x_test)
print('The r2 score is:', r2_score(y_test,xgbpred))
print('The mean squared error', mean_squared_error(y_test,xgbpred))
print('The mean absolute error', mean_absolute_error(y_test,xgbpred))
print('root_mean_squared_error:',np.sqrt(mean_squared_error(y_test,xgbpred)))
```

```
The r2 score is: 0.9804031840462478
The mean squared error 1938634330.2744417
The mean absolute error 26775.886623253784
```

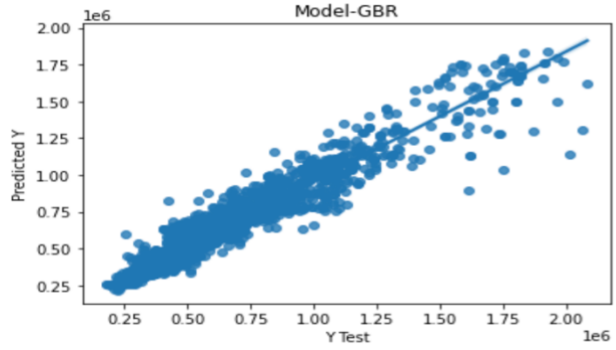


F. SGD Regressor:

The R2 score for SGD Model is 73.10% and CV Score of 69.61% similar to Linear regression and Ridge.

G. Gradient Boosting Regressor:

The R2 score of Gradient Boosting regressor model was 91.82%. But after Hyper parameter tuning, R2 score improved to 95.32% and CV Score of 93.53% Regplot of this model is also linear:



4. Key Metrics for success in solving problem under consideration

* I have used the r2 score as the accuracy score of the model.

* I have used mean squared error, mean absolute error to find the error rate in the model.

* I have used root mean squared error and took the least amount as the best fit model.

* I also used Cross Validation Score to cross verify with r2 score and find the best model which has the least difference between r2 score and cv Score mean.

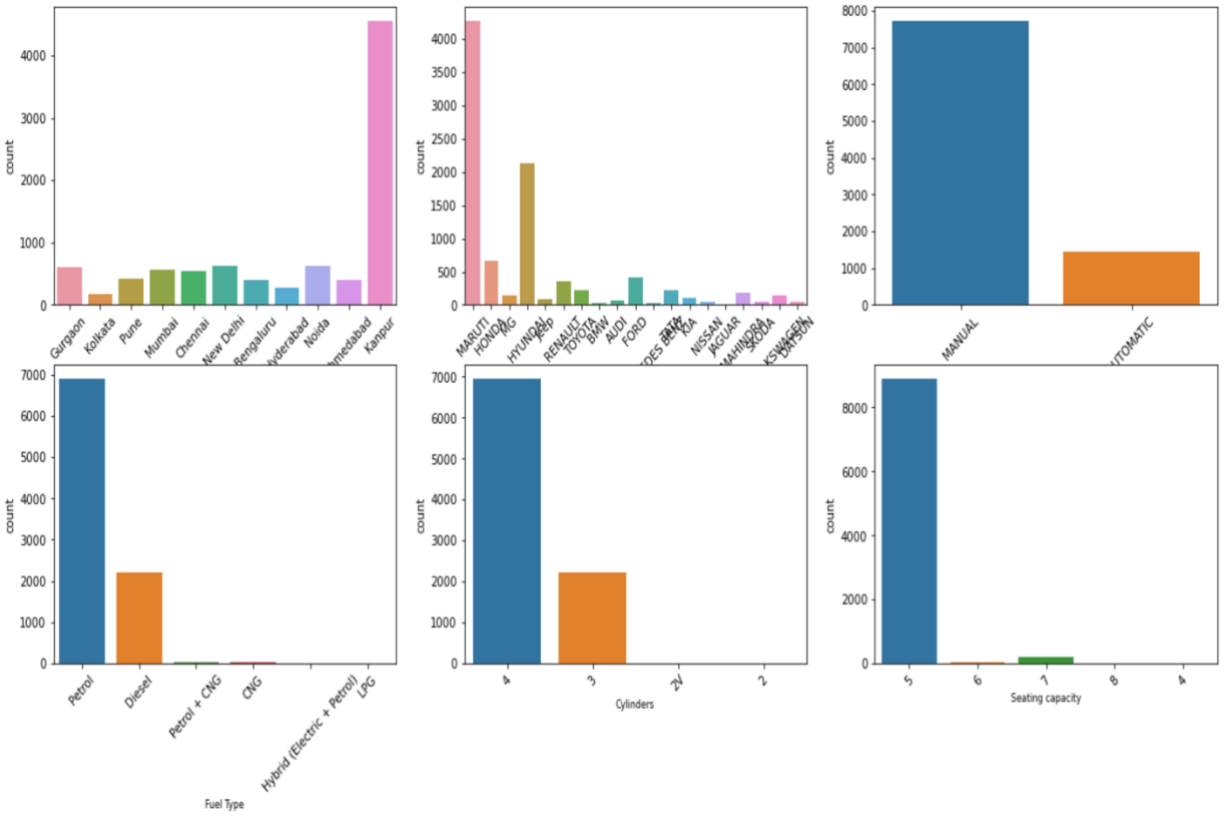
I had Excellent R2 score and Cross-Val score for 4 of the models.

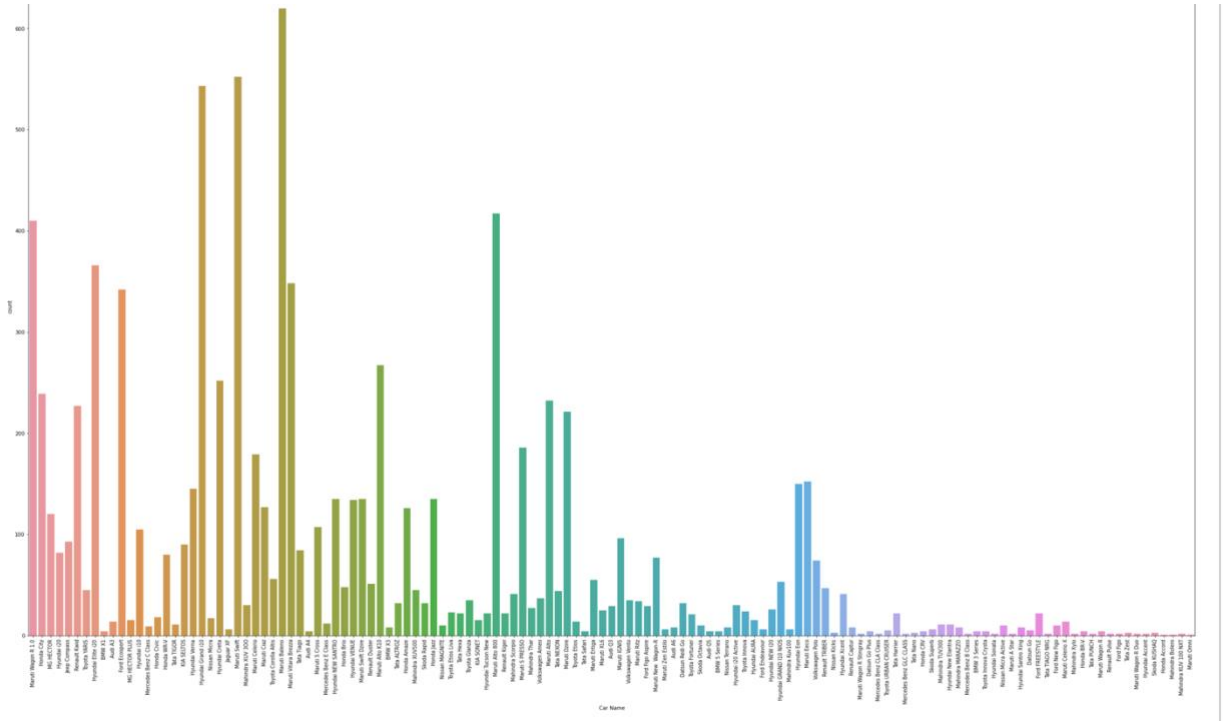
- knn_h
  - The r2 score is: 0.954545964799164
  - The mean absolute error 40378.30493983436
  - The mean squared error 4312576685.253435
  - The cross validation score 0.9464154083476279
- Rdf
  - The r2 score is: 0.9894158276247104
  - The mean absolute error 15711.477481567714
  - The mean squared error 1004202483.1612341
  - The cross validation score 0.9704991901546685
- xgb_h
  - The r2 score is: 0.9804031840462478
  - The mean squared error 1938634330.2744417
  - The mean absolute error 26775.886623253784
  - root_mean_squared_error: 44029.925394831655
  - The cross validation score 0.9752155134770613
- Gbr_h
  - R2_score: 0.9532958060961026
  - mean_squared_error: 4431188932.810062
  - mean_absolute_error: 44146.59763280022
  - root_mean_squared_error: 66567.17609159985
  - The cross-validation score 0.9353157365385949

As we can see all the models have excellent R2 score and CV Score. But the MSE, RMSE and MAE are very high. From above models, I am taking Random Forest model as it is slightly better in terms of Errors.

5. Visualizations

- I Have used barplots to visualize the count of Categorical data.

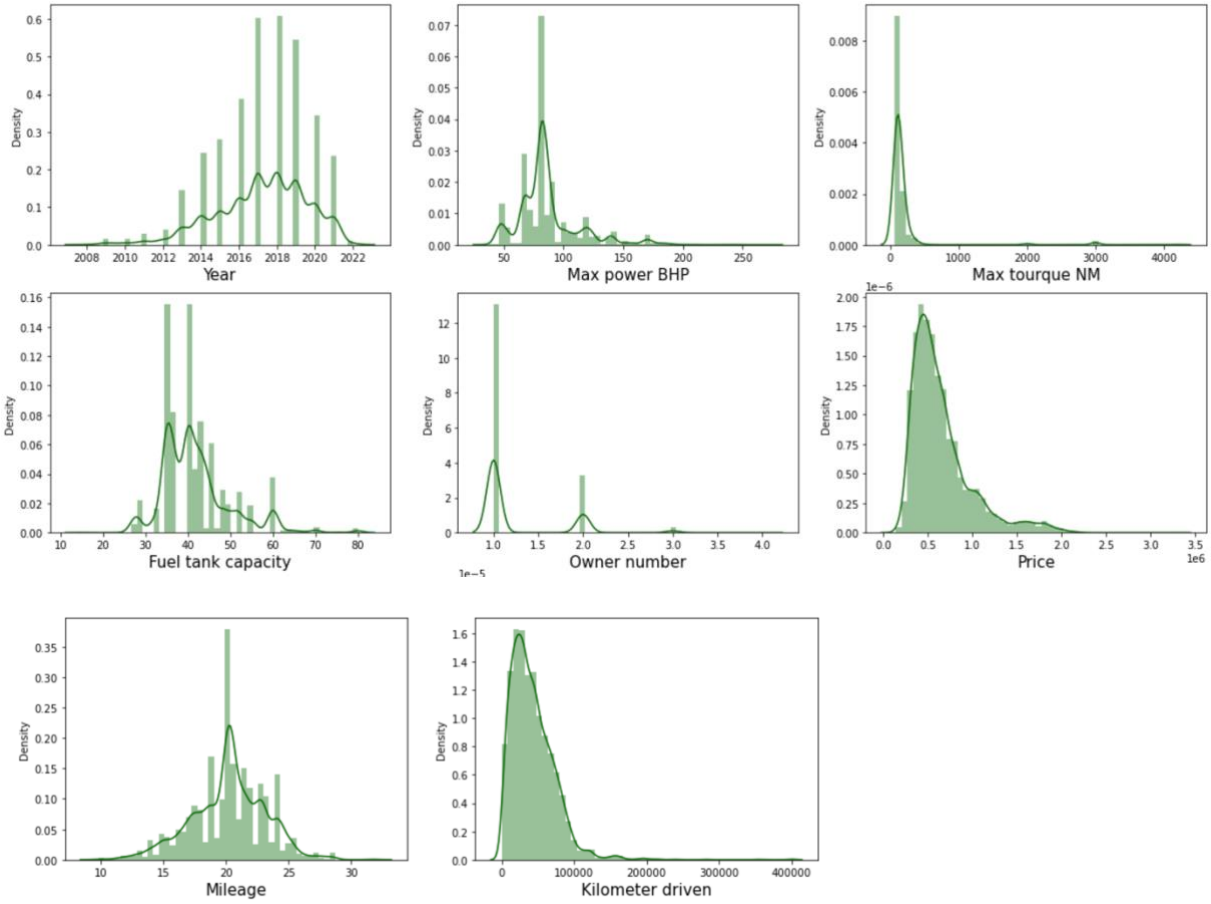




Observations of Count plot of Categorical Data:

- Most of the data of used cars are from City Kanpur and least from Kolkata.
- Maruti is the most popular Brand in the dataset followed by hyundai. Least from Jaguar.
- Most of the cars are Manual Transmission.
- Most of the cars are Petrol as fuel type and i also have a very few number of Hybrid cars as well.
- Most of the cars are 4 cylinders and a very few 2 cylinders inline and V shape.
- Most cars have seating Capacity of 5 Seater.
- Maruti baleno is the most popular in my dataset followed by Maruti Swift and Hyundai grand i10.
- I have Maruti Alto800 and Maruti wagonR1.0 also on the top 5 list.
- We have only 1 data of Honda Accord, Maruti Omni and Mahindra Bolero.
- As mentioned earlier, Maruti is the top Brand followed by Hyundai in my dataset.

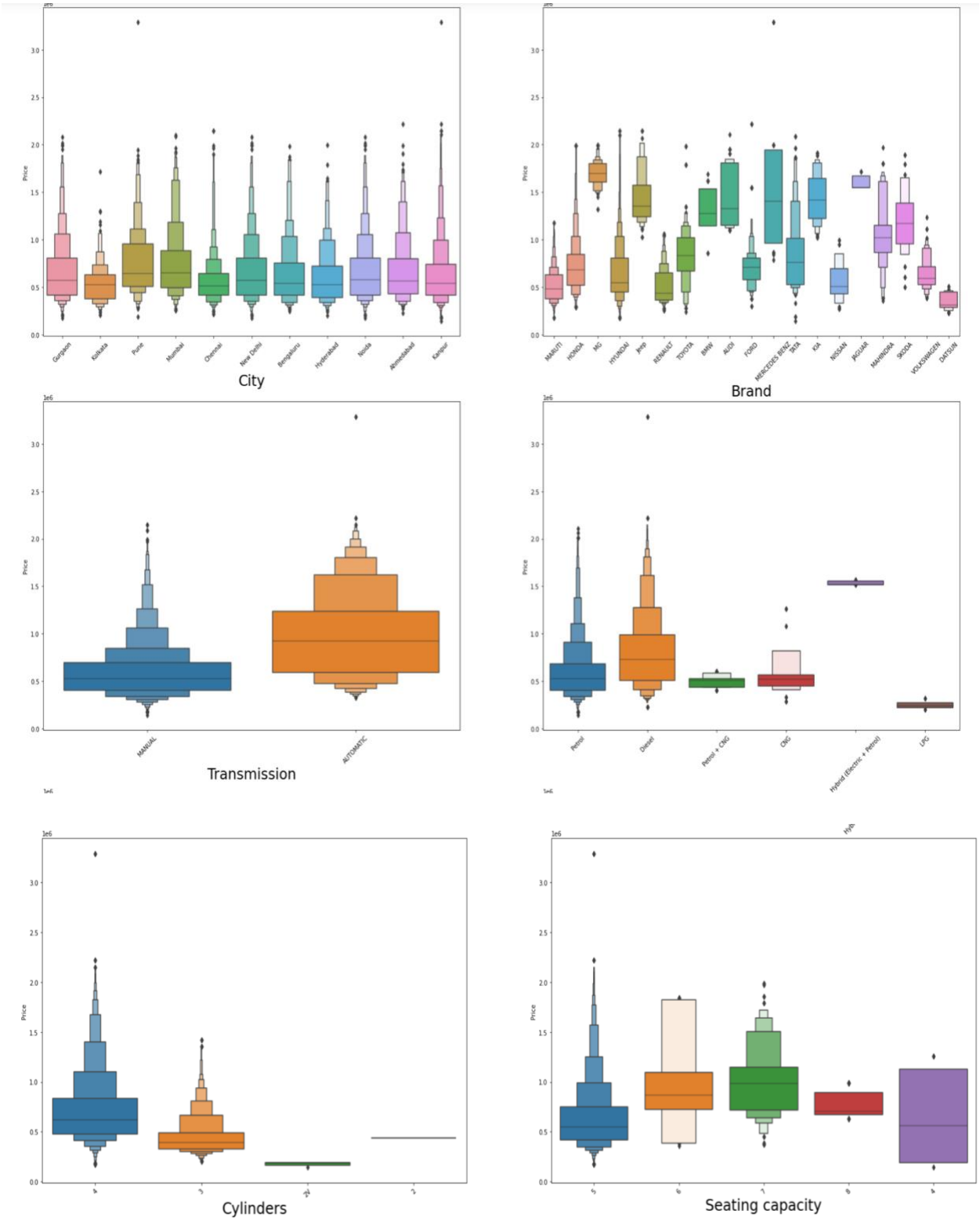
- **I have used distplot to analyze distribution of numerical data.**



Observations:

- Max power BHP, Max torque NM, Price, Kilometer driven, Owner numbers are right skewed.
- Year is left skewed.
- Fuel tank capacity and Mileage are somewhat normally distributed

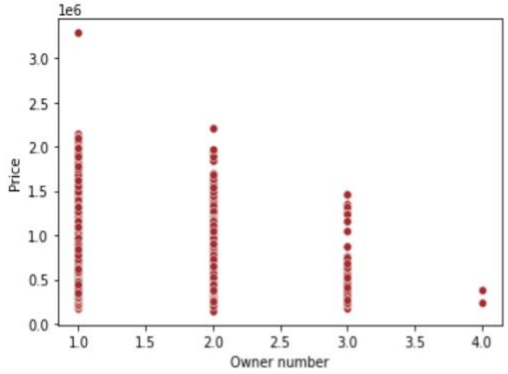
- **I have used a boxen plot to find the relation between categorical and target.**

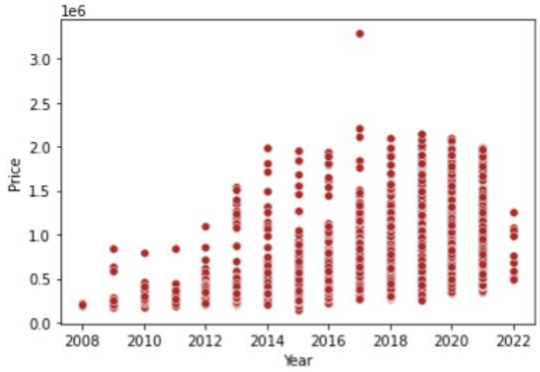


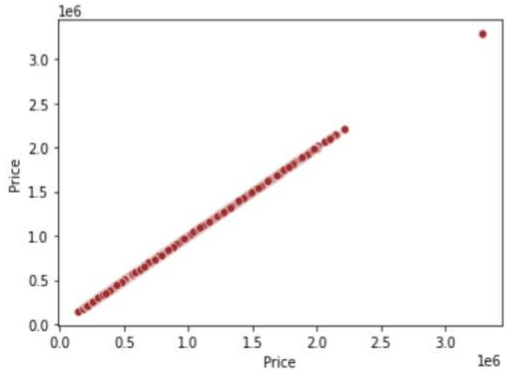
Observations:

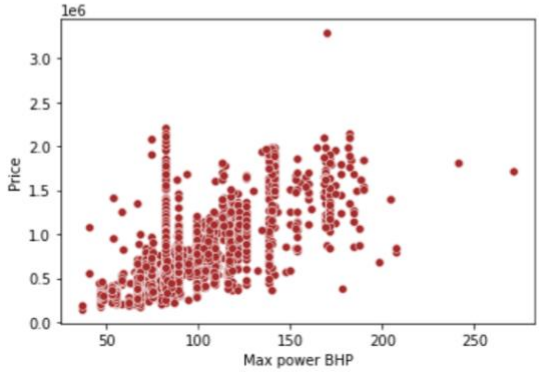
- City - City do not have specific relation with Price.
- Brand - Mercedes is the most expensive Car in my dataset and Datsun Brand cars are mostly cheap.
- There are many outliers in the Brands column as the prices are sometimes very high for few of the models.
- Transmission - As we can see that Automatic cars are very expensive than Manual ones.
- Fuel type - Hybrid and Diesel cars are expensive than others.
- Cylinders - 4 cylinder cars are very expensive and 2 and 2V cylinders are cheap.
- Seating capacity - 6 and 7 seaters are most expensive than 5 and 4 seater.

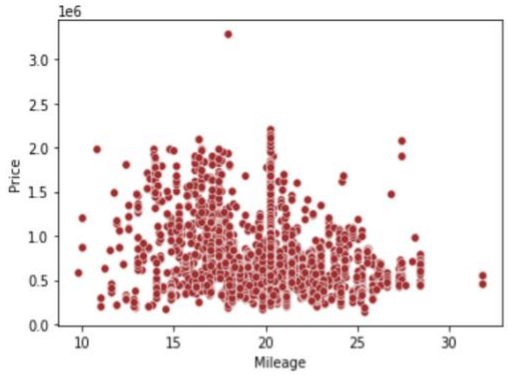
- **I have used scatter plot for visualization of numerical columns with target.**

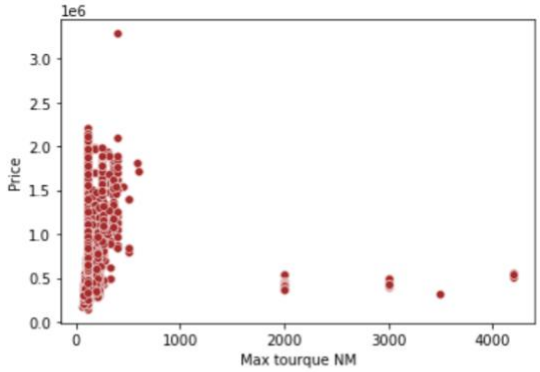


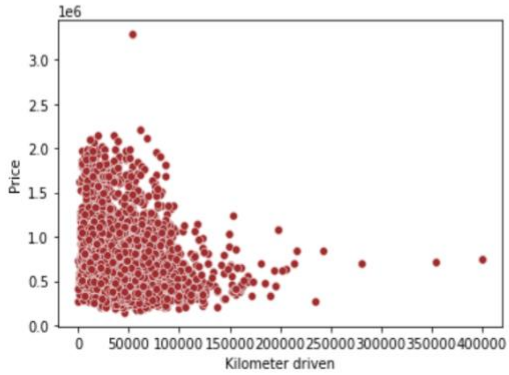


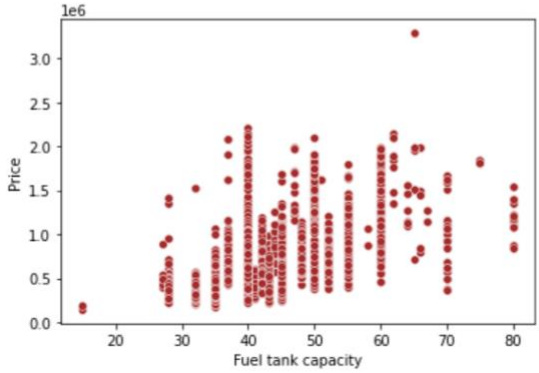












Observations:

- We can see some of the columns have direct relations and some do not have any relation with Target Price.
- Year - As year increases, Price also increases.
- Max power BHP - As max power BHP increases, Price also increases.
- Max torque NM - It shows least the torque, most of the price lies in there. no specific relation.
- Fuel Tank capacity - Most of the Price lies in between 35 to 60 litres capacity.
- Owner Numbwer - Price decreases as owner number increases.
- Mileage - Mileage and Price do not have specific relationship.
- Kilometers driven - Price is more when kilometers driven is less.

CONCLUSION

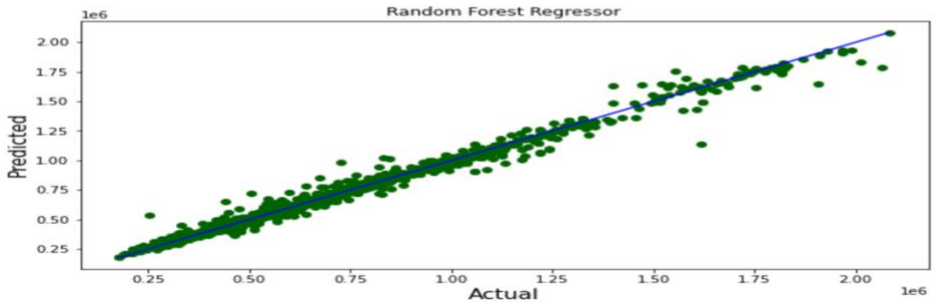
1. Key Findings and Conclusions of the Study

In this project report, I have used machine learning algorithms to predict the Used car prices. I have mentioned the step-by-step procedure to analyze the dataset and find the correlation between the features. Hence, we calculated the performance of each model using different performance metrics and compared them based on these metrics. Then we have also saved the data frame of predicted prices and Actual Price.

I have observed that certain features like Max power BHP, Year of the Vehicle, etc. contribute the most to the Price of the Car. Also, conditions like Kilometers driven negatively affect the price. As years passed the value decreased.

Actual Vs Predicted Plot ¶

```
plt.figure(figsize=(10,5))
plt.scatter(y_test, prediction, c='darkgreen')
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("Random Forest Regressor")
plt.show()
```



2. Learning Outcomes of the Study in respect of Data Science

Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed, and analyzed. The power of visualization has helped us in understanding the data by graphical representation. Data cleaning is one of the most important steps to remove missing values and to replace them with respective mean, median or mode. This study is an exploratory attempt to use seven machine learning algorithms in estimating Car prices, and then compare their results.

I hope this study has moved a small step ahead in providing some methodological and empirical contributions to Car Price appraisal and presenting an alternative approach to the valuation of Used Cars. Future direction of research may consider incorporating additional data from a larger geographical location with more features or analyzing other car types beyond development.

3. Limitations of this work and Scope for Future Work

- ➢ The data was collected from a website, and I found that many of the ads are posted in different cities. This causes duplication of data.
- ➢ There were so many outliers present in the dataset.
- ➢ There was a lot of skewness present in the dataset which will again affect the model as we must transform it.
- ➢ This study did not use all advanced algorithms but only a few simple regression algorithms to a few advanced ones.
- ➢ There were a few columns with more missing values. I must remove those columns.
- ➢ There was multicollinearity and the column had to be removed to prevent multicollinearity.
- ➢ The MSE, MAE and RMSE is very high for the dataset.

Even after all these Limitations and drawbacks, my model tends to perform well with an accuracy of 98.98% with Random Forest model and a CV Score of 97.05%.

REFERENCES

1. https://timesofindia.indiatimes.com/city/chennai/small-town-india-cashing-in-as-used-luxury-car-prices-zoom/articleshow/93199391.cms
2. https://www.analyticsvidhya.com/blog/2021/05/feature-engineering-how-to-detect-and-remove-outliers-with-python-code/
3. https://auto.hindustantimes.com/auto/cars/once-more-expensive-than-new-models-prices-of-used-cars-in-uk-cool-off-41658468481480.html
4. https://www.autoblog.com/2021/12/22/used-car-price-drop-coming-2022/#:~:text=A%20new%20report%20from%20Automotive,relationship%20to%20new%20car%20prices.

THANK YOU