FLIP ROBO

A Project Report on Car Price Prediction Model

Submitted by:

Satu Vinayak Pole

ACKNOWLEDGMENT

I would like to express my special gratitude to my SME Mr. Shwetank Mishra as well as "Flip Robo" team for letting me work for a project named "Car Price Prediction" . Also thanks to my institute 'Data Trained' .

Also I would like to thank websites such as StackOverflow, geeksforgeeks and Youtube who has helped me in solving issues and errors.

[1] INTRODUCTION

{1.1} **Business Problem Framing.**

Predicting the price of used cars is an important and interesting problem. Also it looks easy to say but it's a somewhat difficult task to do as there many factors that play role in deciding price. The most important ones are type of car, age of car, past owners associated with car, build quality of the car, interior and exterior damages on the car, transmission and fule type of car, odometer reading which shows distance travelled by car and main is what brand and model is of that car. In addition to these there are minor factors such as audio system in car, tyre wear, upholstery in car, engine type and power also matters in deciding value of the cars. These mentioned factors does change with person to person ; as for example a normal working person who has to go through traffic daily will opt for smaller car which provides him with good fuel efficiency and also with cheaper fuel options such as CNG/LPG, in other case the person who is a highway cruiser or nature enthusiast will most likely to opt for SUV cars as they have the better capability to provide smooth rides on highways as well as bumpy roads. The service network provided by various car brands is very vital in deciding car price.

Unfortunately, information about all these factors is not always available and the buyer must make the decision to purchase at a certain price based on few factors only. In this work, I have considered only a small subset of the factors which are more important.

{1.2} **Conceptual Background of the Domain Problem**

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. Hence we need to provide new machine learning models which are based on revised attributes and with respect to new prices as are in market. This model will provide approximate prices for used cars which will help car dealers to sell particular car. This price prediction will use previously mentioned factors that help in devising car price.

{1.3} **Review of Literature**

Literature review covers relevant literature with the aim of gaining insight into the factors that are important to predict the price of used cars in the market. In this study, we discuss various applications and methods which inspired us to build our supervised ML techniques to predict the price of used cars in different locations. We did a background survey regarding the basic ideas of our project and used those ideas for the collection of data information by doing web scraping from www.cars24.com website which is a web platform where seller can sell their used car. This project is more about data exploration, feature engineering and preprocessing that can be done on this data. Since we scrape huge amount of data that includes more car related features, we can do better data exploration and derive some interesting features using the available columns. Different techniques like ensemble techniques, and decision trees have been used to make the predictions. The goal of this project is to build an application which can predict the car prices with the help of other features. In the long term, this would allow people to better explain and reviewing their purchase with each other in this increasing digital world.

{1.4} **Motivation for the Problem Undertaken**

Deciding whether a used car is worth the posted price when you see listings online can be difficult. Several factors including mileage, engine displacement, running, 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. So, the main aim is to use machine learning algorithms to develop models for predicting used car prices.

- To build a ML model for forecasting value of a used vehicle based on multiple attributes.
- The system that is being built must be feature based i.e., feature wise prediction must be possible.
- Providing graphical comparisons to provide a better view.

[2] Analytical Problem Framing

{2.1} **Mathematical/ Analytical Modeling of the Problem**

We had to provide with an efficient machine learning model which will help car resellers or dealers to sell used cars with best price possible according to the changing market. This was done in 2 parts.

- → First part included vehicle data collection . This was done using web scrapping from www.cars24.com website for 8 locations. The locations were New Delhi, Mumbai, Bangalore, Hyderabad, Ahmedabad, Gurgaon, Chennai and Pune. The data included hatchbacks, sedans, SUV's and premium cars. This data also included car prices ranging from < INR 2 lakh to > INR 28 lakh.
- → Second part included using above data and cleaning it. This also included data manipulation, Exploratory data analysis, data preprocessing and model building. Best model was selected and was further tuned.

{2.2} **Data Sources and their formats**

Data is collected from www.cars24.com website which is web platform where sellers sell their cars and cars24 as a dealer sells them back to customers. The data was scrapped for 8 locations which are mentioned above using selenium web scrapping tool. Approximately more than 7000 cars data was scrapped for us to make a good machine learning model.

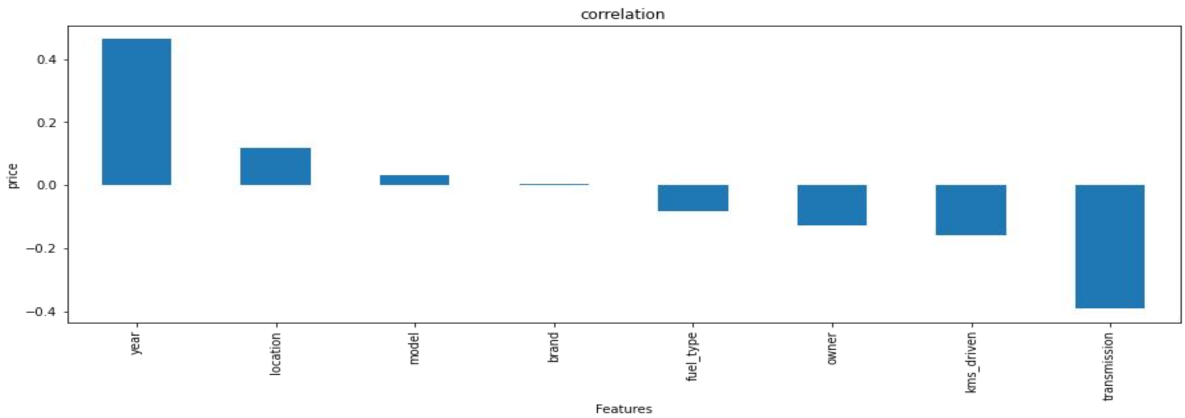
The scrapped data contains 10 columns which includes features such as brand, model, year,transmission, kilometers driven, owner, fuel type, price and location. The datatype was object type for all columns except year column which was showing integer. Since we had to predict prices for the cars, hence price column was considered as 'label' or target column.

Column name	Description
Brand	Car manufacturer name
Model	Name with type of model of car
Year	Year of car manufacture
Transmission	Type of transmission used in car(manual/automatic)
Kms_driven	Kilometers covered by car
owner	Number of owners of car
Fuel_type	Type of fuel used in car
price	Price set for a particular car
location	Location where car is up for sale

{2.3} **Data Preprocessing Done**

→ The scrapped dataset named 'cars24_data' was uploaded using pandas library.
→ The dataset had 7019 rows and 11 columns of which it contained 2 non useful columns such as 'Unnamed:0' and 'Unnamed:0.1'. These columns were dropped as this columns just contained index values for scrapped data.
→ Further the dataset was checked for any duplicate entries and it was found that there were 1219 duplicates present which were later removed. The new dataset shape was 5800 rows and 9 columns.
→ The model column had names of cars in form of "**['Grand', 'i10']**". In this case the outer [] brackets , single inverted commas and comma within 2 words had to be removed as part of data cleaning for whole column. This was done using replace method.
→ For kms_driven column, using replace method its comma present and km letters were removed.
→ As done earlier using replace method, rupee sign was removed from prices column.
→ Further the dataset was checked for null values using pandas method. It showed that there were 546 null values in transmission column and 426 in price column. Since this is fragile data which means if I input mode of transmission column (I.e. manual) and if null values are present in actiual automatic car, then the price prediction would have been lower instead of higher creating faulty model. Hence I did not used any inputting methods and dropped all the null values from dataset. After dropping null values the dataset shape was 4921 rows and 9 columns.
→ Later the datatype of kms_driven and price columns were changed from object to integer type.
→ Then later using visualization plots and graphs data analysis was explained.
→ Later the data was label encoded, and the column/s which contained outliers and skewness were checked and removed. The data loss percentage after removing outliers was 0.14%.
→ Since the columns were less. Feature selection methods were not used.
→ Further the data was split into features and label and features were standardized using Standard scalar method.
→ Further the model building was done and its performance was noted.

{2.4} **Data Inputs- Logic- Output Relationships**



The above graph shows the relation of features/independent variables with price of car. The above graph shows that year is highly positvely related to car price while transmission is highly negatively related to car price. The car brand column is least related to price of car. This is seen in other plots that as the age of car increases its price values also decreases. Also manual transmission vehicles have lesser value compared to automatic transmission cars.

The heatmap plotted suggested that there were no multicollinearity seen within features as well as with label.

{2.5} **Hardware and Software Requirements and Tools Used**

Hardware used to complete the analysis and model building:
Model: ASUS TUF A15
Processor: AMD RYZEN 5 4600H OCTA CORE
RAM: 8GB
ROM:500 GB SSD

Software used to complete the analysis and model building is:

Jupyter Notebook (via Anaconda Navigator) - used for data scrapping, data cleaning, all the analysis
and model building.
Chrome Browser- used to scrape data necessary for model building.

Libraries used:
→ Seaborn and matplotlib for visualization
→ Numpy
→ Pandas
→ Scipy.stats to import z score
→ Sklearn to import standard scaler, power transformer, Random Forest Regressor, AdaBoost Regressor, Decision Tree Regressor, Gradient Boosting Regressor, r2 score, train test split, mean absolute error, mean squared error, cross validation score and grid search cv.
→ Pickle

[3] Model/s Development and Evaluation

{3.1} Identification of possible problem-solving approaches (methods)

I have used both statistical and analytical approach in model building. I have used plots and graphs which shows the relation within features and with respect to target variable. Further data encoding was done using label encoder. Later the dataset was checked if its continuous columns except target variable has any outliers and skewness present or not. It was found that year column had outliers present which were later treated using z-score method. The same column had skewness present which was treated using numpy log transformation method.

The data checked if it had any multicollinearity within features, but it had none. Before model building part started dataset was divided into features and label and after that feature data was standardized for model building purpose. Since the target variable was price, which is a continuous kind of data, hence regression algorithms were used. Four algorithms were trained and tested on dataset and best model was selected. Later hyperparameter tuning was done on the best model based on its R2 score and RMSE value. In this case Random Forest regressor model was chosen as it had high R2 score and low RMSE value. Post hyperparameter tuning, the best model was saved from all 4models based on good R2 score and lower RMSE values.

{3.2} Testing of Identified Approaches (Algorithms)

The price being a continuous type column, it was known that it is going to be a regression kind of problem. Hence after standardizing (scaling) feature data, the data was then split into train test split. Further Random Forest regressor model was used to find best random state possible which gave best training and testing R2 score for model. Following is the list of algorithms that were used in model building:

→ 1. Random Forest regressor
→ 2. AdaBoost regressor
→ 3. Decision Tree regressor
→ 4. Gradient Boosting regressor

{3.3} **Run and Evaluate selected models**

```
rf=RandomForestRegressor()
ab=AdaBoostRegressor()
dt=DecisionTreeRegressor()
gbdt=GradientBoostingRegressor()
```

The above snapshot shows the regression algorithms being saved in variable names.

1) Random Forest Regressor Model

Random Forest model was used to find the best random state which gives highest r2 score. And it was further understood that random state 40 provided highest r2 score. Hence I took random state as 40 to evaluate all the remaining models.

```
#making train and test data split

x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.25,random_state=40)
```

```
rf.fit(x_train,y_train)
```

```
RandomForestRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=rf.predict(x_train)

pred=rf.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:98.43%
testing R2 score:92.98%
```

```
#cross validation score
print('Cross Validation Score for Random Forest regressor model :- ',((cross_val_score(rf,x_scaled,y,cv=17).mean())*100))
```

```
Cross Validation Score for Random Forest regressor model :-  90.3435502561012
```

```
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 50767.64551125576
root mean squared error 87277.81517312815
```

As per above snapshots it can be seen that Random Forest model gave training and testing R2 score of 98.43% and 92.98% with cross validation score of 90.34%. The mean absolute error and root mean square error is 50767.64 and 87277.81 respectively.

2) <u>AdaBosst Regressor Model</u>

```
ab.fit(x_train,y_train)
```

```
AdaBoostRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=ab.predict(x_train)

pred=ab.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:39.26%
testing R2 score:47.17%
```

```
#cross validation score
print('Cross Validation Score for AdaBoost regressor model :- ',((cross_val_score(ab,x_scaled,y,cv=17).mean())*100))
```

```
Cross Validation Score for AdaBoost regressor model :-  27.06511296935603
```

```
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 203502.24884886426
root mean squared error 239430.56028211128
```

As per above snapshot it can be seen that AdaBosst model gave training and testing R2 score of 39.26% and 47.17% with cross validation score of 27.06%. The mean absolute error and root mean square error is 203502.24 and 239430.56 respectively.

3) <u>Decision Tree Regression Model</u>

```
dt.fit(x_train,y_train)
```

```
DecisionTreeRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=dt.predict(x_train)

pred=dt.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:100.00%
testing R2 score:90.47%
```

```
#cross validation score
print('Cross Validation Score for Decision regressor model :- ',((cross_val_score(dt,x_scaled,y,cv=17).mean())*100))
```

```
Cross Validation Score for Decision regressor model :-  83.80270879805306
```

```
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 56863.384865744505
root mean squared error 101694.71020963501
```

As per above snapshots it can be seen that Decision Tree model gave training and testing R2 score of 100% and 90.47% with cross validation score of 83.80%. The mean absolute error and root mean square error is 56863.38 and 101694.71 respectively.

4) <u>Gradient Boosting Regresson Model</u>

```
gbdt.fit(x_train,y_train)
```

```
GradientBoostingRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=gbdt.predict(x_train)

pred=gbdt.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:84.83%
testing R2 score:85.45%
```

```
#cross validation score
print('Cross Validation Score for Gradient Boosting regressor model :- ',((cross_val_score(gbdt,x_scaled,y,cv=17).mean())*10
```

```
Cross Validation Score for Gradient Boosting regressor model :-  81.61435779736496
```

```
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 81581.03192984672
root mean squared error 125653.95103241253
```

As per above snapshots it can be seen that Gradient Boosting model gave training and testing R2 score of 84.83% and 85.45% with cross validation score of 81.61%. The mean absolute error and root mean square error is 81581.03 and 125656.95 respectively.

Looking at all the models it can be seen Random Forest model had better training and testing R2 score compared to other 3 models. Its difference between cross validation score and testing R2 score was lower as well as its RMSE score was also lower compared to other models. Hence Random Forest model was selected as best model to be used for hyperparameter tuning.

Hyperparameter tuning was done using GridSearch CV method.

5) Hyperparameter tuning on Random Forest model.

```
#setting parameters for tuning
grid_param=[{'criterion':["squared_error", "absolute_error", "poisson"],
            'min_samples_split':[2,2.5,3,3.5,4],
             'max_depth':[1,2,3,4,5],
             'bootstrap':[True,False]}]
```

```
grid=GridSearchCV(rf,param_grid=grid_param)
```

```
grid.fit(x_train,y_train)
```

```
GridSearchCV(estimator=RandomForestRegressor(),
             param_grid=[{'bootstrap': [True, False],
                          'criterion': ['squared_error', 'absolute_error',
                                        'poisson'],
                          'max_depth': [1, 2, 3, 4, 5],
                          'min_samples_split': [2, 2.5, 3, 3.5, 4]}])
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
#best parameters for tuning
grid.best_params_
```

```
{'bootstrap': True,
 'criterion': 'squared_error',
 'max_depth': 5,
 'min_samples_split': 3}
```

The above mentioned list of parameters were specified based on Random Forest algorithm and grid search cv trained and used those parameters to find out which combination of parameters from them would provide best R2 score. Based on this best parameters were printed as shown in above snapshot.

```
#using best parameters to train

rf1=RandomForestRegressor(criterion='squared_error'
                          ,min_samples_split=3,
                          max_depth=5,
                          bootstrap=True)
```

```
rf1.fit(x_train,y_train)
```

```
RandomForestRegressor(max_depth=5, min_samples_split=3)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=rf1.predict(x_train)

pred=rf1.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:64.27%
testing R2 score:69.25%
```

```
#cross validation score
print('Cross Validation Score for Random forest tuned model :- ',((cross_val_score(rf1,x_scaled,y,cv=17).mean())*100))
```

```
Cross Validation Score for Random forest tuned model :-  62.27580920105422
```

```
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 127350.5450237147
root mean squared error 182659.95770899273
```

As shown in above snapshot, the best parameters were then used to train random forest algorithm and it gave training and testing R2 score of 64.27% and 69.25% with cross validation score as 62.27%. The mean absolute error and root mean square error was 127350.54% and 182659.95% respectively.

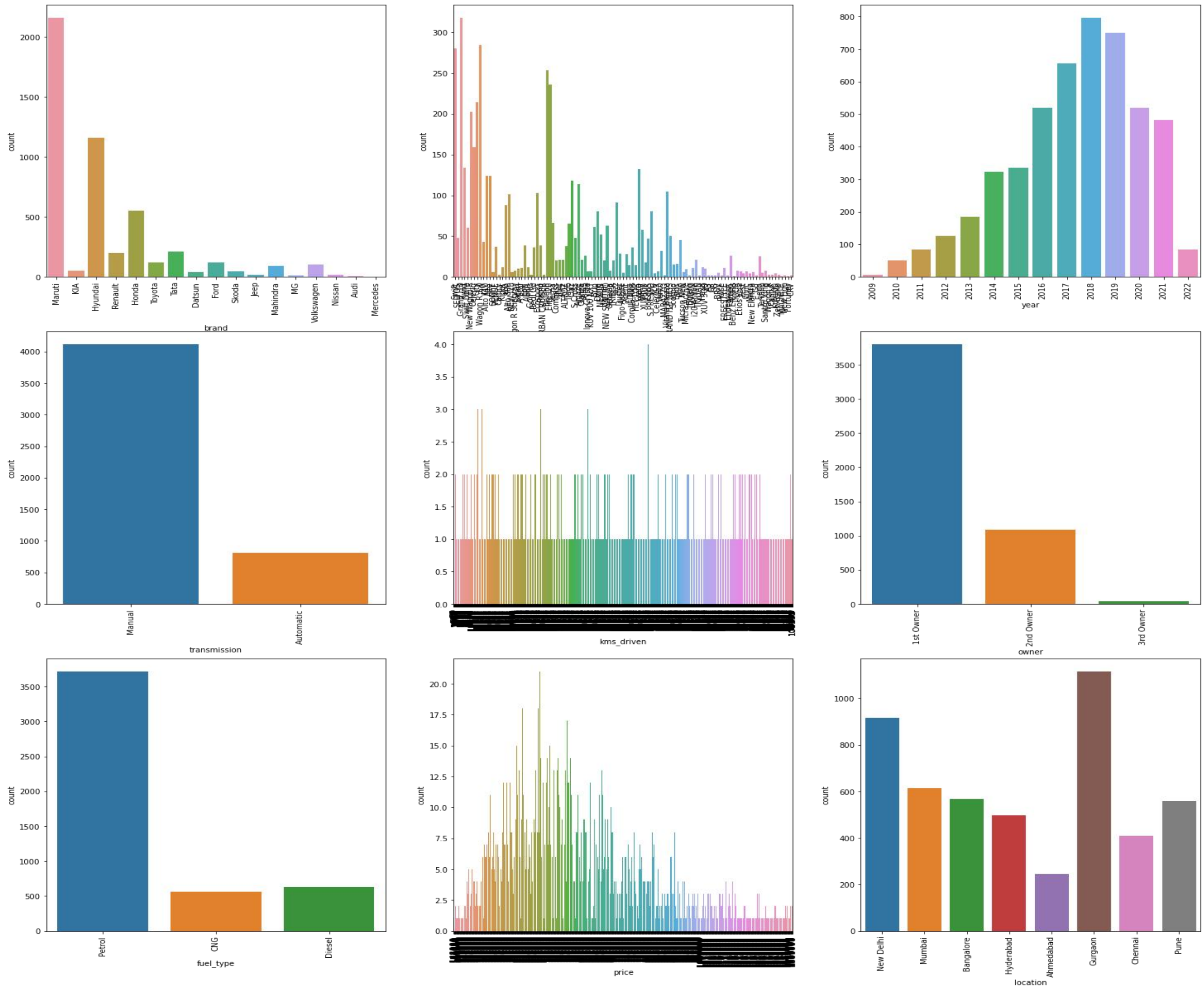
It can be seen that the tuned model does not give good R2 score, hence the original Random Forest model was saved as the best model.

{3.4} Key Metrics for success in solving problem under consideration

The key metrics used in this models are:

- → R2 score- This is the most important metrics in determining the accuracy of the model. It is a statistical measure that represents the goodness of fit of a regression model. The ideal value for R2 Score is 1 or 100%. The closer the value of R2 Score to 1, the better is the model fitted.
- → Cross validation (CV) score: This is used on models to check if the model is overfitting or not. If the testing R2 score and CV score is equal or near equal then model is said to be best fitted model or else overfit model.
- → MAE (Mean Absolute error): Represents average error i.e. error for every single data point and takes its average. This error is lesser the better the model. Zero MAE means best model.
- → RMSE(Root Mean Squared Error): This is similar to MAE but noise is exaggerated and larger error are punished. This is the main metric used in interpreting model. RMSE close or equal to zero means best model.

{3.5}**Visualizations**



<u>Observations for above plots</u>

→ Incase of car brands, The most number of cars that are for sale are from Maruti brand while least are from Mercedes brand.

→ Incase of model of cars, Grand i10 is the most available car for sale followed by Wagonr and swift.

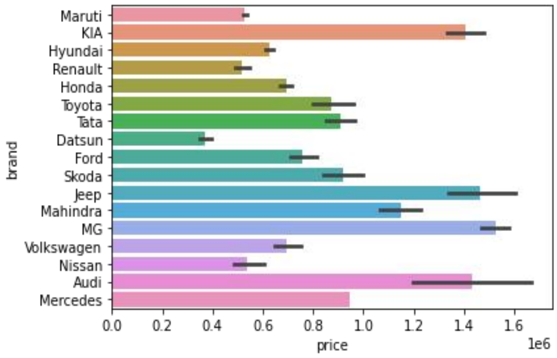
→ incase of years of cars, most cars for sale are from the year 2018 followed by 2019 and 2017. The least cars for sale are from the year 2009.

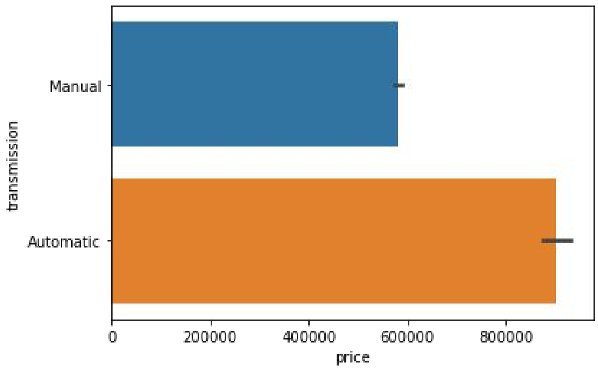
→ Incase of transmission, most number of cars for sale are with manual transmission while very less cars are with automatic transmission.

→ Most number of cars are owned by single person, while approx. 1000 cars had 2 owners in its lifetime while some cars were owned by 3 people.

→ Most number of used cars for sale run on petrol while very less number of cars for sale run on cng and diesel.

- Most number of cars for sale are from Gurgaon region while less number of cars for sale are from Ahmdedabad region.

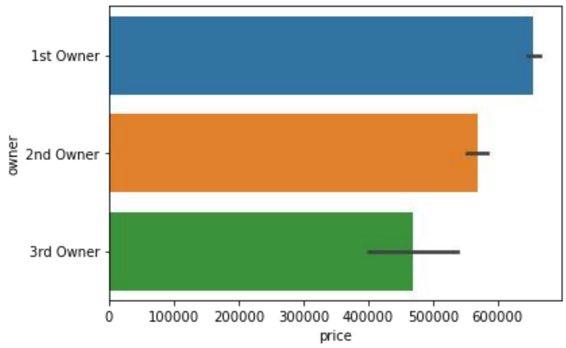


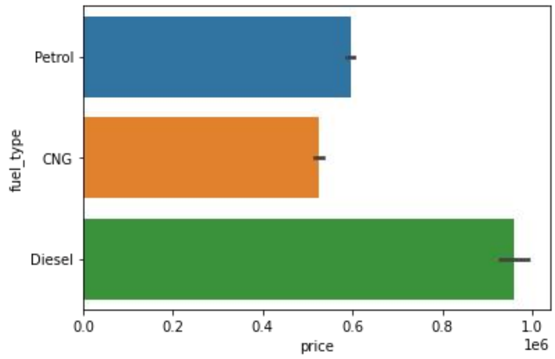


<u>Observation for above plots</u>:

→ It can be seen from the plot that though as seen earlier Maruti has highest number of cars for sale but their unit car price is low compared to other brands. MG brand has their cars priced higher for resale, followed by Jeep, Audi and Kia.

→ Automatic cars are priced higher compared to manual transmission cars.

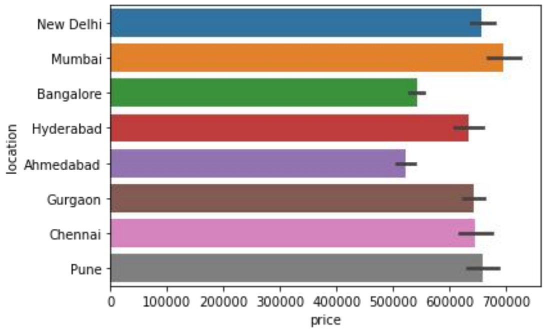


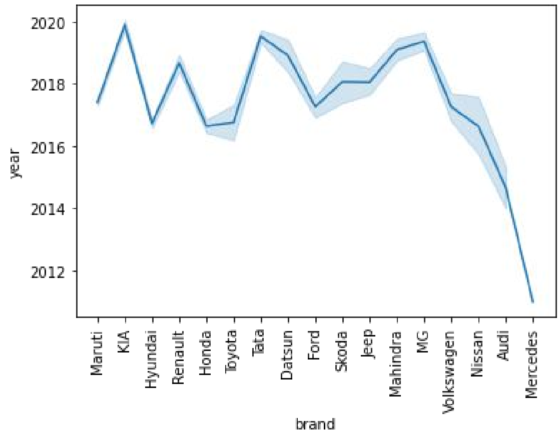


<u>Observation for above plots</u>

→ the cars that are owned by 3 owners are priced lower compared to cars owned by 2 and single owner.

→ The diesel powered cars are priced higher followed by petrol cars. The cheapest cars are with CNG fuel.

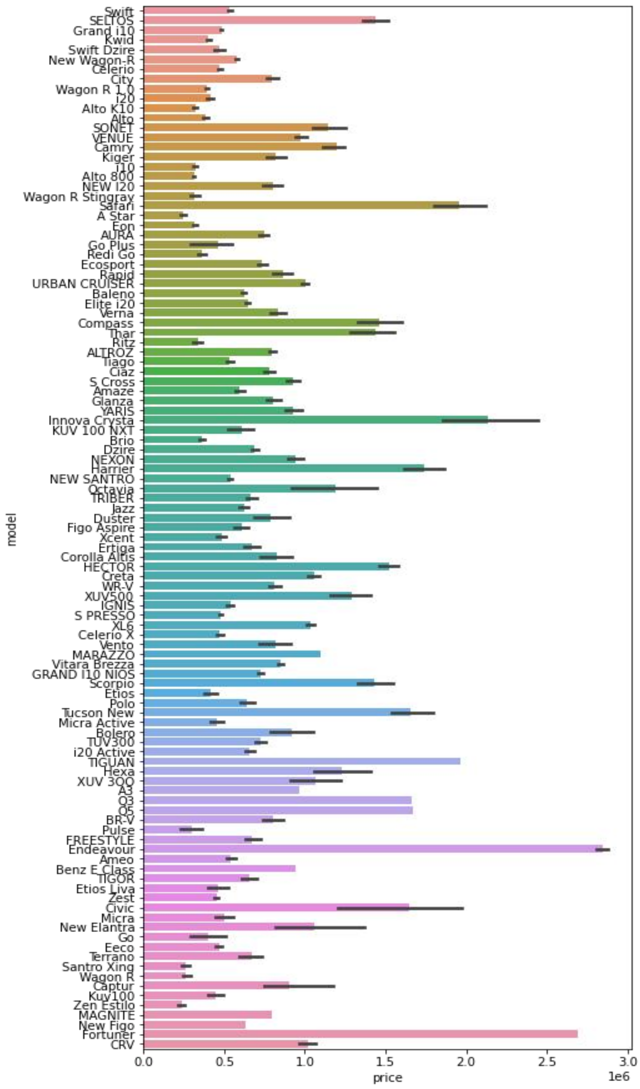




<u>Observations for above plots</u>

→ It can be seen from the plot that cars from Mumbai are priced higher compared to other cities or locations.

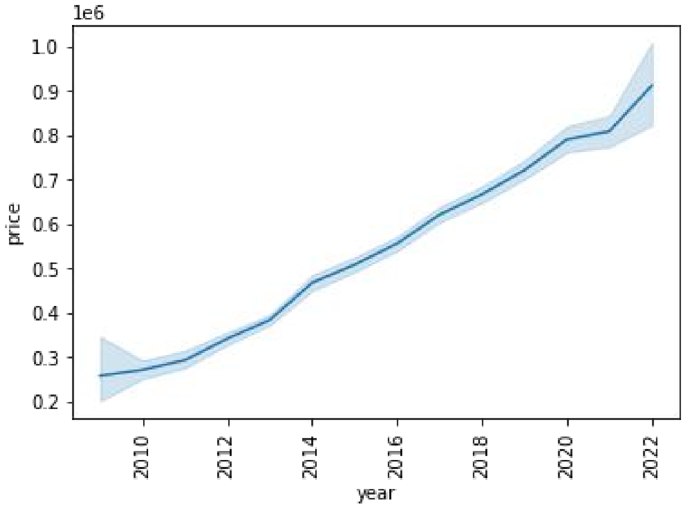
→ It can be seen that Maruti, Tata and MG brand cars for sale are mostly new cars while cars from brands like Nissan, Audi and Mercedes are older ones.



<u>Observation for plot:</u>

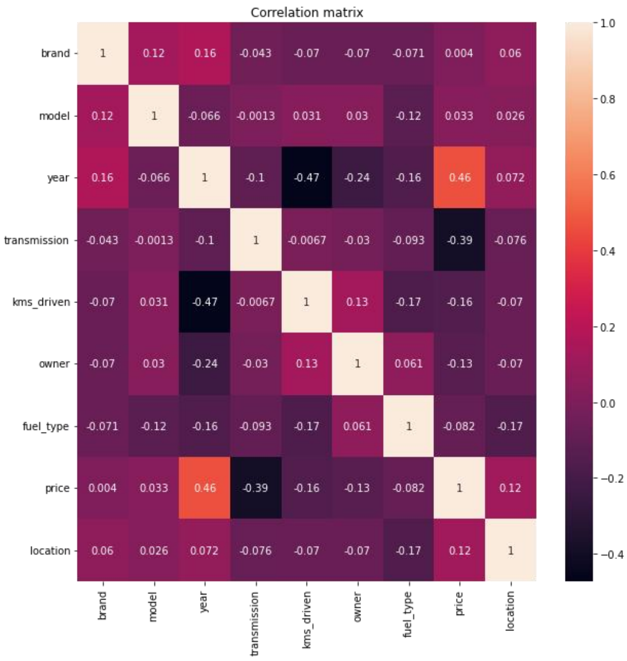
It is seen that Endeavour, Fortuner, Innova Crysta and Safari cars are priced higher compared to other cars. These are actually big premium SUV cars.

It is also seen that price of smaller hatchback cars are priced low.

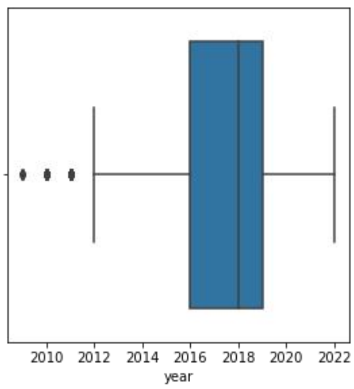


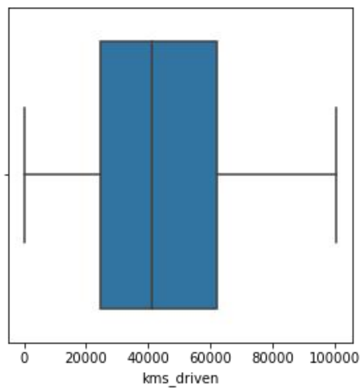
<u>Observation for above plot</u>

→ It can be seen that the cars which are new are priced higher compared to cars that are old. The oldest cars from year 2009 are priced lowest.



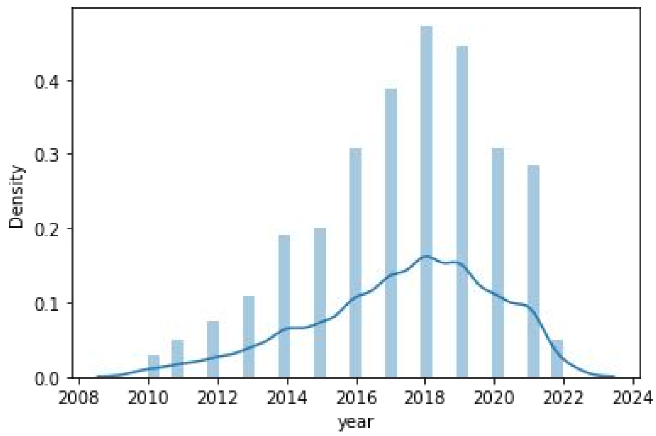
The heatmap shows that no multicollinearity seen within features.





→ The boxplots plotted shows that in year column there is presence of outliers while in kms_driven column there are none.

→ The outliers present were further removed using z-score method



- → The displot shows that data in year column was left skewed indicating presence of skewness.
- → The skewness present was further removed with the help of power transformer method.

[4] CONCLUSION

{4.1} **Key Findings and Conclusions of the Study**

In this project report, I have used ML algorithms to predict car prices. I have mentioned the step by step procedure to analyze the dataset and finding the correlation between the features. The features and label were then subjected to 4 algorithms. Undergoing that I was able to find best model based on its performance using different metrics mentioned earlier. Further best model was then tuned for hyperparameter tuning. Further best model was selected and saved using pickle.

The main conclusions that can be derived from this study are :

→ The car prices are inversely proportional to age of cars which means that as the age of the cars increases their resale value decreases. This is because most of the people want to buy cars that are as new as possible so they don't have to pay more on their servicing costs as happens for older cars.

→ Since the bigger and premium cars have more features present, hence these cars have higher resale value compared to small cars.

→ Diesel being a cheaper fuel option in India and also providing greater fuel economy plus automatic cars giving ease of driving in city as well as highway conditions. Hence cars with automatic transmission and diesel fuel type have higher resale value compared to manual and petrol cars

→ If the car is previously owned by 2 or 3 owners its resale values decreases compared. The reason according to me is that person tend to sell their car if the car is having any mechanical problems or failures. Hence the 2nd person owning the car feels the same and sells it. Hence it value decreases over the time.

{4.2} **Limitations of this work and Scope for Future Work**

Limitations: The main limitation of this study is the low number of records that have been used. In the dataset, the data is not properly distributed in one or more columns. Many of the values in the had to be replaced to remove their brackets and commas as mentioned earlier while presence of null values and high number of duplicates was also a big issue. So, because of that data our models may not make the right patterns and the performance of the model also reduces.This issues need to be taken care of.

Scope for future work: As future work, we intend to collect more data and to use more advanced techniques like artificial neural networks to predict car prices. In future this machine learning model may bind with various website which can provide real time data for price prediction. Also, we may add large historical data of car price which can help to improve accuracy of the machine learning model.