FLIP ROBO

Car Price Prediction

Submitted by:

Ankita Ramdas Mhetre

ACKNOWLEDGMENT

I would like to express my deep and sincere gratitude towards Fliprobo technologies for providing me the internship opportunity and a great chance for learning and professional development.

I express my deepest gratitude and special thanks to the subject matter expert (SME), Mr Shubham Yadav who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on correct path ,motivated me for taking part in useful decision & giving necessary advices to do the project and providing invaluable guidance throughout.

He was never too busy to spare his valuable time for this work. No words are adequate to express my gratitude towards him. I would also like to thank him for his friendship and empathy.

I would also like to thank Khushboo Garg for her dynamism, vision; sincerity and motivation have deeply inspired me. She has taught me the methodology to carry out the task and to present the project works as clearly as possible. It was a great privilege and honour to work and study under her guidance. I am extremely grateful for what she has offered me.

INTRODUCTION

- ## Business Problem Framing

With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model.

This project contains two phase Data Collection Phase – we have to scrape at least 5000 used cars data. We can scrape more data as well, it's up to us. More the data better the model. In this section .We need to scrape the data of used cars from websites (Olx, cardekho, Cars24 etc.) We will need web scraping for this. We will fetch data for different locations. Model Building Phase-After collecting the data, we   need to build a machine learning model. Before model building we will do all data pre-processing steps. Try different models with different hyper parameters and select the best model.

Conceptual Background of the Domain Problem

Predicting the resale value of a car is not a simple task. The value of used cars depends on a number of factors. The most important ones are usually the age of the car, its make (and model), the origin of the car (the original country of the manufacturer), its mileage (the number of kilometres it has run) and its horsepower. Due to rising fuel prices, fuel economy is also of prime importance. Unfortunately, in practice, most people do not know exactly how much fuel their car consumes for each km driven. Other factors such as the type of fuel it uses, the interior style, the braking system, acceleration, the volume of its cylinders (measured in cc), safety index, its size, number of doors, paint colour, weight of the car, consumer reviews, prestigious awards won by the car manufacturer, its physical state, whether it is a sports car, whether it has cruise control, whether it is automatic or manual transmission, whether it belonged to an individual or a company and other options such as air conditioner, sound system, power steering, cosmic wheels, GPS navigator all may influence the price as well. Some special factors which buyers attach importance in are, whether the car had been involved in serious accidents and whether it is a lady-driven car. The look and feel of the car certainly contributes a lot to the price. As we can see, the price depends on a large number of factors. 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, we have considered only a small subset of the factors mentioned above.

Review of Literature

Predicting the price of used cars in both an important and interesting problem. According to data obtained from the National Transport Authority the number of cars registered between 2003 and 2013 has witnessed a spectacular increase of 234%. From 68, 524 cars registered in 2003, this number has now reached 160, 701. With difficult economic conditions, it is likely that sales of second-hand imported (reconditioned) cars and used cars will increase. It is reported by Motors mega that the sales of new cars has registered a decrease of 8% in 2013. In many developed countries, it is common to lease a car rather than buying it outright. A lease is a binding contract between a buyer and a seller (or a third party – usually a bank, insurance firm or other financial institutions) in which the buyer must pay fixed instalments for a pre-defined number of months/years to the seller/financer. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e. its expected resale value. Thus, it is of commercial interest to seller/financers to be able to predict the salvage value (residual value) of cars with accuracy. If the residual value is under-estimated by the seller/financer at the beginning, the instalments will be higher for the clients who will certainly then opt for another seller/financer. If the residual value is over-estimated, the instalments will be lower for the clients but then the seller/financer may have much difficulty at selling these high-priced used cars at this over-estimated residual value. Thus, we can see that estimating the price of used cars is of very high commercial importance as well. Manufacturers' from Germany made a loss of 1 billion Euros in their USA market because of miscalculating the residual value of leased cars. Most individuals in India who buy new cars are also very apprehensive about the resale value of their cars after certain number of years when they will possibly sell it in the used cars market.

Surprisingly, work on estimating the price of used cars is very recent but also very sparse. MSc thesis,Listiani in her project work showed that the regression mode build using support vector machines (SVM) can estimate the residual price of leased cars with higher accuracy than simple multiple regression or multivariate regression. SVM is better able to deal with very high dimensional data (number of features used to predict the price) and can avoid both over-fitting and under fitting. In particular, she used a genetic algorithm to find the optimal parameters for SVM in less time. The only drawback of this study is that the improvement of SVM regression over simple regression was not expressed in simple measures like mean deviation or variance.

In A thesis on determinants of used car resale value by Richardson , the hypothesis states that car manufacturers are more willing to produce vehicles which do not depreciate rapidly. In particular, by using a multiple regression analysis, he showed that hybrid cars (cars which use two different power sources to propel the car, i.e. they have both an internal combustion engine and an electric motor) are more able to keep their value than traditional vehicles. This is likely due to more environmental concerns about the climate and because of its higher fuel efficiency. The importance of other factors like age, mileage, make and MPG (miles per gallon) were also considered in this study. He collected all his data from various websites.

Wu et al. used neuro-fuzzy knowledge based system to predict the price of used cars. Only three factors namely: the make of the car, the year in which it was manufactured and the engine style were considered in this study. The proposed system produced similar results as compared to simple regression methods.

Car dealers in USA sell hundreds of thousands of cars every year through leasing. Most of these cars are returned at the end of the leasing period and must be resold. Selling these cars at the right price have major economic connotation for their success. In response to this, the ODAV system (Optimal Distribution of Auction Vehicles) was developed by Du et al. This system not only estimates a best price for reselling the cars but also provides advice on where to sell the car. Since the United States is a huge country, the location where the car is sold also has a non-trivial impact on the selling price of used cars. A k-nearest neighbour regression model was used for forecasting the price. Since this system was started in 2003, more than two million vehicles have been distributed via this system.

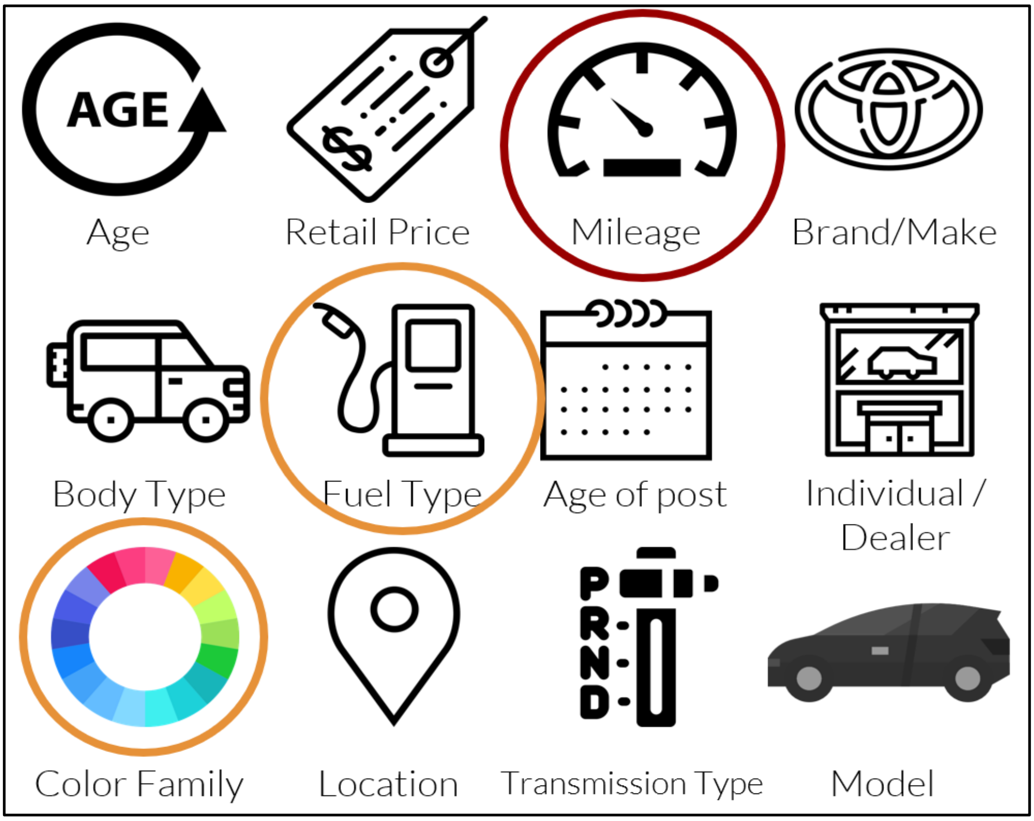
Gonggi [GONGGI, S., 2011. New model for residual value prediction of used cars based on BP neural network and non-linear curve fit. In: Proceedings of the 3 rd IEEE International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Vol 2. pp. 682-685, IEEE Computer Society, Washington DC, USA.] proposed a new model based on artificial neural networks to forecast the residual value of private used cars. The main features used in this study were: mileage, manufacturer and estimate useful life. The model was optimised to handle nonlinear relationships which cannot be done with simple linear regression methods. It was found that this model was reasonably accurate in predicting the residual value of used cars.

- ## Motivation for the Problem Undertaken

With the prohibitive cost of car ownership in mind, analysis of data from the used car market may potentially offer savings and gains for the car portal, the consumer and car dealers.

Imagine a situation where you have an old car and want to sell it. You may of course approach an agent for this and find the market price, but later may have to pay pocket money for his service in selling your car. But what if you can know your car selling price without the intervention of an agent? Or if you are an agent, definitely this will make your work easier. Yes, this system has already learned about previous selling prices over years of various cars.

So, to be clear, this deployed web application will provide you with the approximate selling price for your car based on the fuel type, years of service, showroom price, the number of previous owners, kilometres driven, if dealer/individual, and finally if the transmission type is manual/automatic. And that's a brownie point.



An accurate used car price evaluation is a catalyst for the healthy development of used car market. Data mining has been applied to predict used car price in several articles. However, little is studied on the comparison of using different algorithms in used car price estimation. This project collects more than 5,000 used car dealing records throughout India to do empirical analysis on a thorough comparison of different algorithms like: linear regression, random forest, Gradient Boosting Regressor and XGBoost Regression

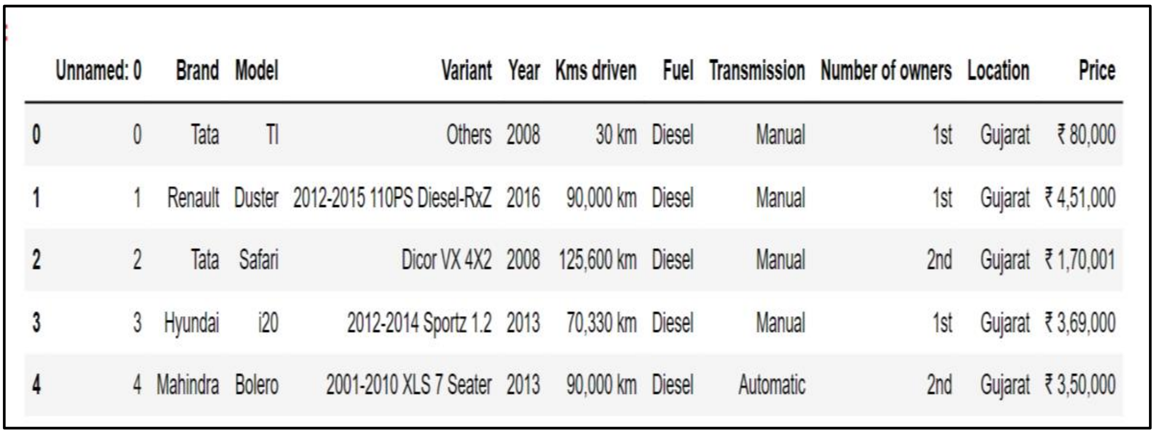
Analytical Problem Framing

- ## Data Sources and their formats

The complete project is divided into 2 segments:

1) Data Collection: in this phase the data of used cars is collected from an online website olx.com.
   Features such as Brand, model, Variant, fuel type, transmission, Number of owners, kilometres driven and Price are collected using web scraping technique.
2) Model Building: in this phase the collected data is pre-processed, redundant and unnecessary records are deleted and a machine learning model is built to predict the car selling price.

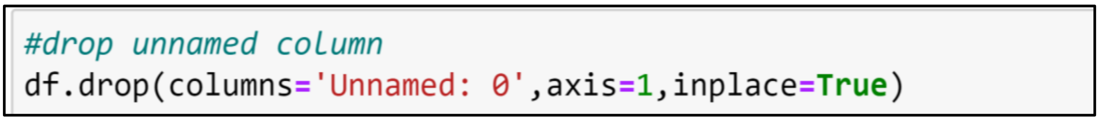
   The shape of data collected was 6069 rows and 10 columns. An instance of data is shown below:



	Unnamed: 0	Brand	Model	Variant	Year	Kms driven	Fuel	Transmission	Number of owners	Location	Price
0	0	Tata	TI	Others	2008	30 km	Diesel	Manual	1st	Gujarat	₹ 80,000
1	1	Renault	Duster	2012-2015 110PS Diesel-RxZ	2016	90,000 km	Diesel	Manual	1st	Gujarat	₹ 4,51,000
2	2	Tata	Safari	Dicor VX 4X2	2008	125,600 km	Diesel	Manual	2nd	Gujarat	₹ 1,70,001
3	3	Hyundai	i20	2012-2014 Sportz 1.2	2013	70,330 km	Diesel	Manual	1st	Gujarat	₹ 3,69,000
4	4	Mahindra	Bolero	2001-2010 XLS 7 Seater	2013	90,000 km	Diesel	Automatic	2nd	Gujarat	₹ 3,50,000

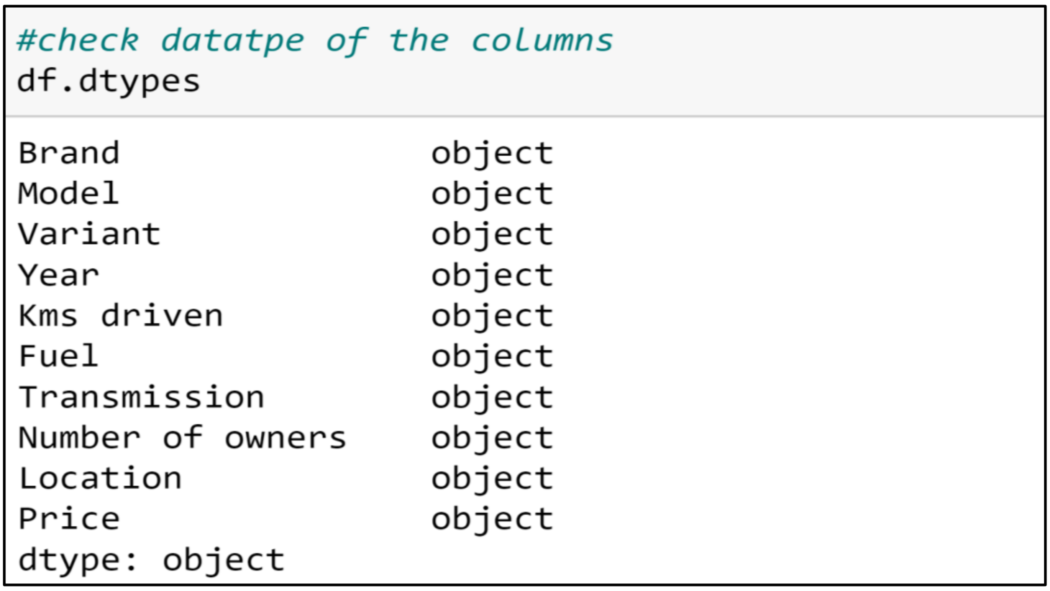
- **Data Pre-processing Done**

1) The dataset had a total of 6069 rows and 11 columns, out of which the unnamed column was deleted as it was just the indexing.



```
#drop unnamed column
df.drop(columns='Unnamed: 0',axis=1,inplace=True)
```

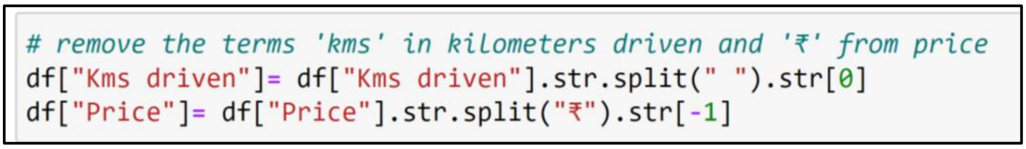
2) Initially all the columns were in object data type so the very first step was to convert the data types for numeric column.



```
#check datatpe of the columns
df.dtypes
```

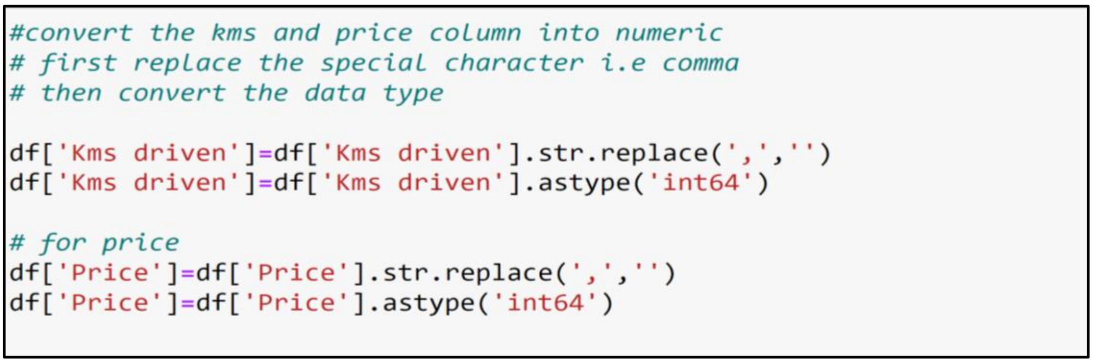
```
Brand                object
Model                object
Variant              object
Year                 object
Kms driven           object
Fuel                 object
Transmission         object
Number of owners     object
Location             object
Price                object
dtype: object
```

3) As the price column had a rupee symbol and kms had kms written after each value, we initially cleaned this symbols and prefixes.



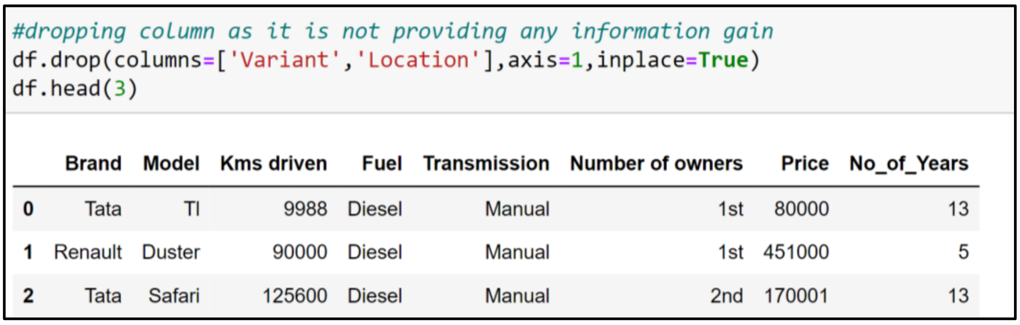
```
# remove the terms 'kms' in kilometers driven and '₹' from price
df["Kms driven"]= df["Kms driven"].str.split(" ").str[0]
df["Price"]= df["Price"].str.split("₹").str[-1]
```

4) After removal of special characters data conversion is done. Kms driven and Price column were converted from object to integer data type.



```
#convert the kms and price column into numeric
# first replace the special character i.e comma
# then convert the data type

df['Kms driven']=df['Kms driven'].str.replace(',','')
df['Kms driven']=df['Kms driven'].astype('int64')

# for price
df['Price']=df['Price'].str.replace(',','')
df['Price']=df['Price'].astype('int64')
```

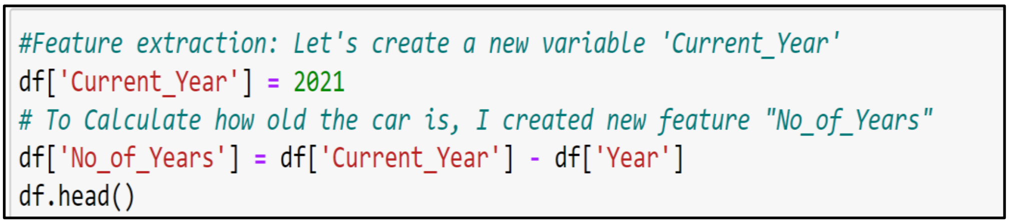
5) Variant and location columns were dropped. Variant had too many unique values and the car prices did not vary with the Location hence the decision to drop those two columns.



```
#dropping column as it is not providing any information gain
df.drop(columns=['Variant','Location'],axis=1,inplace=True)
df.head(3)
```

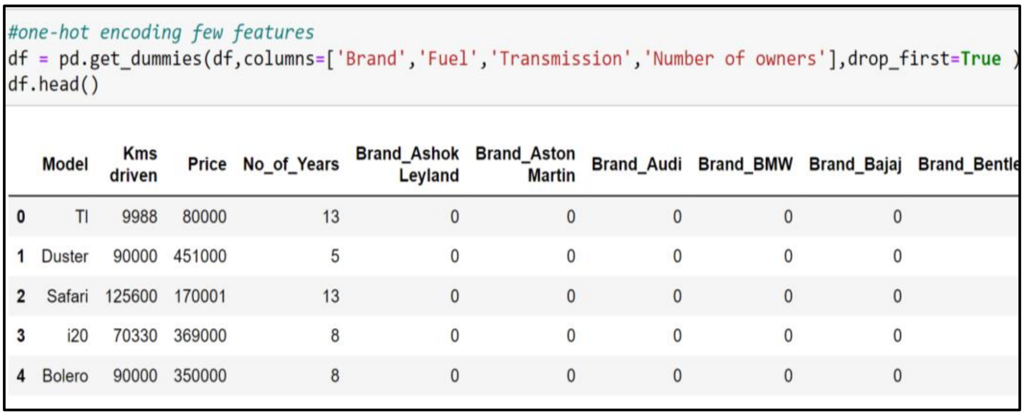
	Brand	Model	Kms driven	Fuel	Transmission	Number of owners	Price	No_of_Years
0	Tata	TI	9988	Diesel	Manual	1st	80000	13
1	Renault	Duster	90000	Diesel	Manual	1st	451000	5
2	Tata	Safari	125600	Diesel	Manual	2nd	170001	13

6) For feature extraction, the year column is subtracted from the current year i.e. 2021 which gave us the age of car



```
#Feature extraction: Let's create a new variable 'Current_Year'
df['Current_Year'] = 2021
# To Calculate how old the car is, I created new feature "No_of_Years"
df['No_of_Years'] = df['Current_Year'] - df['Year']
df.head()
```

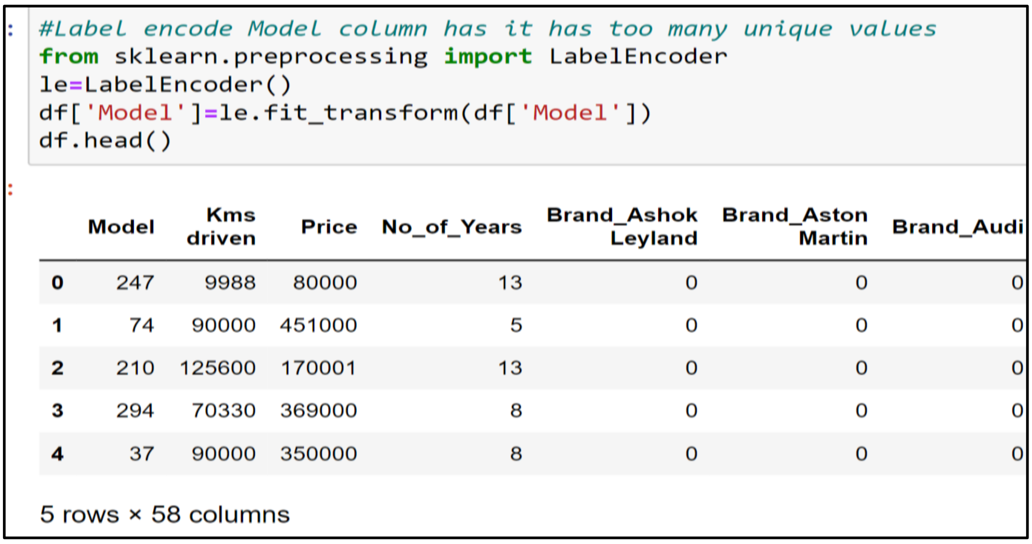
7) One hot encoding was carried out on other categorical columns.



```
#one-hot encoding few features
df = pd.get_dummies(df,columns=['Brand','Fuel','Transmission','Number of owners'],drop_first=True )
df.head()
```

	Model	Kms driven	Price	No_of_Years	Brand_Ashok Leyland	Brand_Aston Martin	Brand_Audi	Brand_BMW	Brand_Bajaj	Brand_Bentl
0	TI	9988	80000	13	0	0	0	0	0	
1	Duster	90000	451000	5	0	0	0	0	0	
2	Safari	125600	170001	13	0	0	0	0	0	
3	i20	70330	369000	8	0	0	0	0	0	
4	Bolero	90000	350000	8	0	0	0	0	0	

8) “Model” was passed through Label Encoder.

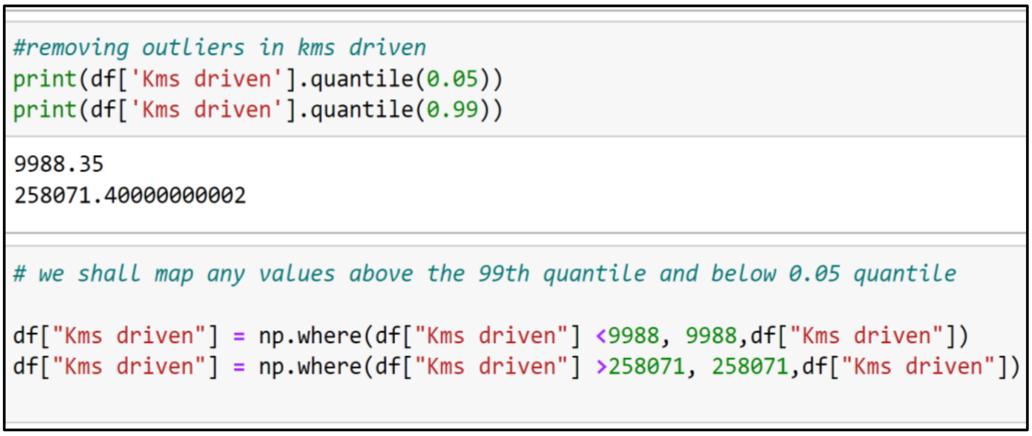


```
#Label encode Model column has it has too many unique values
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['Model']=le.fit_transform(df['Model'])
df.head()
```

	Model	Kms driven	Price	No_of_Years	Brand_Ashok Leyland	Brand_Aston Martin	Brand_Audi
0	247	9988	80000	13	0	0	0
1	74	90000	451000	5	0	0	0
2	210	125600	170001	13	0	0	0
3	294	70330	369000	8	0	0	0
4	37	90000	350000	8	0	0	0

5 rows × 58 columns

9) As the numeric features showed high skewness we did quantile based flooring and capping for both kilometres driven and number of years column.

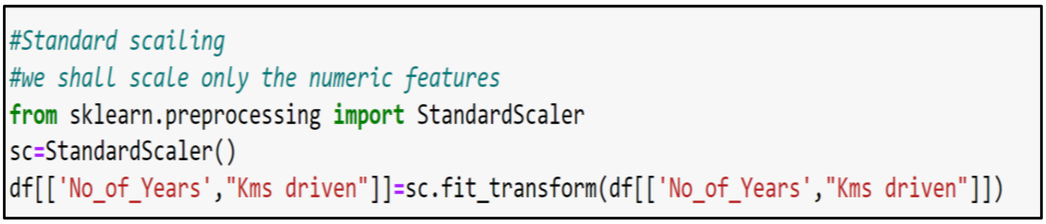


```
#removing outliers in kms driven
print(df['Kms driven'].quantile(0.05))
print(df['Kms driven'].quantile(0.99))
```

```
9988.35
258071.40000000002
```

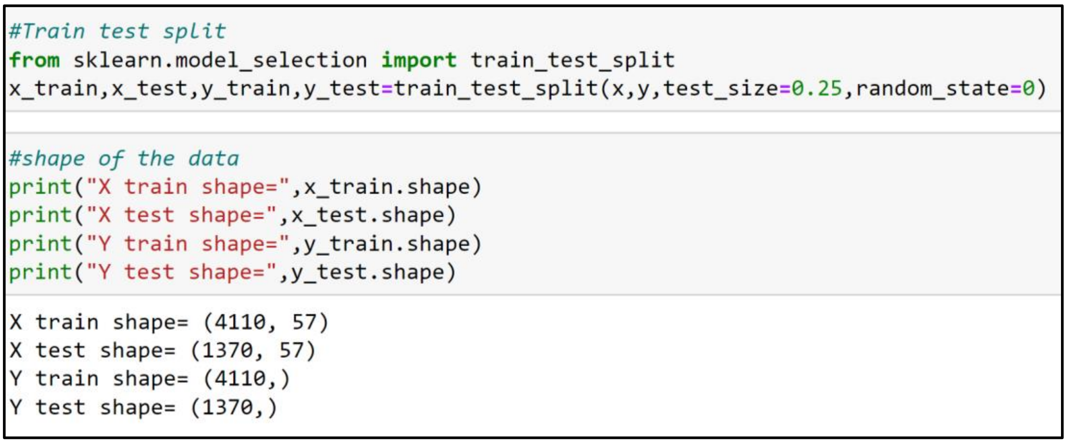
```
# we shall map any values above the 99th quantile and below 0.05 quantile

df["Kms driven"] = np.where(df["Kms driven"] <9988, 9988,df["Kms driven"])
df["Kms driven"] = np.where(df["Kms driven"] >258071, 258071,df["Kms driven"])
```

The numeric columns kilometres driven and year were scaled using standard scalar to avoid preferences to higher numbers in kilometres.



```
#Standard scailing
#we shall scale only the numeric features
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
df[['No_of_Years',"Kms driven"]]=sc.fit_transform(df[['No_of_Years',"Kms driven"]])
```

At the end the data was passed through train test split with a test size of 25 % and train size of 75 % with 0 random state .



```
#Train test split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

```
#shape of the data
print("X train shape=",x_train.shape)
print("X test shape=",x_test.shape)
print("Y train shape=",y_train.shape)
print("Y test shape=",y_test.shape)
```

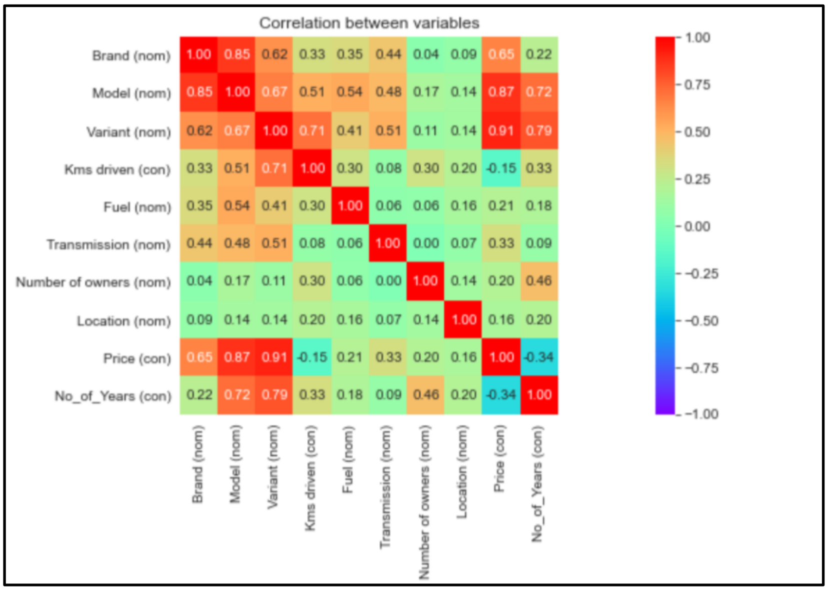
```
X train shape= (4110, 57)
X test shape= (1370, 57)
Y train shape= (4110,)
Y test shape= (1370,)
```

Various machine learning models were then implemented and evaluation metrics for each model was studied respectively.

- ## Data Inputs- Logic- Output Relationships

Describe the relationship behind the data input, its format, the logic in between and the output. Describe how the input affects the output.

The relationship between various features and target is visualized through the heatmap which shows the correlation between the two.



The highest correlation of target was with 'Variant'=0.91. The second highest correlation was 0.87 which was between Model and the price.

Brand also had a good correlation with target (0.65). Kilometres driven and age of the car i.e. number of years had a negative correlation with cars selling price. This indicates that for higher Kilometres driven the car prices were low. Same goes for Number of years, the older the car the lesser the selling price of car.

Other features like number of owners, fuel, transmission and location has moderate positive correlation.

• Hardware and Software Requirements and Tools Used

Open source web-application used for programming:

1. **Jupyter Notebook**

Python Libraries / Packages used were:

1. **Pandas**: pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

   We have used pandas to import the csv file using pd.read_csv all data analysis have been done using the pandas and numpy libraries. The data characteristics have been studied using pandas functions like df.shape(), df .dtypes, df.columns etc.

2. **NumPy:** NumPy is an open-source numerical Python library. NumPy contains a multi-dimensional array and matrix data structures. It can be utilised to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic.

3. **Matplotlib**: library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

The Matplotlib libraries pyplot function is used for making plots ,plt.show() ,plt.figure(figsize) that has been used is a part of matplotlib library.

4. **Seaborn**: Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

   All the visualizations made are built using the seaborn library. Alias used for seaborn is sns.

   Sns.boxplot(), sns.heatmap(), sns.distplot(), sns.scatterplot() ,sns.stripplot, ,heatmap are few of the libraries used

5. **Sklearn:** Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, and clustering and dimensionality reduction via a consistence interface in Python.
   All the Machine Learning regression algorithms have been imported from the sklearn package. Simple imputer used is also a part of sklearn.

   The evaluation metrics, RMSE,MSE,MAE functions are also imported from same.

6. **Dython:** is a set of Data analysis tools in python 3.x. which gives us measures of association for categorical features, Plots features correlation and association for mixed data-sets (categorical and continuous features) in an easy and simple way.

Model/s Development and Evaluation

• Identification of possible problem-solving approaches (methods)

The approach followed can be described briefly as:

1. Data Cleaning (Identifying null values, data conversion and removing outliers).
2. Data Pre-processing (Standardization or Normalization, one hot encoding and label encoder)
3. Data Visualization for Gaining some insights from data
4. ML Models were implemented: Linear Regression, Random Forest Regressor, Adaboost Regressor, and XGBoost
5. Comparison of the performance of the models

• Testing of Identified Approaches (Algorithms)

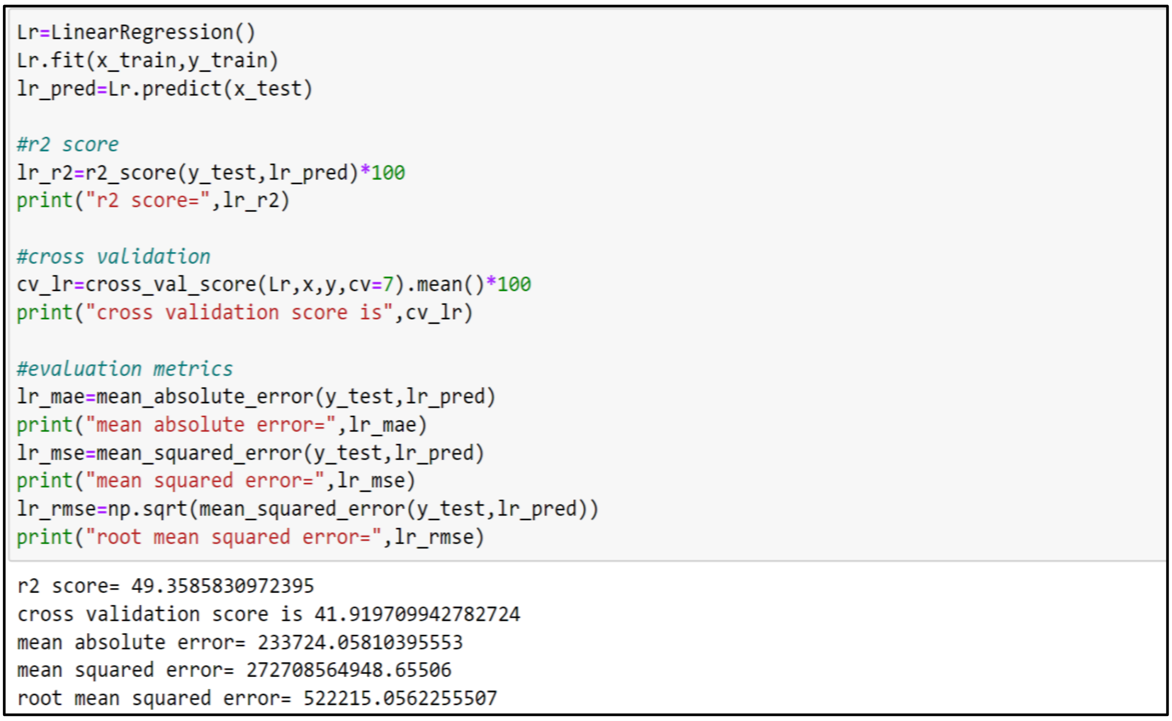
Following were the algorithms used for the training and testing:

1) Linear Regression
2) Random Forest Regression
3) Gradient Boosting regression
4) XGBoost Regression

• Run and Evaluate selected models

- **Linear Regression:**

Linear Regression is used between the target variable and other independent variables to find a linear relationship between them. It is a regression model, used for predicting continuous values. Given, a dataset $\{y , xi1, \ldots , xip\}$ i=1 n of n observations presumes that p-vector of x regression and y, the dependent variable, has a linear relationship. There is a term for disturbance referred to as ε, the variable for error that adds unwanted noise in the relationship between the regressors and the dependent variables.



```
Lr=LinearRegression()
Lr.fit(x_train,y_train)
lr_pred=Lr.predict(x_test)

#r2 score
lr_r2=r2_score(y_test,lr_pred)*100
print("r2 score=",lr_r2)

#cross validation
cv_lr=cross_val_score(Lr,x,y,cv=7).mean()*100
print("cross validation score is",cv_lr)

#evaluation metrics
lr_mae=mean_absolute_error(y_test,lr_pred)
print("mean absolute error=",lr_mae)
lr_mse=mean_squared_error(y_test,lr_pred)
print("mean squared error=",lr_mse)
lr_rmse=np.sqrt(mean_squared_error(y_test,lr_pred))
print("root mean squared error=",lr_rmse)
```

```
r2 score= 49.3585830972395
cross validation score is 41.919709942782724
mean absolute error= 233724.05810395553
mean squared error= 272708564948.65506
root mean squared error= 522215.0562255507
```

Random Forest Regression:

Random Forest is a supervised learning algorithm. The decision tree is contemplated as a base of the Random Forest algorithm .Random Forest can be used for regression as well as classification just like the Decision Tree algorithm .For the classification model it works with a categorical target, whereas in the regression model, it predicts the values of a continuous variable. Random Forest is a collection of several decision trees, just like a real forest that consists of trees. It is often also referred to an ensemble learning technique. For the processing, a set of samples of data are picked arbitrary which comprises different decision trees. Each tree gives the prediction, and the average of them is selected for the case in the regression model. In this project, the regression algorithm of random forests is used. First of all, it divides the dataset into multiple sets, each set produces a decision tree based on different evaluation metrics used. Later the mean of all the predictions is considered as the final prediction.

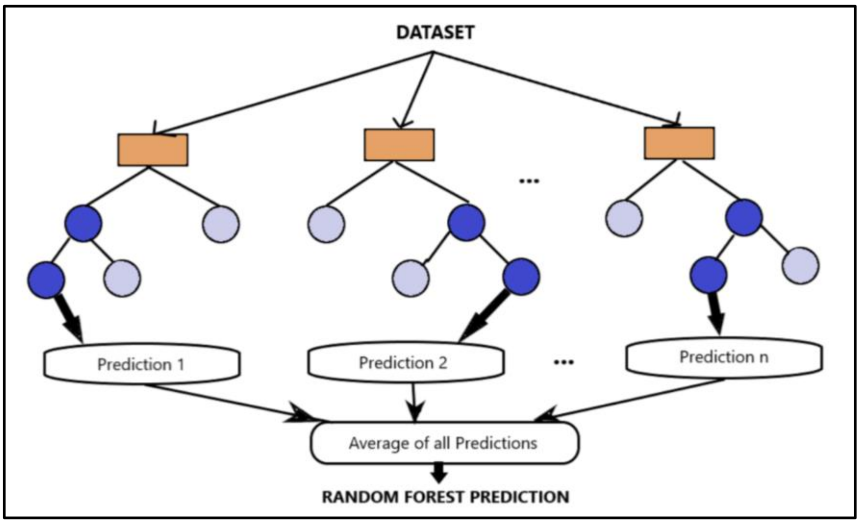
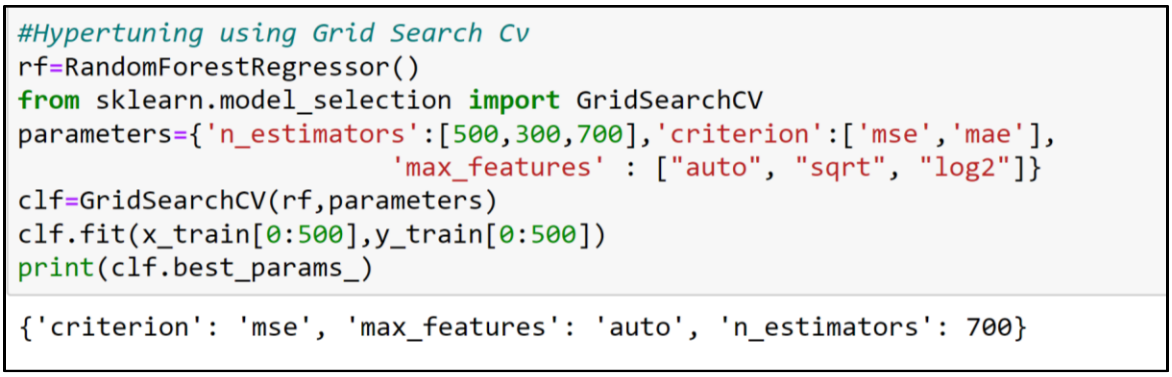
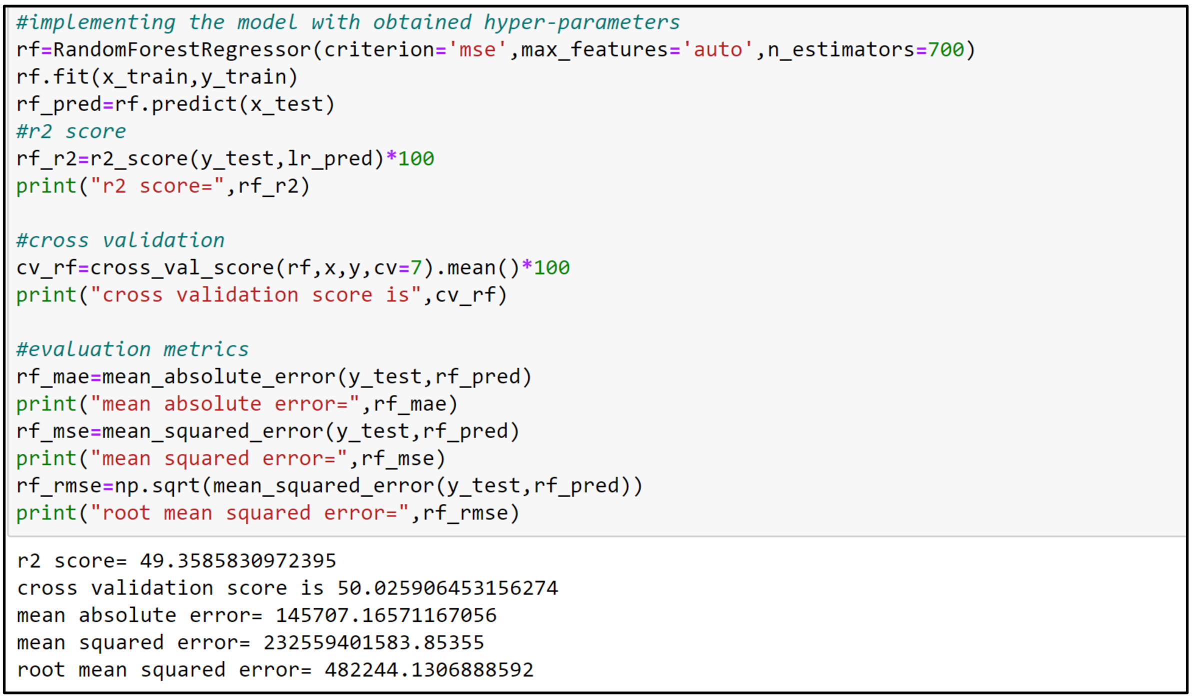


Fig: Overview of Random Forest Regression



```
#Hypertuning using Grid Search Cv
rf=RandomForestRegressor()
from sklearn.model_selection import GridSearchCV
parameters={'n_estimators':[500,300,700],'criterion':['mse','mae'],
                        'max_features' : ["auto", "sqrt", "log2"]}
clf=GridSearchCV(rf,parameters)
clf.fit(x_train[0:500],y_train[0:500])
print(clf.best_params_)
```

```
{'criterion': 'mse', 'max_features': 'auto', 'n_estimators': 700}
```

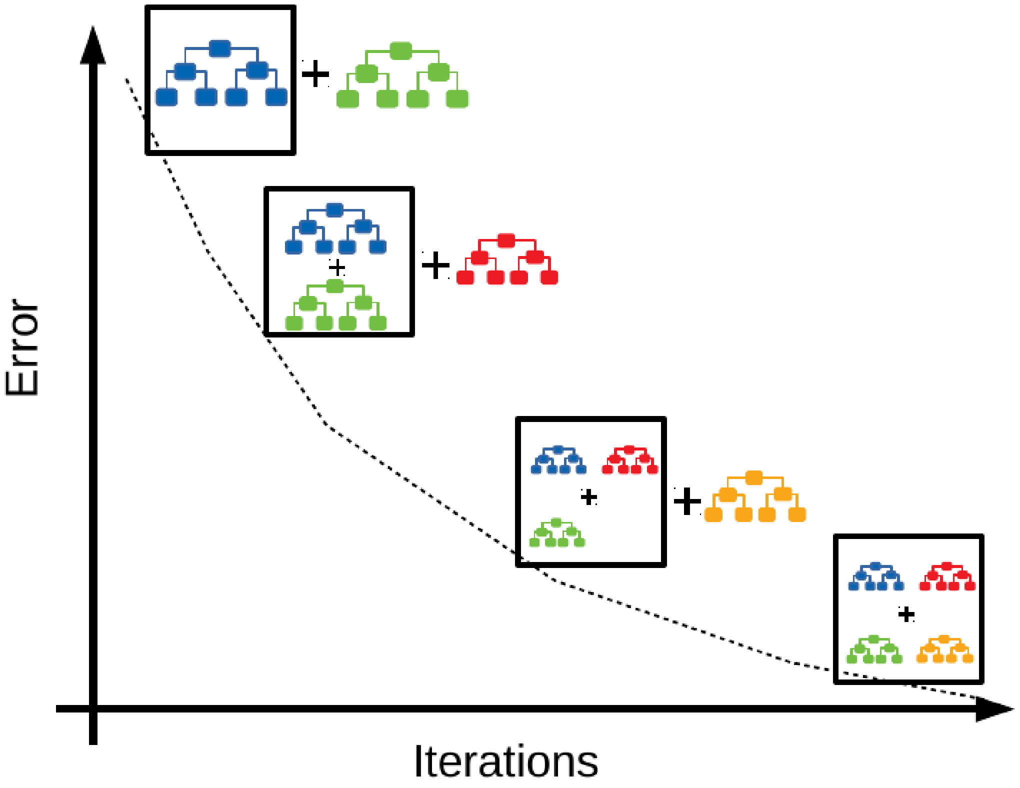


```
#implementing the model with obtained hyper-parameters
rf=RandomForestRegressor(criterion='mse',max_features='auto',n_estimators=700)
rf.fit(x_train,y_train)
rf_pred=rf.predict(x_test)
#r2 score
rf_r2=r2_score(y_test,lr_pred)*100
print("r2 score=",rf_r2)

#cross validation
cv_rf=cross_val_score(rf,x,y,cv=7).mean()*100
print("cross validation score is",cv_rf)

#evaluation metrics
rf_mae=mean_absolute_error(y_test,rf_pred)
print("mean absolute error=",rf_mae)
rf_mse=mean_squared_error(y_test,rf_pred)
print("mean squared error=",rf_mse)
rf_rmse=np.sqrt(mean_squared_error(y_test,rf_pred))
print("root mean squared error=",rf_rmse)
```

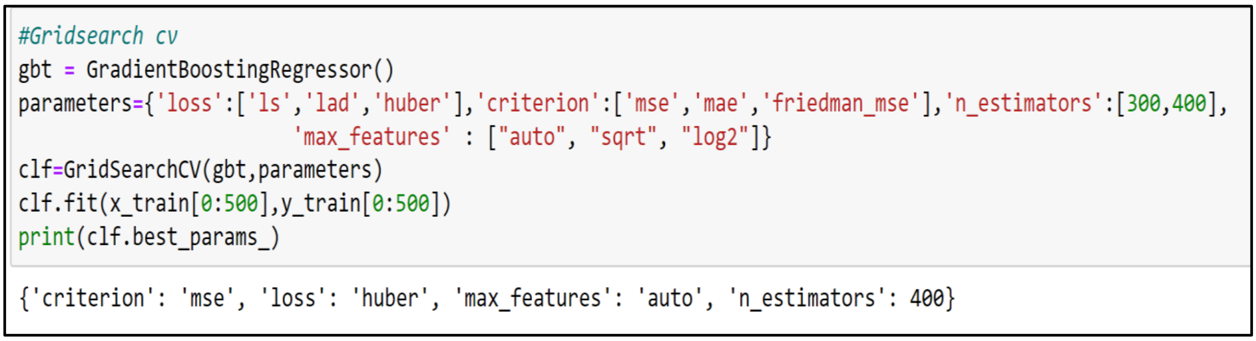
```
r2 score= 49.3585830972395
cross validation score is 50.025906453156274
mean absolute error= 145707.16571167056
mean squared error= 232559401583.85355
root mean squared error= 482244.1306888592
```

Gradient Boosting Regressor

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function.

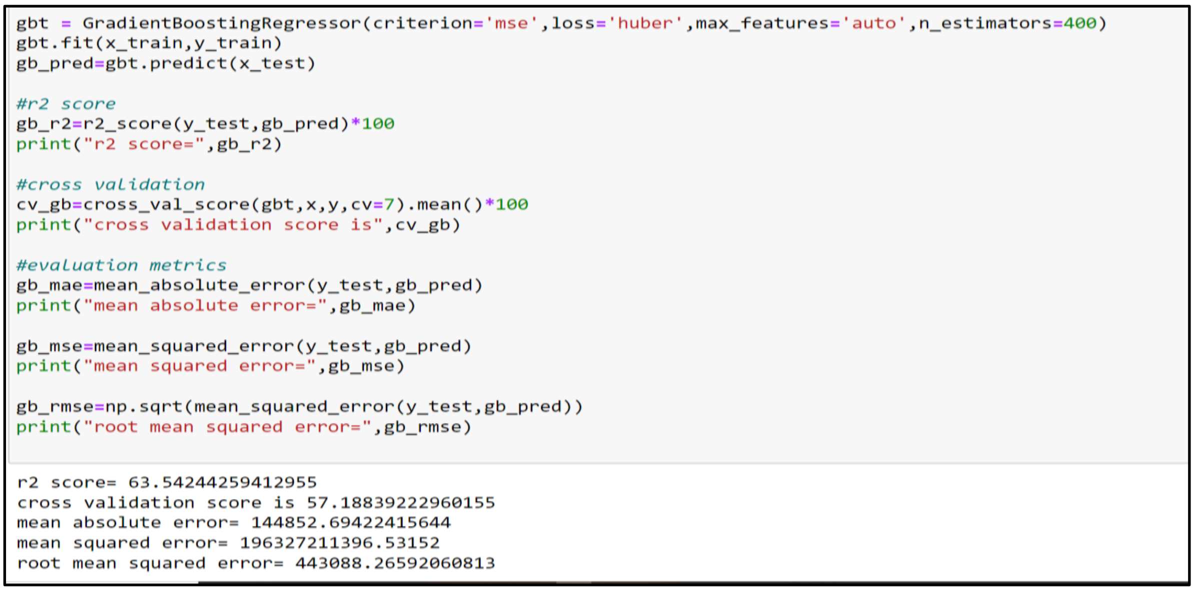


Gradient Boosting produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient boosted trees, which usually outperforms random forest. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.



```
#Gridsearch cv
gbt = GradientBoostingRegressor()
parameters={'loss':['ls','lad','huber'],'criterion':['mse','mae','friedman_mse'],'n_estimators':[300,400],
                         'max_features' : ["auto", "sqrt", "log2"]}
clf=GridSearchCV(gbt,parameters)
clf.fit(x_train[0:500],y_train[0:500])
print(clf.best_params_)
```

```
{'criterion': 'mse', 'loss': 'huber', 'max_features': 'auto', 'n_estimators': 400}
```

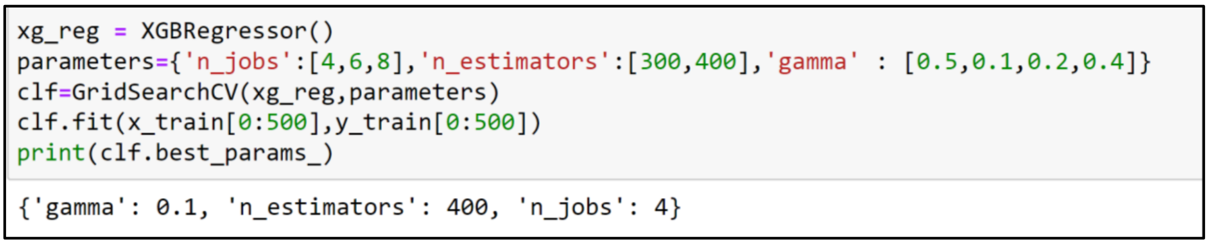


```
gbt = GradientBoostingRegressor(criterion='mse',loss='huber',max_features='auto',n_estimators=400)
gbt.fit(x_train,y_train)
gb_pred=gbt.predict(x_test)

#r2 score
gb_r2=r2_score(y_test,gb_pred)*100
print("r2 score=",gb_r2)

#cross validation
cv_gb=cross_val_score(gbt,x,y,cv=7).mean()*100
print("cross validation score is",cv_gb)

#evaluation metrics
gb_mae=mean_absolute_error(y_test,gb_pred)
print("mean absolute error=",gb_mae)

gb_mse=mean_squared_error(y_test,gb_pred)
print("mean squared error=",gb_mse)

gb_rmse=np.sqrt(mean_squared_error(y_test,gb_pred))
print("root mean squared error=",gb_rmse)
```

```
r2 score= 63.54244259412955
cross validation score is 57.18839222960155
mean absolute error= 144852.69422415644
mean squared error= 196327211396.53152
root mean squared error= 443088.26592060813
```

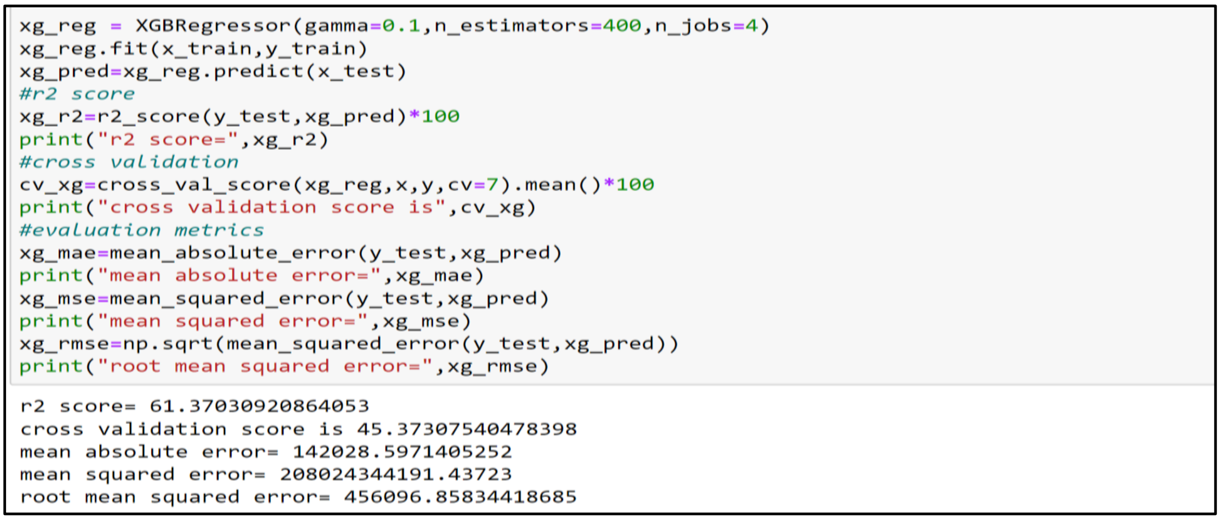
XGBoost Regressor

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm. XGBoost is an efficient implementation of gradient boosting that can be used for regression predictive modelling.



```
xg_reg = XGBRegressor()
parameters={'n_jobs':[4,6,8],'n_estimators':[300,400],'gamma' : [0.5,0.1,0.2,0.4]}
clf=GridSearchCV(xg_reg,parameters)
clf.fit(x_train[0:500],y_train[0:500])
print(clf.best_params_)
```

```
{'gamma': 0.1, 'n_estimators': 400, 'n_jobs': 4}
```



```
xg_reg = XGBRegressor(gamma=0.1,n_estimators=400,n_jobs=4)
xg_reg.fit(x_train,y_train)
xg_pred=xg_reg.predict(x_test)
#r2 score
xg_r2=r2_score(y_test,xg_pred)*100
print("r2 score=",xg_r2)
#cross validation
cv_xg=cross_val_score(xg_reg,x,y,cv=7).mean()*100
print("cross validation score is",cv_xg)
#evaluation metrics
xg_mae=mean_absolute_error(y_test,xg_pred)
print("mean absolute error=",xg_mae)
xg_mse=mean_squared_error(y_test,xg_pred)
print("mean squared error=",xg_mse)
xg_rmse=np.sqrt(mean_squared_error(y_test,xg_pred))
print("root mean squared error=",xg_rmse)
```

```
r2 score= 61.37030920864053
cross validation score is 45.37307540478398
mean absolute error= 142028.5971405252
mean squared error= 208024344191.43723
root mean squared error= 456096.85834418685
```

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

  The objective of Regression is to find a line that minimizes the prediction error of all the data points. The essential step in any machine learning model is to evaluate the accuracy of the model. The Mean Squared Error, Mean absolute error, Root Mean Squared Error, and R-Squared or Coefficient of determination metrics are used to evaluate the performance of the model in regression analysis.

- The Mean absolute error represents the average of the absolute difference between the actual and predicted values in the dataset. It measures the average of the residuals in the dataset.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}|$$

Where,

$$\hat{y} \ - predicted\ value\ of\ y$$
$$\bar{y} - mean\ value\ of\ y$$

- Mean Squared Error represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2$$

- Root Mean Squared Error is the square root of Mean Squared error. It measures the standard deviation of residuals.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

- The coefficient of determination or R-squared represents the proportion of the variance in the dependent variable which is explained by the linear regression model. It is a scale-free score i.e. irrespective of the values being small or large, the value of R square will be less than one.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

- Differences among these evaluation metrics

  Mean Squared Error (MSE) and Root Mean Square Error penalizes the large prediction errors vi-a-vis Mean Absolute Error (MAE). However, RMSE is widely used than MSE to evaluate the performance of the regression model with other random models as it has the same units as the dependent variable (Y-axis).

  MSE is a differentiable function that makes it easy to perform mathematical operations in comparison to a non-differentiable function like MAE. Therefore, in many models, RMSE is used as a default metric for calculating Loss Function despite being harder to interpret than MAE.

  MAE is more robust to data with outliers.

  The lower value of MAE, MSE, and RMSE implies higher accuracy of a regression model. However, a higher value of R square is considered desirable.

R Squared is used for explaining how well the independent variable in the linear regression model explains the variability in the dependent variable. R Squared value always increases with the addition of the independent variables which might lead to the addition of the redundant variables in our model.

For comparing the accuracy among different linear regression models, RMSE is a better choice than R Squared.

• Visualizations

As a data scientist, we always question the amount of time we put into data visualization. Don't we? Ideally we should put more emphasis and efforts into ensuring a thorough analysis rather than just making the graphs pretty and investing lot of time in just decorating them .Prettier graphs won't necessarily mean great analysis.

Data visualization plays a very important role of presenting data in a powerful and credible way.

It is quit useful to have a quick overview of different features distribution v/s carprice. Hence we shall have a quick look at few important features through visualizations.

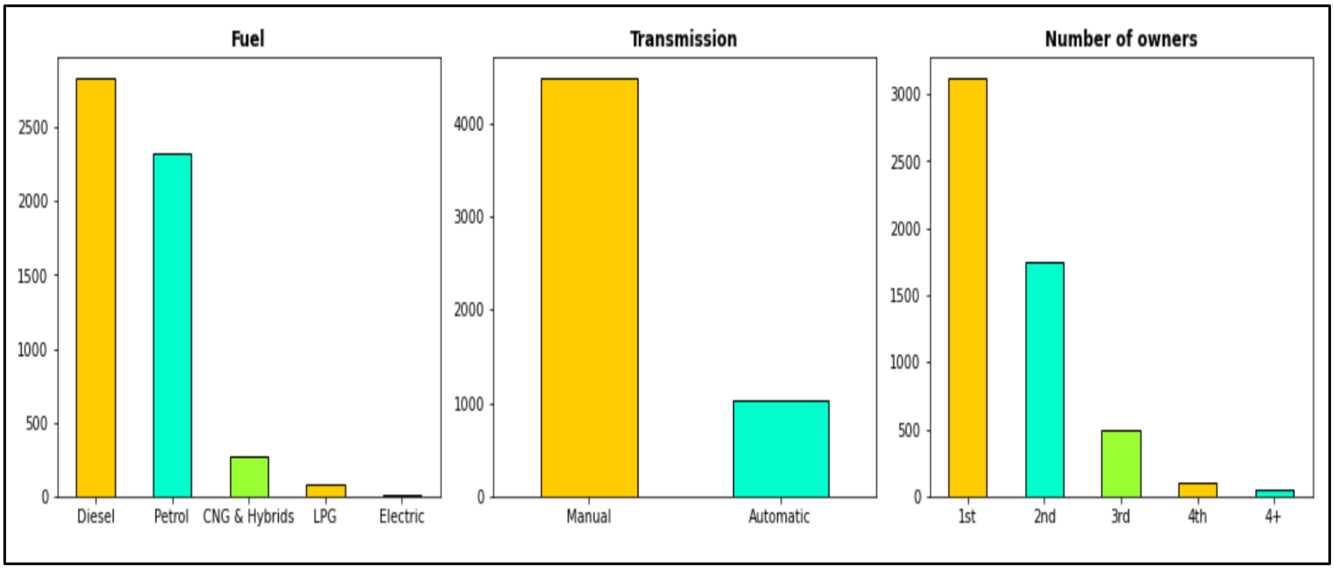


Fig : Count plot of unique values in features

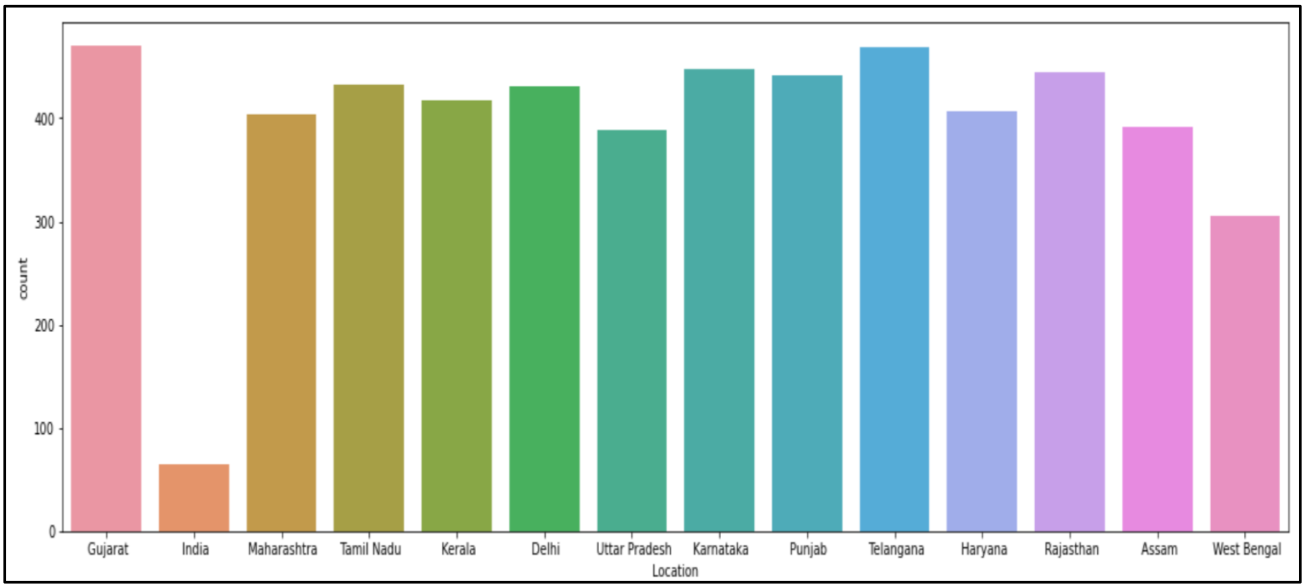


Fig : Count plot of unique values in Location

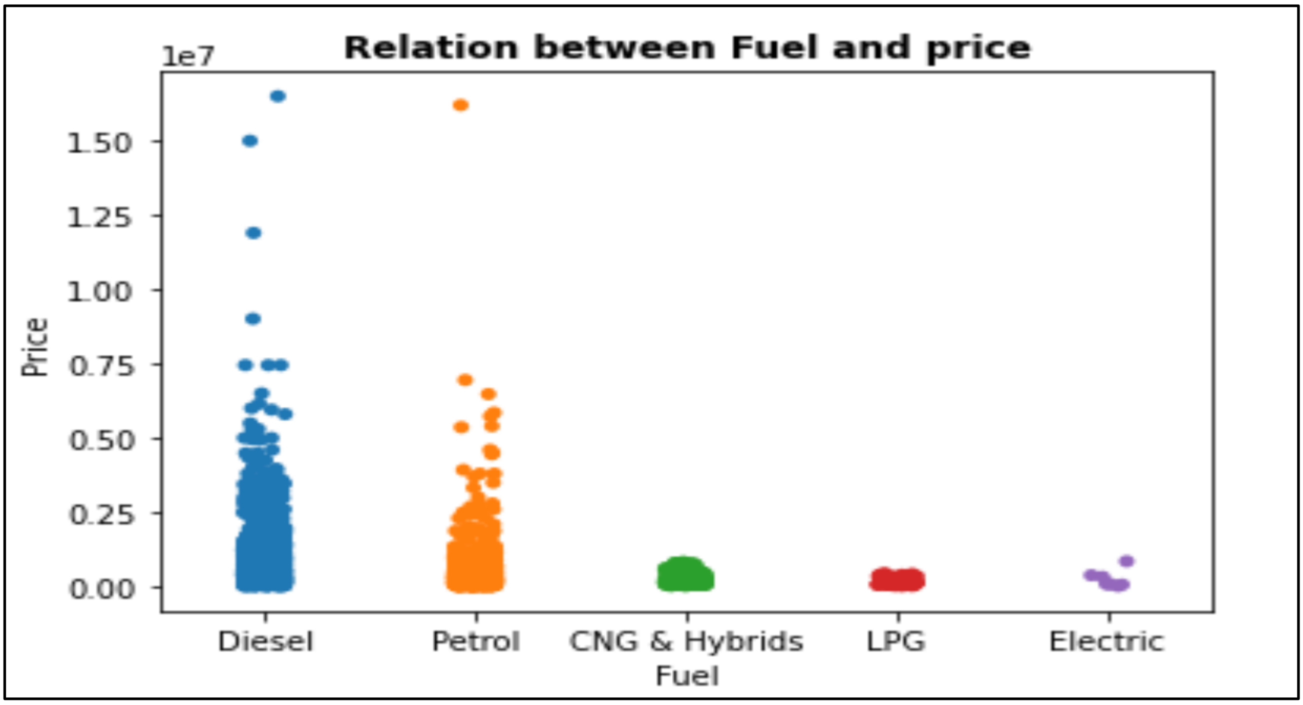


Fig : Relation between Fuel and Car prices

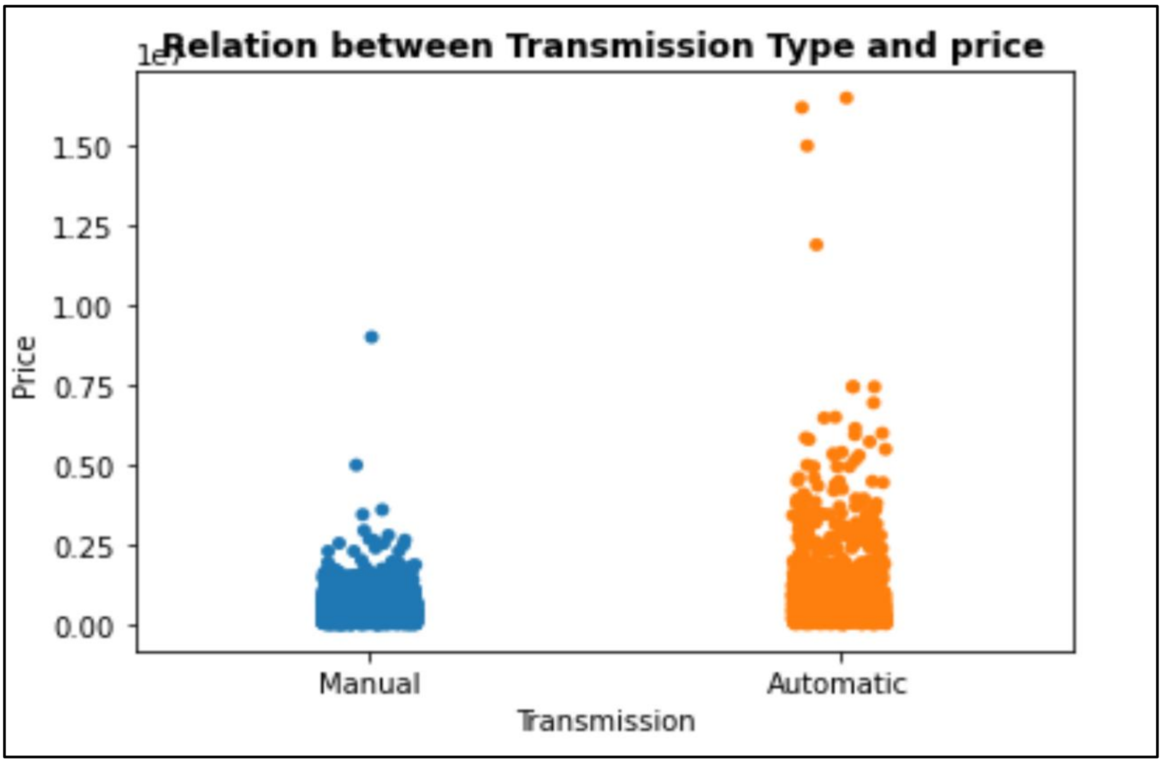


Fig : Relation between Transmission and Car prices

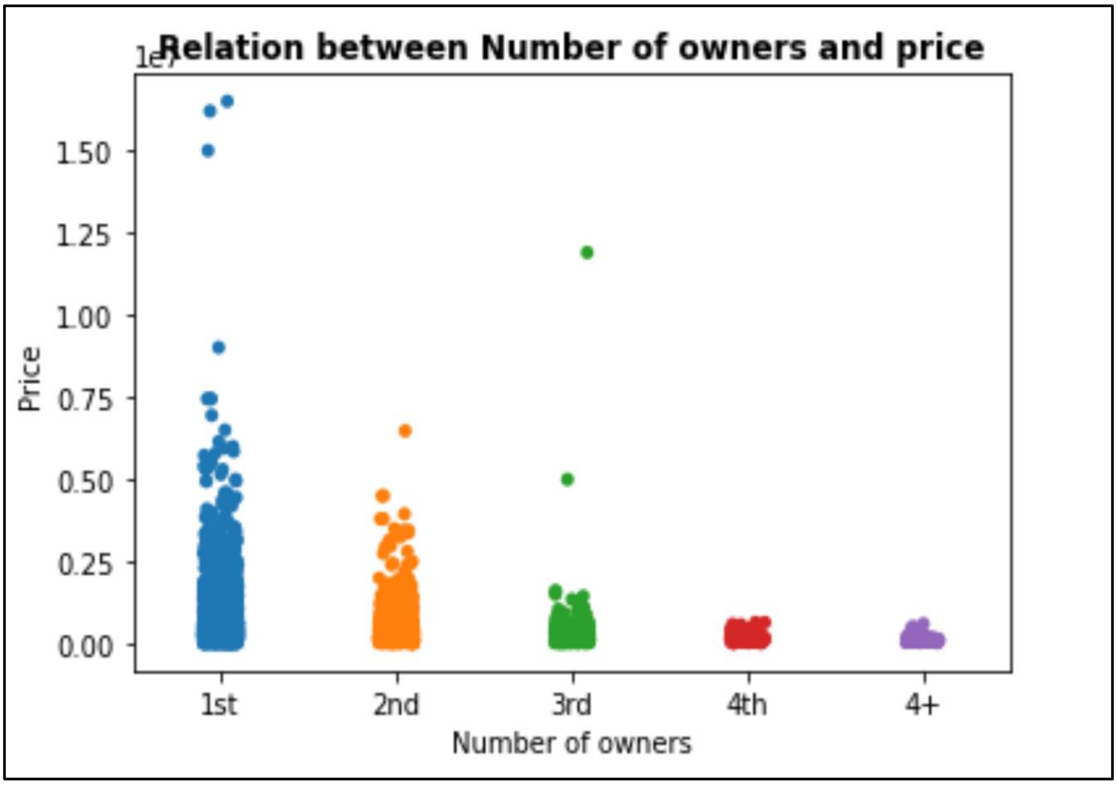


Fig : Relation between Number of owners and Car prices

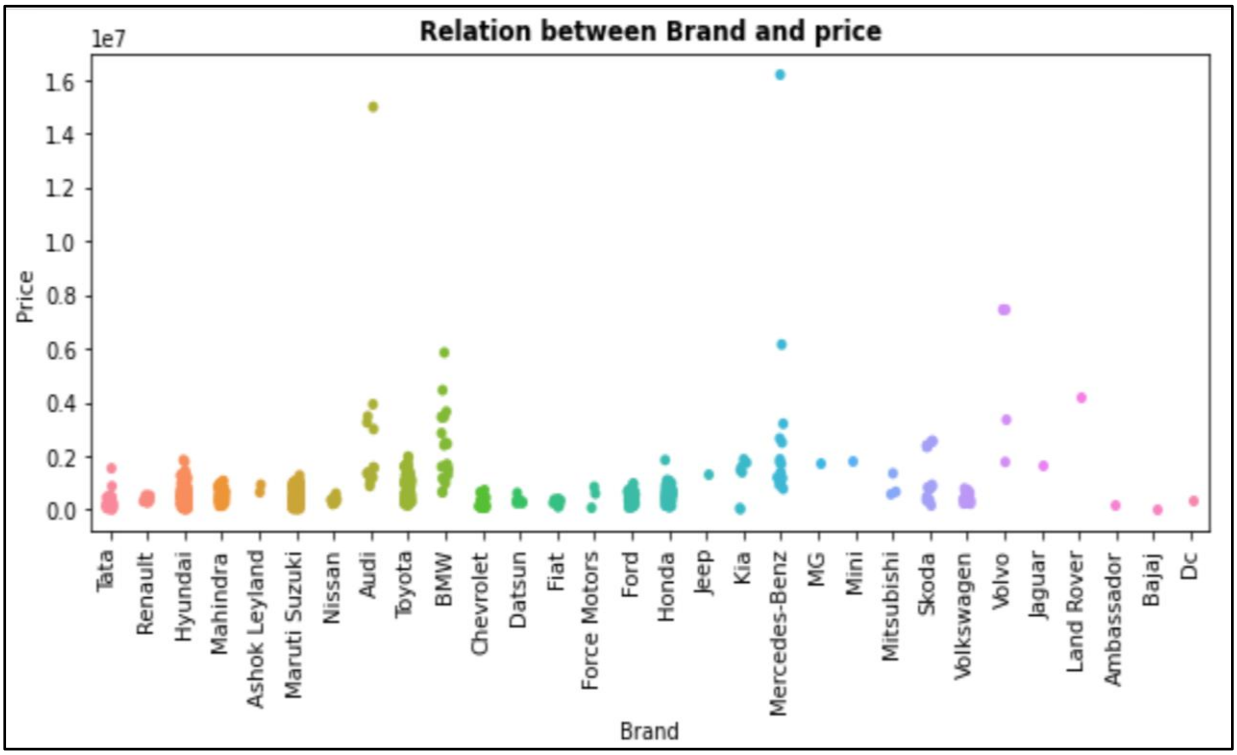


Fig : Relation between Brand and Car prices

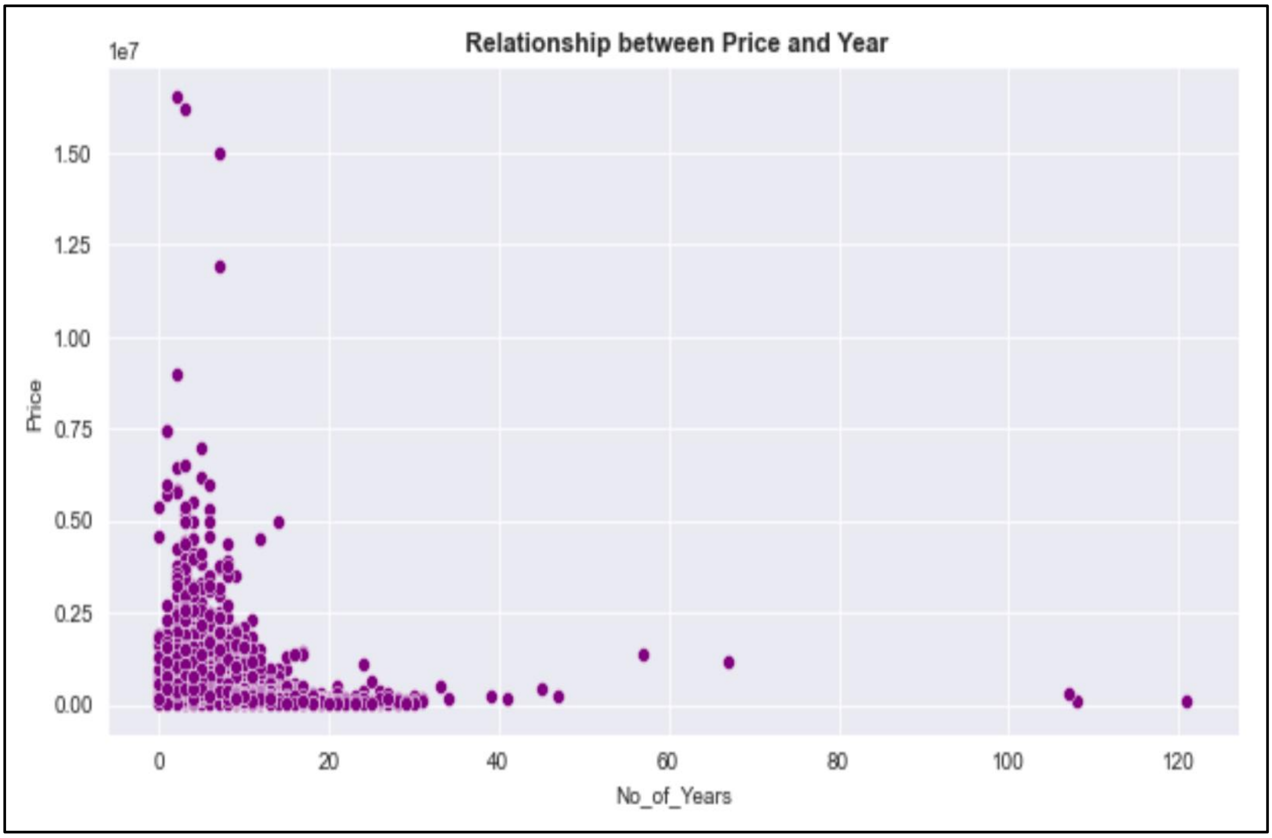


Fig : Relation between Number of years and Car prices

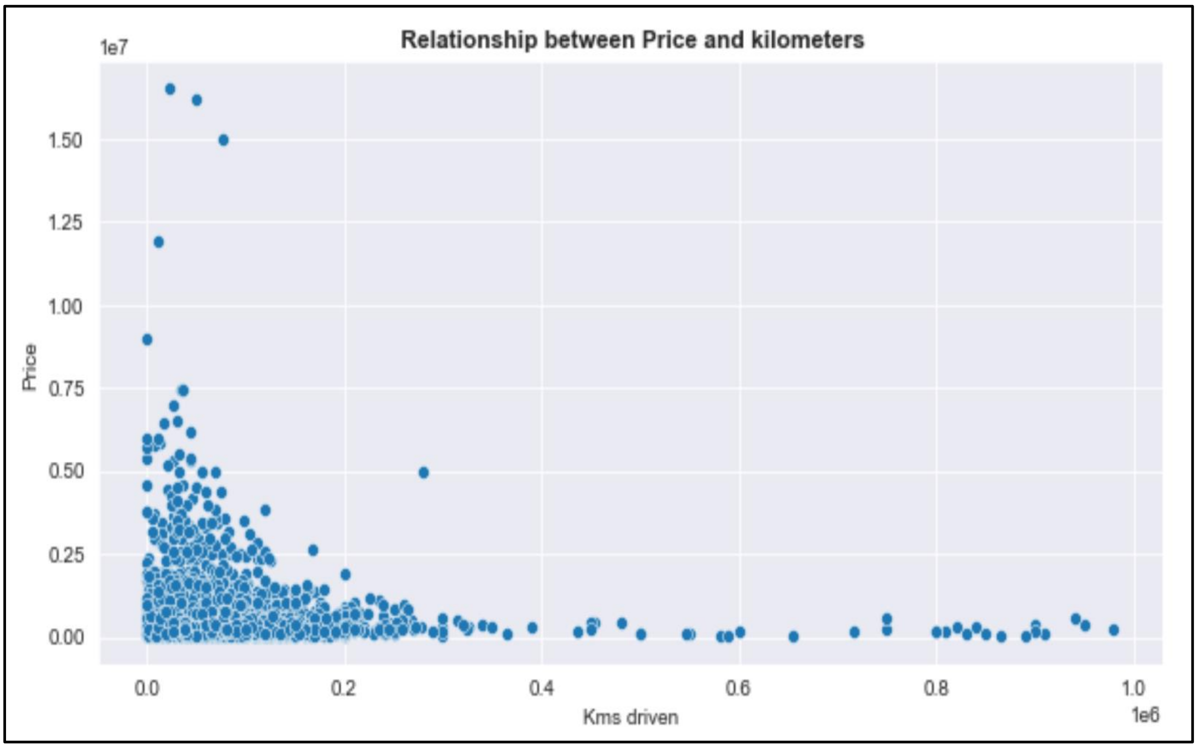


Fig : Relation between Kilometres driven and Car prices

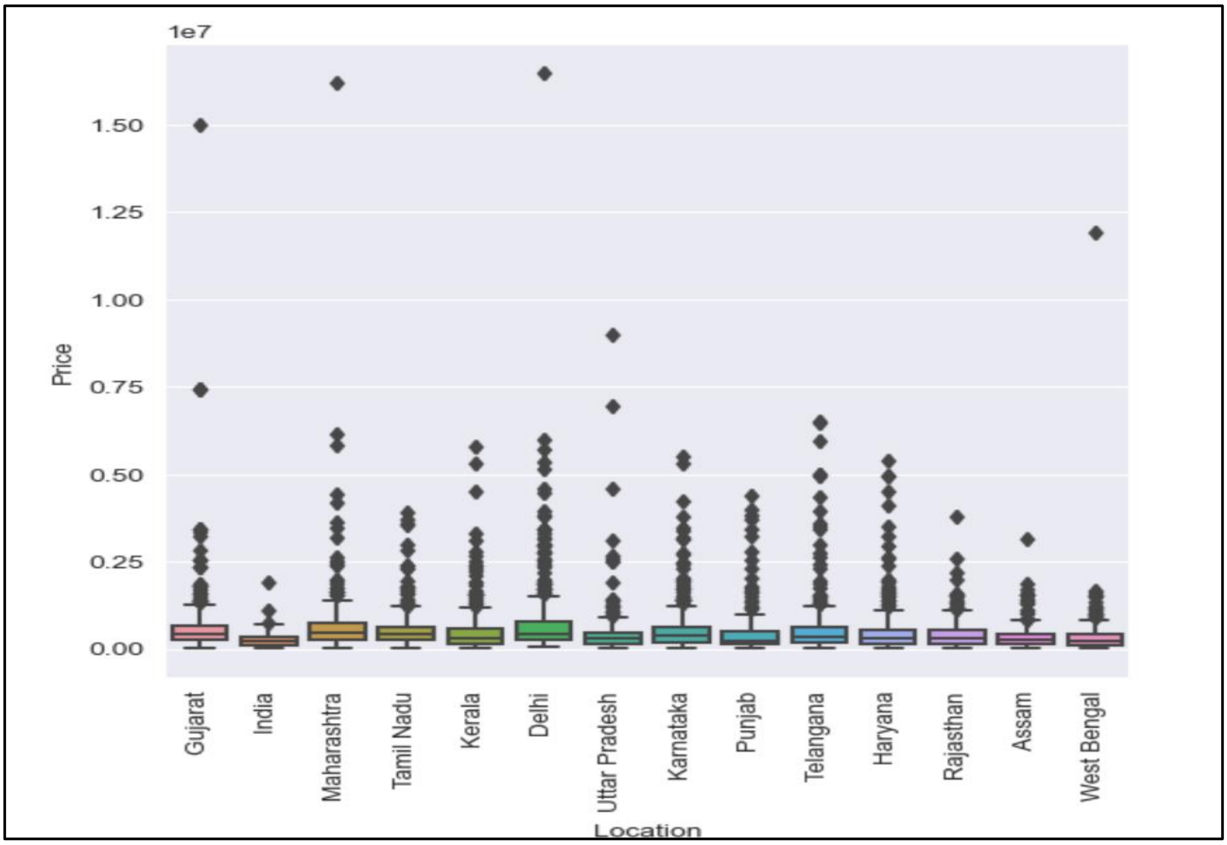


Fig : Box plot for Car prices according to Location

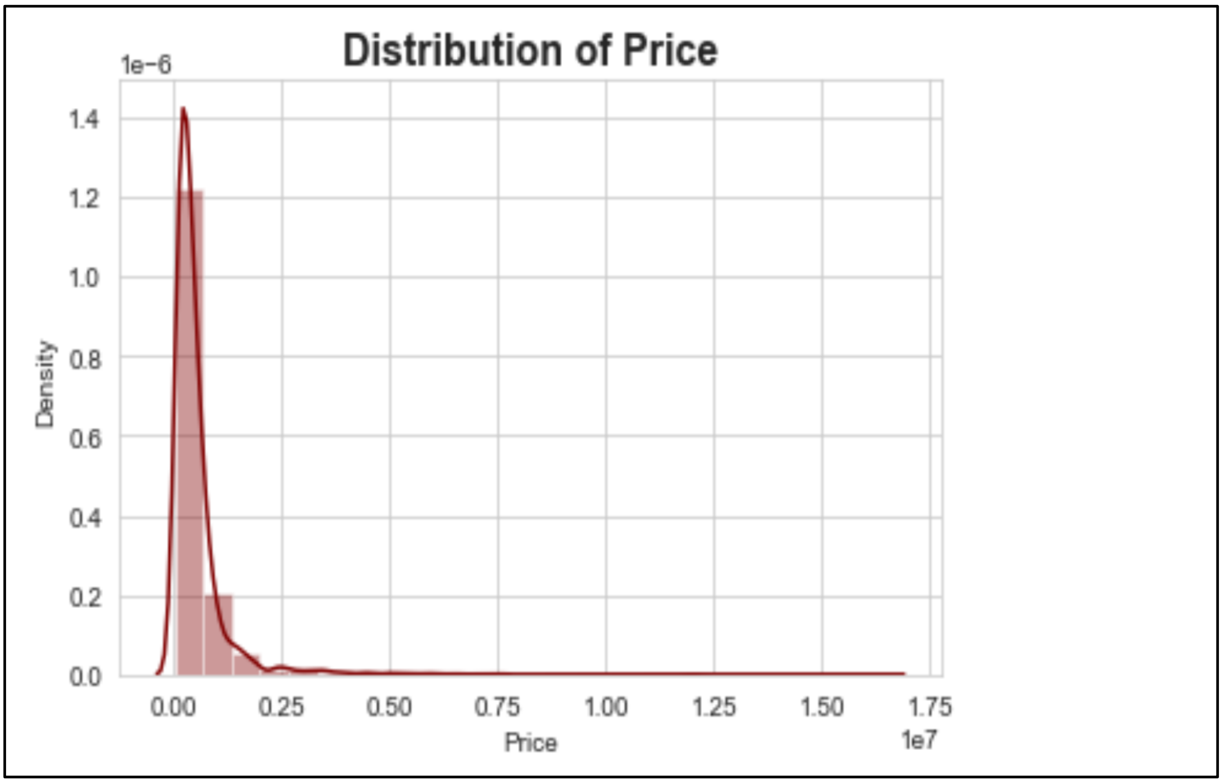


Fig : Distribution plot of car prices

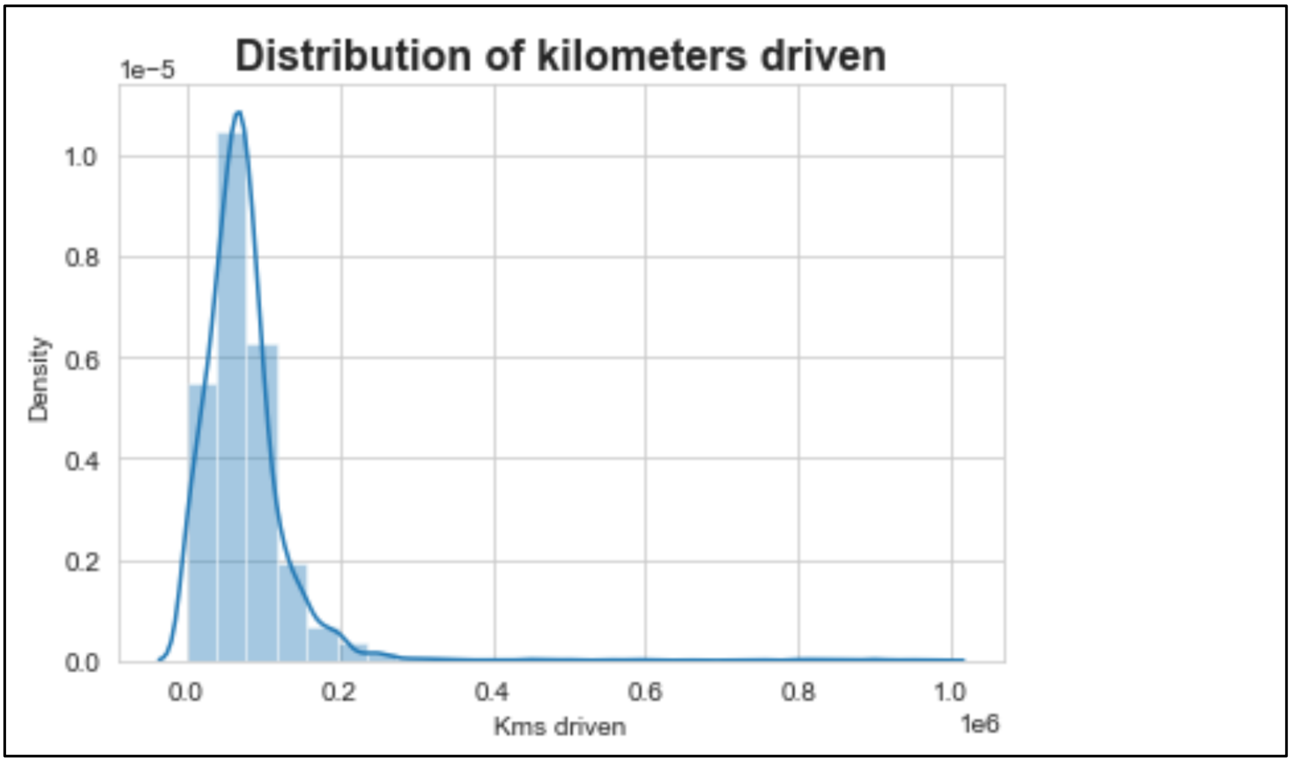


Fig : Distribution plot of Kilometres driven

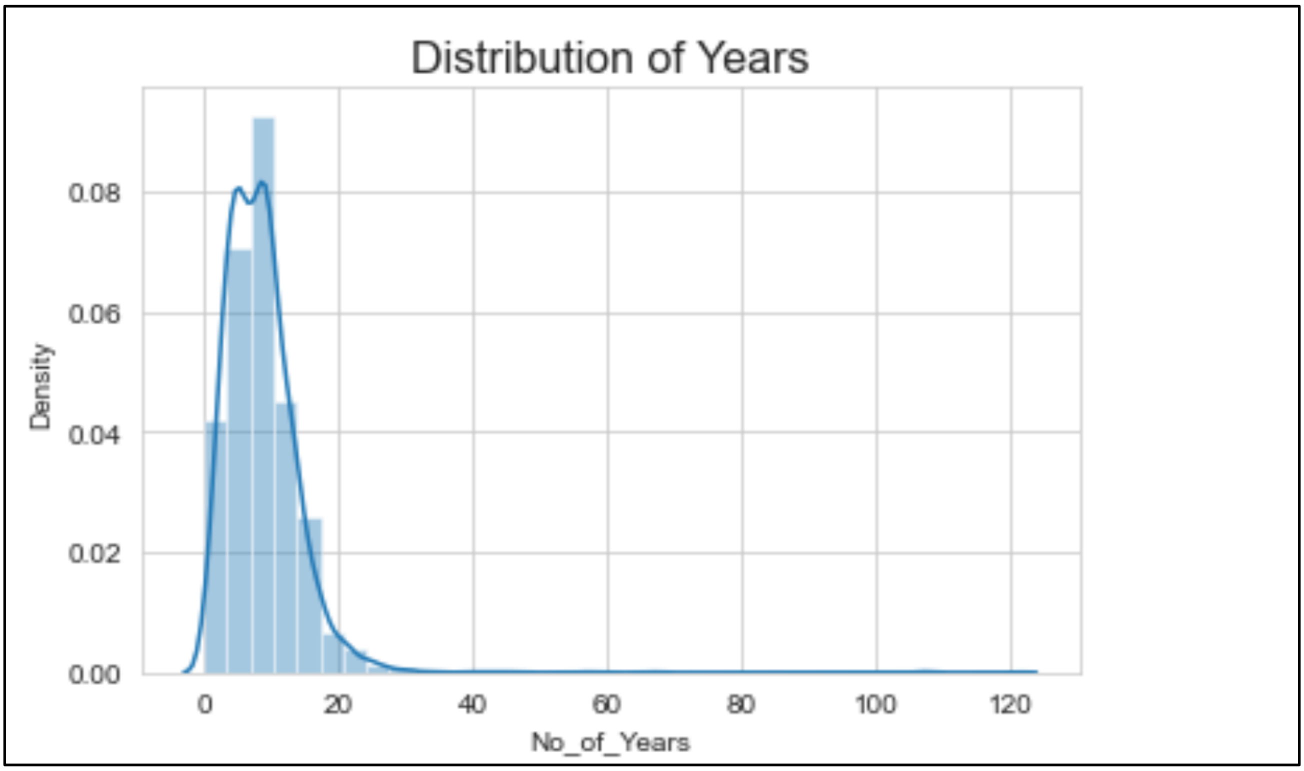


Fig : Distribution plot of Years

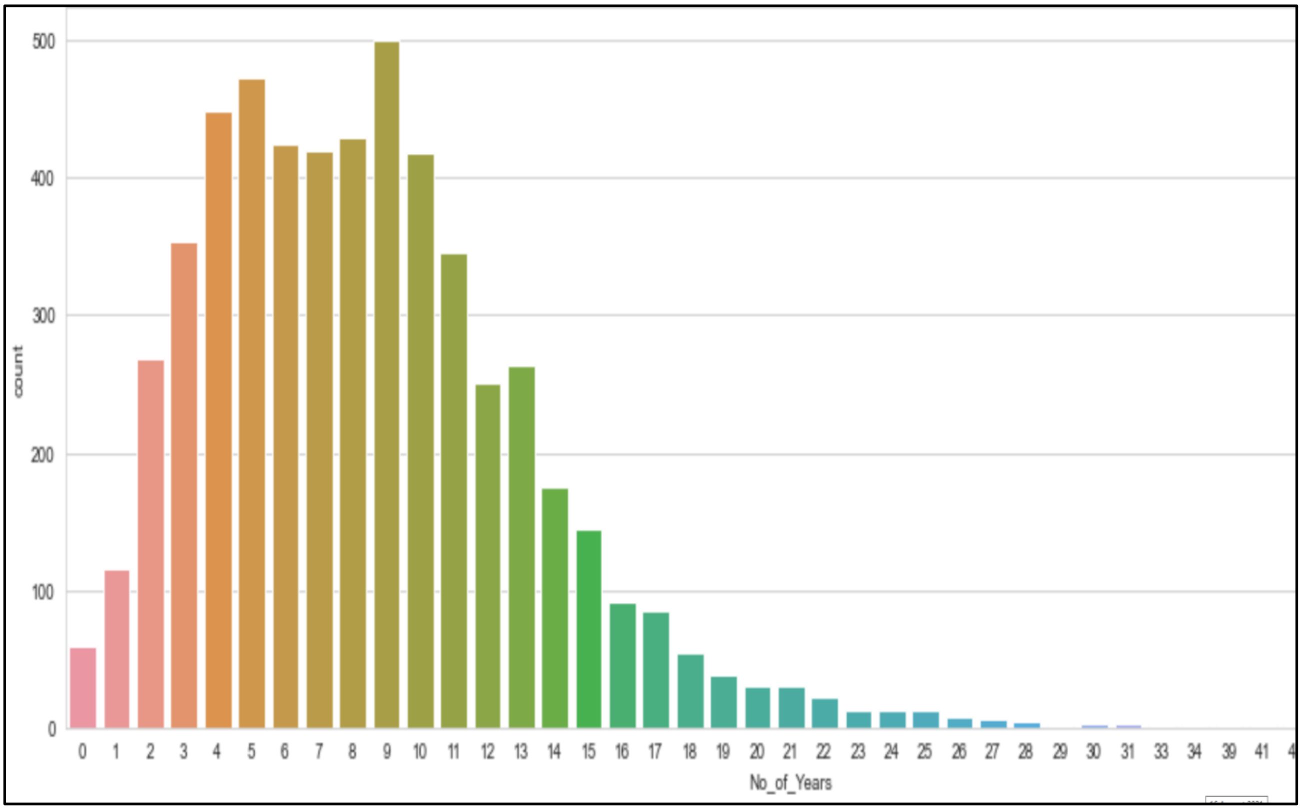


Fig : Count plot of age of the car

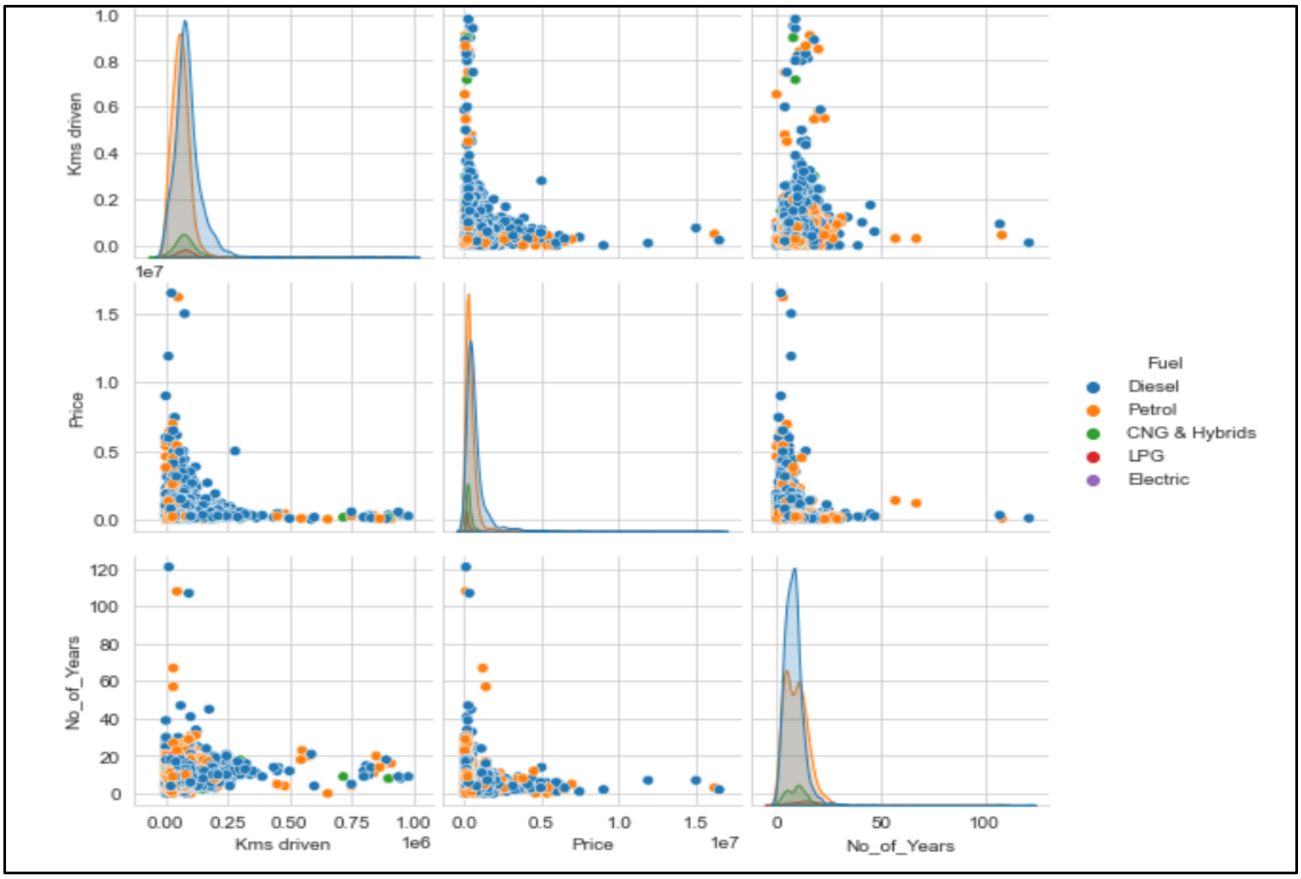


Fig : Pair plot of the variables

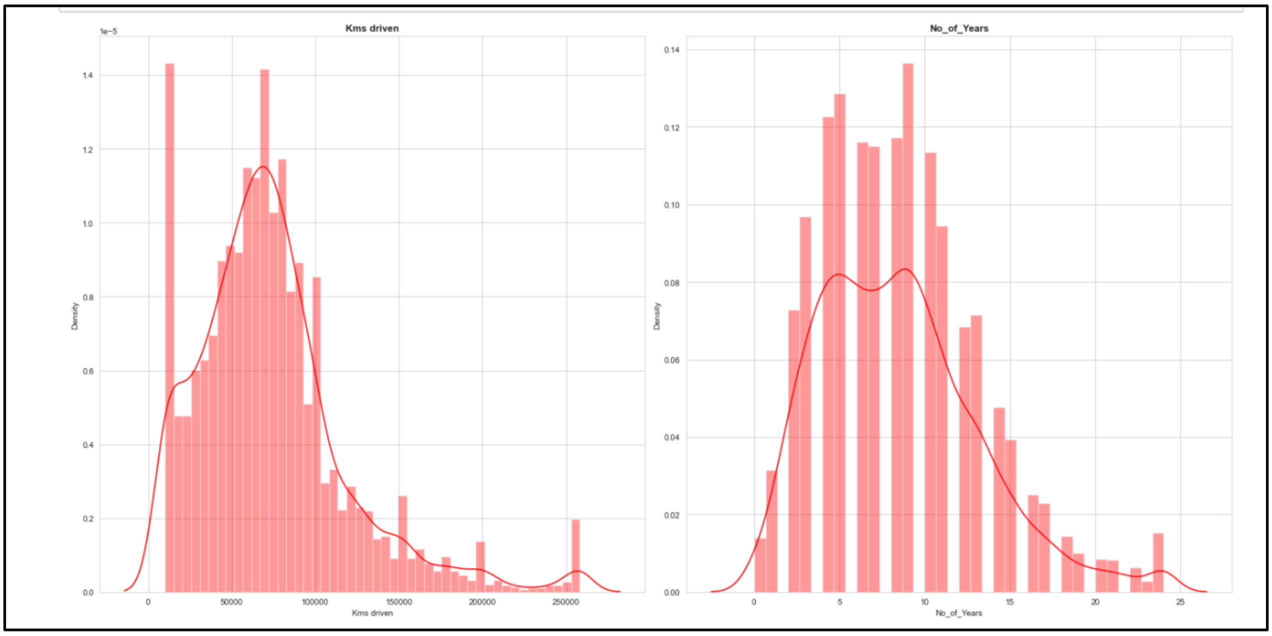


Fig : Distribution plot of Years and kms driven after removal of outliers

- ## Interpretation of the Results

Inferences from visualizations

1) For Transmission type there are 2 unique values 1) Manual and 2) Automatic out of which more than 4000 cars have manual transmission.
2) Most of the cars had diesel and petrol as the fuel type with CNG as third highest and electric as the least.
3) Very few cars were sold 4+ times with 1st owners being the highest.
4) The highest number of cars were 9 years old. The second highest was 5 years and then 4 years
5) There were very few cars which were 0 years old i.e. they were sold within a year of buying a brand new car

Relationship between Target (price) and features:

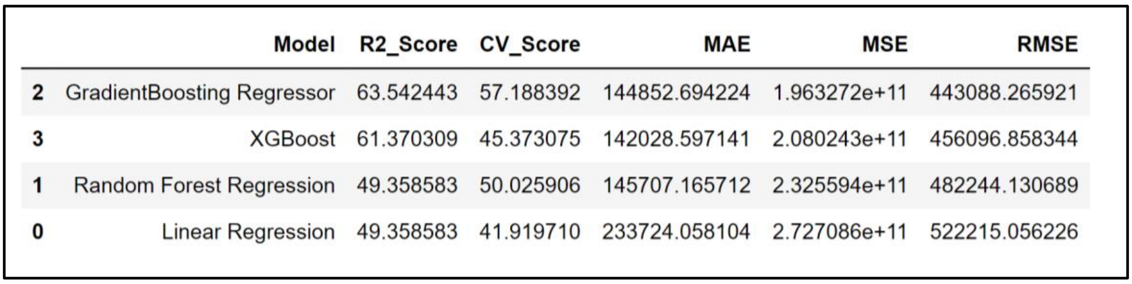
1) Brands like Audi, BMW and Mercedes Benz had the highest selling prices.
2) Bajaj and DC brands were sold at the minimum prices.
3) Cars which were sold more than 4+ times did not earned high selling prices.
4) Cars with Automatic transmission were sold at higher prices as compared to those with manual transmission.
5) Petrol and diesel cars are sold at higher prices whereas Electric cars had the least selling price among all fuel types.
6) The cars aged between 0 to 10 years had the highest prices.
7) The cars above 25 years had really low prices . Few outliers could be seen for car with 0 years old.
8) when the kms driven was less the prices were high. After 0.6(1e6) kms the prices were almost stable at 0.13(1e6)

Distribution plots:

1) Price and kilometres driven shows right skew in their distribution which indicated that the mean is greater than median and mode.

2) Among all the states in India the highest prices were earned in Maharashtra followed by Delhi, Telangana and Karnataka.
3) Assam and West Bengal has low selling prices for used cars.

The model performances can be summarized in the following table:



	Model	R2_Score	CV_Score	MAE	MSE	RMSE
2	GradientBoosting Regressor	63.542443	57.188392	144852.694224	1.963272e+11	443088.265921
3	XGBoost	61.370309	45.373075	142028.597141	2.080243e+11	456096.858344
1	Random Forest Regression	49.358583	50.025906	145707.165712	2.325594e+11	482244.130689
0	Linear Regression	49.358583	41.919710	233724.058104	2.727086e+11	522215.056226

In terms of r2 score Gradient Boosting and XGBoost Regressor are performing well, but the major focus being on RMSE we selected Gradient Boosting as our final model.

The RMSE are high as our target variable do not follow a normal distribution curve and have too many outliers because of the fact that a Maruti Suzuki brand and Audi will have a huge price gap which cannot be ignored.

CONCLUSION

- **Key Findings and Conclusions of the Study**

The major step in the prediction process is collection and pre-processing of the data. By performing different ML models, we aim to get a better result or less error with max accuracy. Our purpose was to predict the price of the used cars having 9 predictors and 6069 data entries.

Initially, data cleaning is performed to remove the null values and outliers from the dataset then ML models are implemented to predict the price of cars.

Next, with the help of data visualization features were explored deeply. The relation between the features is examined.

Considerable number of distinct attributes are examined for the reliable and accurate prediction. To build a model for predicting the price of used cars in India, we applied four machine learning techniques (Linear Regression, Random Forest, Gradient Boosting and XGBoost). The data used for the prediction was collected from the web portal olx.com using web scraper that was written in Python programming language with the help of selenium. Since manual data collection is time consuming task, especially when there are numerous records to process, a “web scraper” is created to get this job done automatically and reduce the time for data gathering. Web scraping is well known technique to extract information from websites and save data into local file or database. Manual data extraction is time consuming and therefore web scrapers are used to do this job in a fraction of time.

Respective performances of different algorithms were then compared to find one that best suits the available data set.

Due to the very high number of combinations of available options within even the same year/make/model car (ex. Brand, transmission, fuel, location, etc.), making fair comparisons becomes very difficult, if not impossible, with a dataset of our size.

The most relevant features used for this prediction are price, kilometre driven, brand, fuel, Model, Number of owners, Transmission and Year . By filtering out outliers and irrelevant features of the dataset a sophisticated model, Gradient Boosting gives good accuracy in comparison to others. Gradient Boost as a regression model gave an accuracy of 63.5 and best MSE and RMSE values.

- **Learning Outcomes of the Study in respect of Data Science**

An accurate used car price evaluation is a catalyst for the healthy development of used car market. Data mining has been applied to predict used car price in several articles. However, little is studied on the comparison of using different algorithms in used car price estimation. This project collects more than 5,000 used car dealing records throughout India to do empirical analysis on a thorough comparison of various machine learning algorithms like linear regression ,random forest, gradient boosting and XGBoost.

This study used different models in order to predict used car prices. However, there was a relatively small dataset for making a strong inference because number of observations was only 6069. Gathering more data can yield more robust predictions. Secondly, there could be more features that can be good predictors. For example, here are some variables that might improve the model: number of doors, gas/mile (per gallon), colour, mechanical and cosmetic reconditioning time, used-to-new ratio, appraisal-to-trade ratio.

- ## Limitations of this work and Scope for Future Work

Although, this system has achieved good performance in car price prediction problem our aim for the future research is to test this system to work successfully with various data sets. We can extend our test data with ebay, cardheko and car24 website for used cars data sets and validate the proposed approach.

Other factors such as the make of car , the interior style, the braking system, acceleration, the volume of its cylinders (measured in cc), safety index, its size, number of doors, paint colour, weight of the car, consumer reviews, prestigious awards won by the car manufacturer, its physical state, whether it is a sports car, whether it has cruise control, whether it belonged to an individual or a company and other options such as air conditioner, sound system, power steering, cosmic wheels, GPS navigator, the look and feel of the car etc. can be studied as they certainly contributes a lot to the price and may influence the price as well .Future work may include studying all these factors and its impact on the car price.

As suggestion for further studies, while pre-processing data, instead of using one hot encoder method Label encoder can be used. This may cause a serious change in performance of predictive models. Also, after training the data, instead of standard scaler ,min-max scaler can be performed and results can be compared. Different scalers can be checked whether there is an improvement in prediction power of models or not.