FLIP ROBO

USED CAR PRICE PREDICTION PROJECT

Submitted by:

Akanksha Padhye

ACKNOWLEDGMENT

I have gave my best in this project. I am highly indebted to Flip Robo Technologies Bangalore for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. I want to thank my SME Ms. Khushboo Garg for providing the Dataset and helping us to solve the problem and addressing out our Query in right time.

INTRODUCTION

Business Problem Framing

Impact of COVID-19 on Indian automotive sector

The Indian automotive sector was already struggling in FY20. before the Covid-19 crisis. It saw an overall degrowth of nearly 18 per cent. This situation was worsened by the onset of the Covid-19 pandemic and the ongoing lockdowns across India and the rest of the world. These two years (FY20 and FY21) are challenging times for the Indian automotive sector on account of slow economic growth, negative consumer sentiment, BS-VI transition, changes to the axle load norms, liquidity crunch, low-capacity utilisation and potential bankruptcies.

The return of daily life and manufacturing activity to near normalcy in China and South Korea, along with extended lockdown in India, gives hope for a U-shaped economic recovery. Our analysis indicates that the Indian automotive sector will start to see recovery in the third quarter of FY21. We expect the industry demand to be down 15-25 per cent in FY21. With such degrowth, OEMs, dealers and suppliers with strong cash reserves and better access to capital will be better positioned to sail through.
Auto sector has been under pressure due to a mix of demand and supply factors. However, there are also some positive outcomes, which we shall look at.

- With India’s GDP growth rate for FY21 being downgraded from 5% to 0% and later to (-5%), the auto sector will take a hit. Auto demand is highly sensitive to job creation and income levels and both have been impacted. CII has estimated the revenue impact at $2 billion on a monthly basis across the auto industry in India.
- Supply chain could be the worst affected. Even as China recovers, supply chain disruptions are likely to last for some more time. The problems on the Indo-China border at Ladakh are not helping matters. Domestic suppliers are chipping in but they will face an inventory surplus as demand remains tepid.
- The Unlock 1.0 will coincide with the implementation of the BS-VI norms and that would mean heavier discounts to dealers and also to customers. Even as auto companies are managing costs, the impact of discounts on profitability is going to be fairly steep.
- The real pain could be on the dealer end with most of them struggling with excess inventory and lack of funding options in the post COVID-19 scenario. The BS-VI price increases are also likely to hit auto demand.

There are two positive developments emanating from COVID-19. The China supply chain shock is forcing major investments in the “Make in India” initiative. The COVID-19 crisis has exposed chinks in the automobile business model and it could catalyse a big move towards electric vehicles (EVs). That could be the big positive for auto sector.

Conceptual Background of the Domain Problem

Understanding the above business problem, there are certain factors that will influence the automotive industries in the future. Some of them include digital technologies, changing customer preferences, electrical vehicles, intelligent ability, and technical advancements. Technologies such as artificial intelligence, machine learning, cloud computing, and internet of things will also play an important role in developing new business models. Apart from that, they enable customers to ensure a better mobility experience. In other words, technologies may impact automotive industry units significantly that will change the markets. The introduction of electrical cars and hybrid vehicles may transform the automobile industries in coming years.

Review of Literature.

As per the requirement of our client, I have scrubbed data from different used cars selling merchants websites, and so based on the data collected I have tried analysing based on what factors the used car price is decided? What is the relationship between cost of the used cars and other factors like Fuel type, Brand and Model, year the car is purchased and No. Of owners before selling? And so based on all the above consideration I have developed a model that will predict the price of the used cars.

Motivation for the Problem Undertaken

I have taken this problem based on the requirement of the client and also, with a curiosity to know how the used cars markets are at the time of pandemic.

ANALYTICAL PROBLEM FRAMING

Mathematical/ Analytical Modelling of the Problem

We are building a model in Machine Learning to predict the actual value price of the prospective cars and decide whether to buy them or not. So, this model will help us to determine which variables are important to predict the price of variables & also how do these variables describe the price of the car. This will help to determine the price of cars with the available independent variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns.

- Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features'). The most common form of regression analysis is linear regression, in which one finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion. For specific mathematical reasons this allows the researcher to estimate the conditional expectation of the dependent variable when the independent variables take on a given set of values.
- Regression analysis is also a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.

  The different Mathematical/Analytical models that are used in this project are as below:

  1.Linear regression - is a linear model, e.g., a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

2. Lasso - In statistics and machine learning, lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

3. Ridge - regression is a way to create a parsimonious model when the number of predictor variables in a set exceeds the number of observations, or when a data set has multi co linearity (correlations between predictor variables).

4. Elastic Net - is a popular type of regularized linear regression that combines two popular penalties, specifically the L1 and L2 penalty functions. Elastic net linear regression uses the penalties from both the lasso and ridge techniques to regularize regression models. The technique combines both the lasso and ridge regression methods by learning from their shortcomings to improve on the regularization of statistical models.

5. K Neighbors Regressor - KNN algorithm can be used for both classification and regression problems. The KNN algorithm uses 'feature similarity' to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.

6. Decision Tree - is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application. It is a tree-structured classifier with three types of nodes.

7. Random forest - is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and

control over-fitting. A Random Forest's nonlinear nature can give it a leg up over linear algorithms, making it a great option.

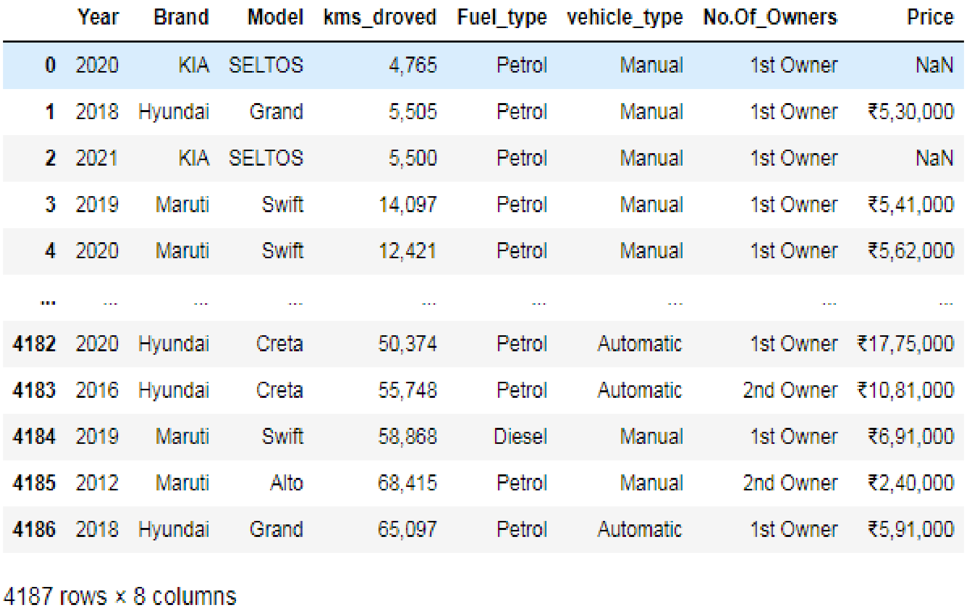
8. AdaBoost Regressor - is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction.

9. 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.

- First, use the train dataset and do the EDA process, fitting the best model and saving the model.
- Then, use the test dataset, load the saved model and predict the values over the test data.

Data Sources and their formats

- Dataset has 4187 rows and 8 columns.
- There are null values present in the dataset.
- Dataset contains categorial and continuous data.



	Year	Brand	Model	kms_droved	Fuel_type	vehicle_type	No.Of_Owners	Price
0	2020	KIA	SELTOS	4,765	Petrol	Manual	1st Owner	NaN
1	2018	Hyundai	Grand	5,505	Petrol	Manual	1st Owner	₹5,30,000
2	2021	KIA	SELTOS	5,500	Petrol	Manual	1st Owner	NaN
3	2019	Maruti	Swift	14,097	Petrol	Manual	1st Owner	₹5,41,000
4	2020	Maruti	Swift	12,421	Petrol	Manual	1st Owner	₹5,62,000
...	...	...	...	...	...	...	...	...
4182	2020	Hyundai	Creta	50,374	Petrol	Automatic	1st Owner	₹17,75,000
4183	2016	Hyundai	Creta	55,748	Petrol	Automatic	2nd Owner	₹10,81,000
4184	2019	Maruti	Swift	58,868	Diesel	Manual	1st Owner	₹6,91,000
4185	2012	Maruti	Alto	68,415	Petrol	Manual	2nd Owner	₹2,40,000
4186	2018	Hyundai	Grand	65,097	Petrol	Automatic	1st Owner	₹5,91,000

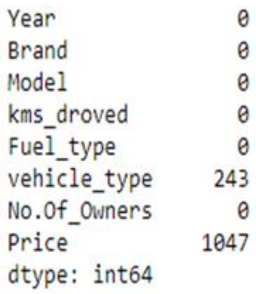
4187 rows × 8 columns

Data description

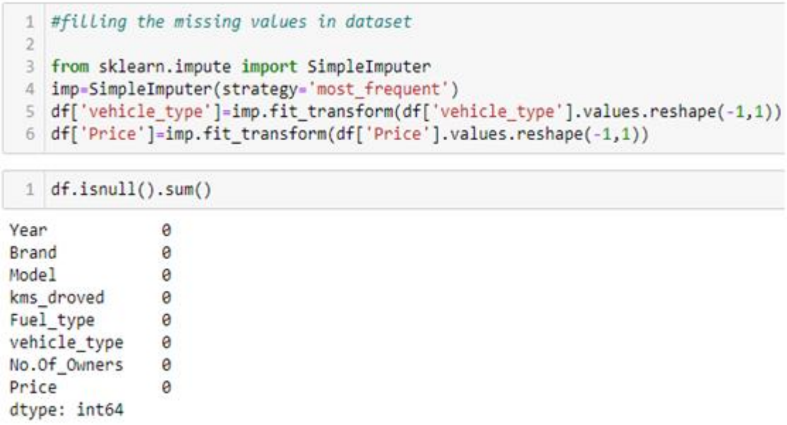
Data contains 4187 entries each having 8 variables. The details of the features are given below:

1. 'YEAR' – At what year the car is manufactured
2. 'BRAND' – Brand is manufacturer or which company made
3. 'MODEL' – It is basically the model of the car.
4. 'VEHICLE_TYPE' – Gear shift variant is (Automatic, Manual, Semi-Automatic)
5. 'DRIVEN_KM' – no of Kms driven before selling
6. 'FUELTYPE' – Petrol, diesel, CNG, LPG, Electric
7. 'NOOF_OWNERS' – 1nd, 2nd or 3nd
8. 'PRICE' – our target variable that tells what is the price of the used car.

CHECKING THE NUMBER OF NULL VALUES & HANDELING THEM:



```
Year               0
Brand              0
Model              0
kms_droved         0
Fuel_type          0
vehicle_type     243
No.Of_Owners       0
Price           1047
dtype: int64
```



```python
#filling the missing values in dataset

from sklearn.impute import SimpleImputer
imp=SimpleImputer(strategy='most_frequent')
df['vehicle_type']=imp.fit_transform(df['vehicle_type'].values.reshape(-1,1))
df['Price']=imp.fit_transform(df['Price'].values.reshape(-1,1))
```

```python
df.isnull().sum()
```

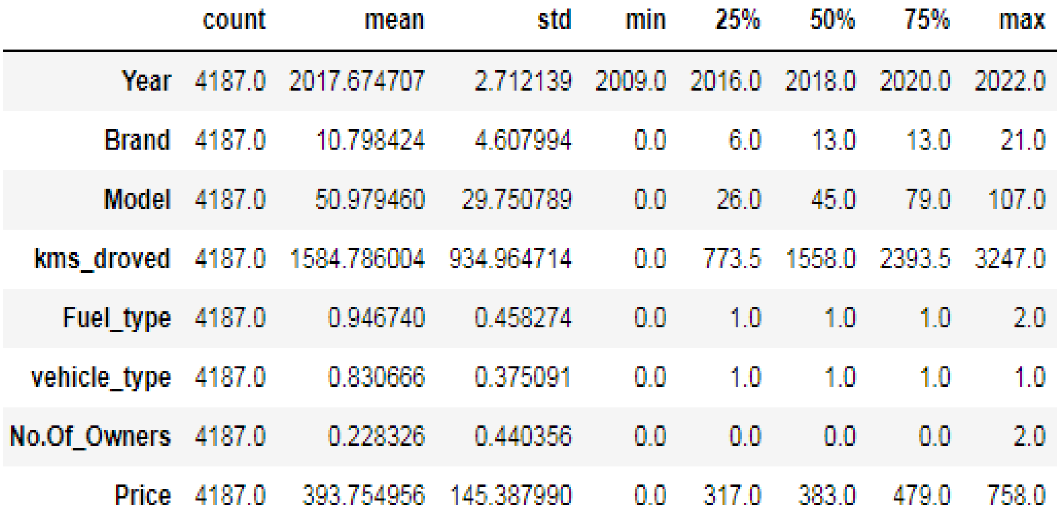
```
Year            0
Brand           0
Model           0
kms_droved      0
Fuel_type       0
vehicle_type    0
No.Of_Owners    0
Price           0
dtype: int64
```

DATA PREPROCESSING

Data pre-processing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre- process our data before feeding it into our model.

STATISTICAL SUMMARY

In descriptive statistics, summary statistics are used to summarize a set of observations, in order to communicate the largest amount of information as simply as possible. Summary statistics summarize and provide information about your sample data. It tells something about the values in data set. This includes where the average lies and whether the data is skewed. The describe() function computes a summary of statistics pertaining to the Data Frame columns. This function gives the mean, count, max, standard deviation and IQR values of the dataset in a simple understandable way.



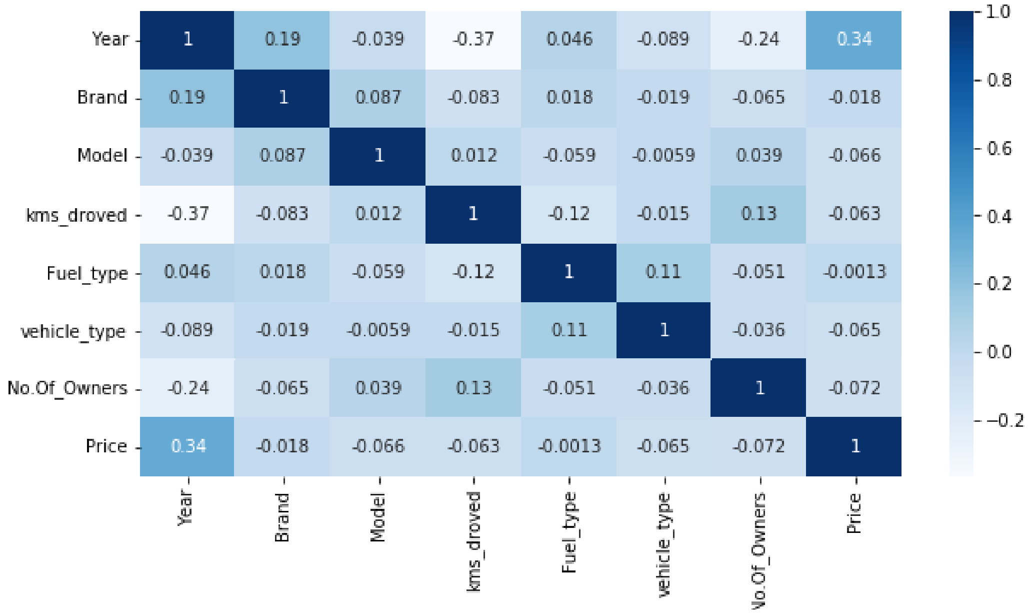
	count	mean	std	min	25%	50%	75%	max
Year	4187.0	2017.674707	2.712139	2009.0	2016.0	2018.0	2020.0	2022.0
Brand	4187.0	10.798424	4.607994	0.0	6.0	13.0	13.0	21.0
Model	4187.0	50.979460	29.750789	0.0	26.0	45.0	79.0	107.0
kms_droved	4187.0	1584.786004	934.964714	0.0	773.5	1558.0	2393.5	3247.0
Fuel_type	4187.0	0.946740	0.458274	0.0	1.0	1.0	1.0	2.0
vehicle_type	4187.0	0.830666	0.375091	0.0	1.0	1.0	1.0	1.0
No.Of_Owners	4187.0	0.228326	0.440356	0.0	0.0	0.0	0.0	2.0
Price	4187.0	393.754956	145.387990	0.0	317.0	383.0	479.0	758.0

Observation:

- Count is same for all the columns.
- For some columns mean is greater than 50% and for some columns mean is lesser than 50%.
- For all the columns there is a difference between max and 75%.

Correlation Factor

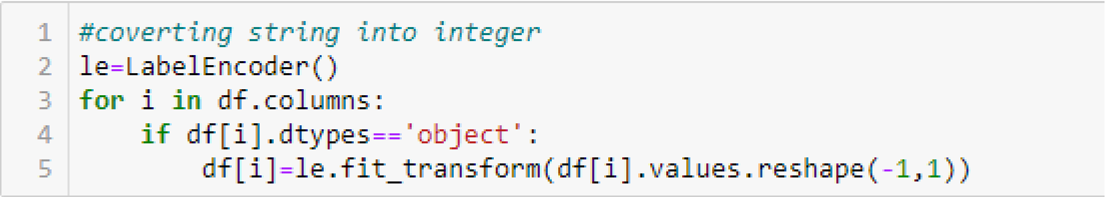
The statistical relationship between two variables is referred to as their correlation. The correlation factor represents the relation between columns in a given dataset. A correlation can be positive, meaning both variables are moving in the same direction or it can be negative, meaning that when one variable's value increasing, the other variable’s value is decreasing.



	Year	Brand	Model	kms_droved	Fuel_type	vehicle_type	No.Of_Owners	Price
Year	1	0.19	-0.039	-0.37	0.046	-0.089	-0.24	0.34
Brand	0.19	1	0.087	-0.083	0.018	-0.019	-0.065	-0.018
Model	-0.039	0.087	1	0.012	-0.059	-0.0059	0.039	-0.066
kms_droved	-0.37	-0.083	0.012	1	-0.12	-0.015	0.13	-0.063
Fuel_type	0.046	0.018	-0.059	-0.12	1	0.11	-0.051	-0.0013
vehicle_type	-0.089	-0.019	-0.0059	-0.015	0.11	1	-0.036	-0.065
No.Of_Owners	-0.24	-0.065	0.039	0.13	-0.051	-0.036	1	-0.072
Price	0.34	-0.018	-0.066	-0.063	-0.0013	-0.065	-0.072	1

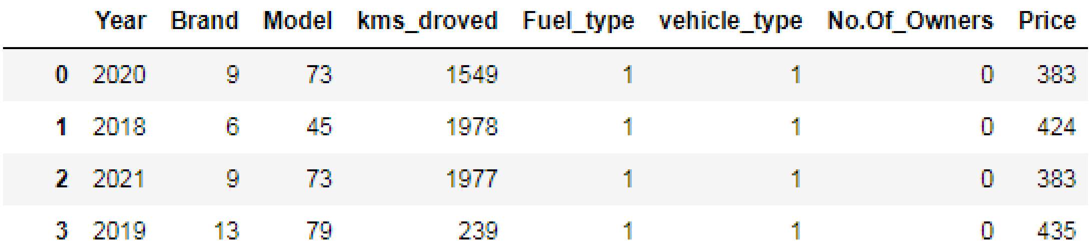
Observation:
1. year is highly positively correlated with target varibale.
2. fuel_type is least correlated with target variable.
3. model and no.of owners is highly negatively correlation with target variable.

ENCODING NON-NUMERIC DATA USING LABEL ENCODER



```
#coverting string into integer
le=LabelEncoder()
for i in df.columns:
    if df[i].dtypes=='object':
        df[i]=le.fit_transform(df[i].values.reshape(-1,1))
```

```
df
```

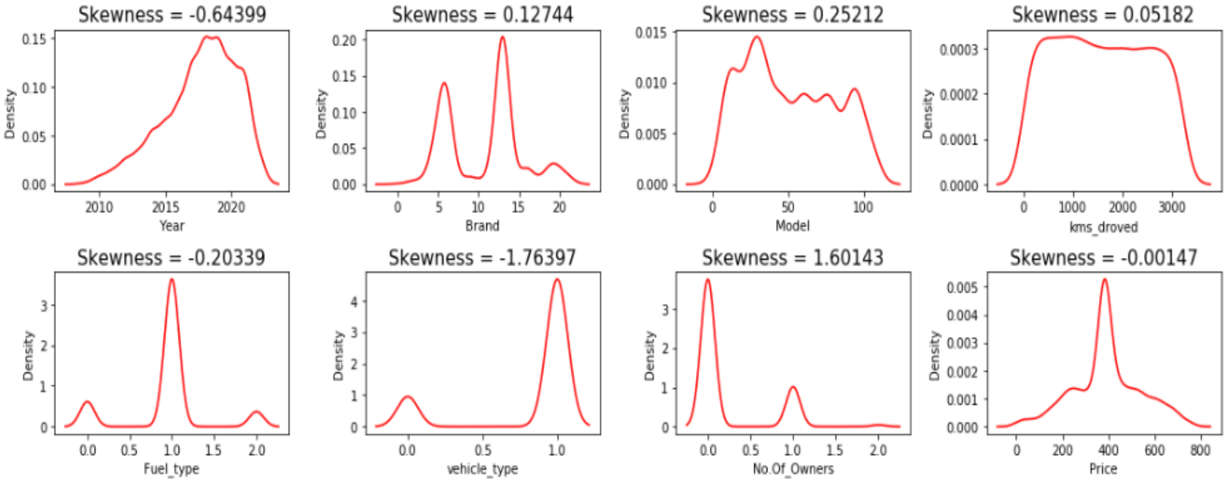


	Year	Brand	Model	kms_droved	Fuel_type	vehicle_type	No.Of_Owners	Price
0	2020	9	73	1549	1	1	0	383
1	2018	6	45	1978	1	1	0	424
2	2021	9	73	1977	1	1	0	383
3	2019	13	79	239	1	1	0	435

Using LabelEncoder() to convert the object dataype into integer datatype so that all the columns can go under the model building process.

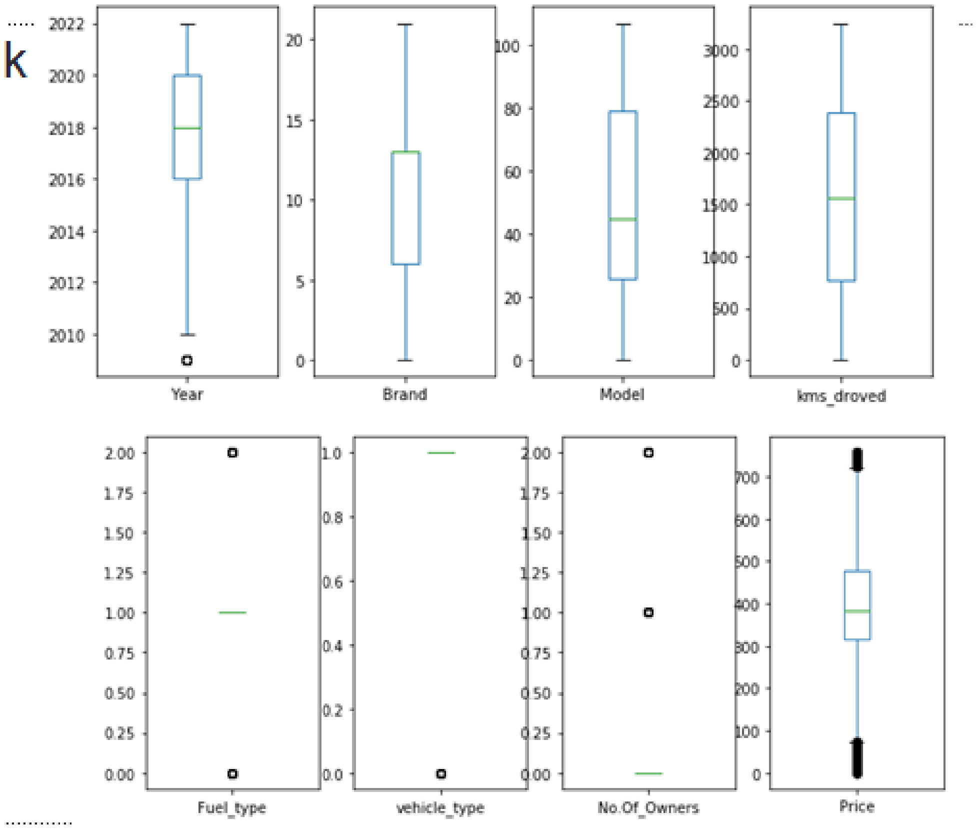
<u>CHECKING SKEWNESS</u>

Skewness refers to distortion or asymmetry in a symmetrical bell curve, or normal distribution in a set of data. Besides positive and negative skew, distributions can also be said to have zero or undefined skew. The skewness value can be positive, zero, negative, or undefined.

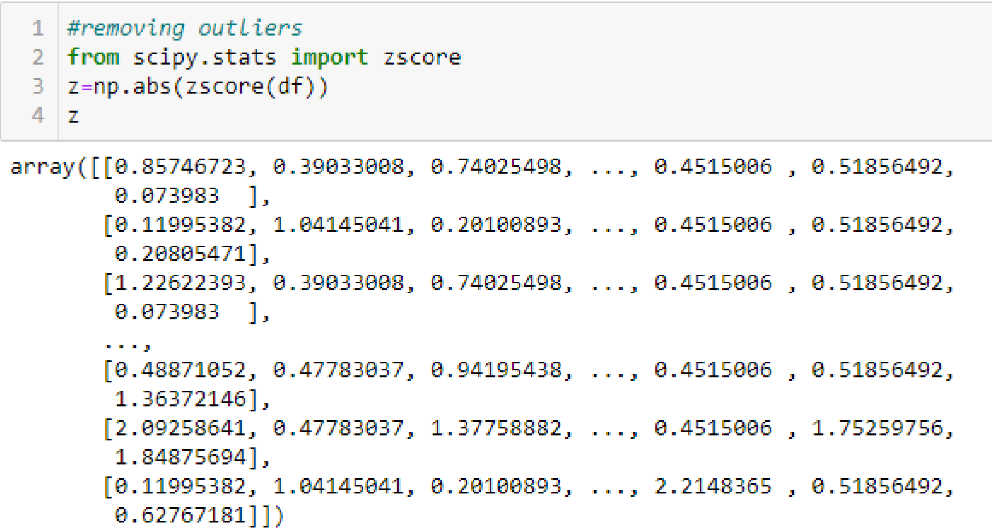


CHECKING OUTLIERS

An outlier is a data point in a data set which is distant or far from all other observations available. It is a data point which lies outside the overall distribution which is available in the dataset.

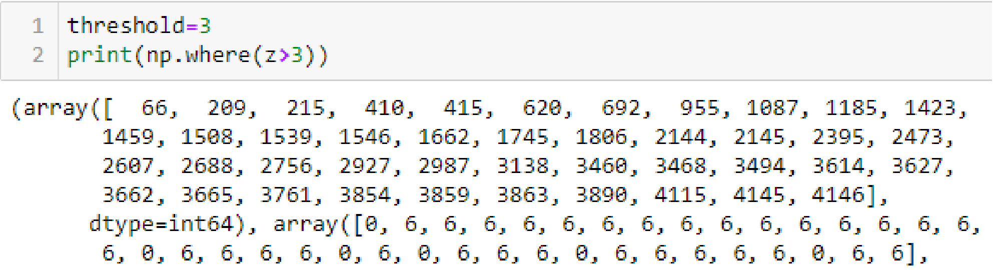


TREATING SKEWNESS & OUTLIERS



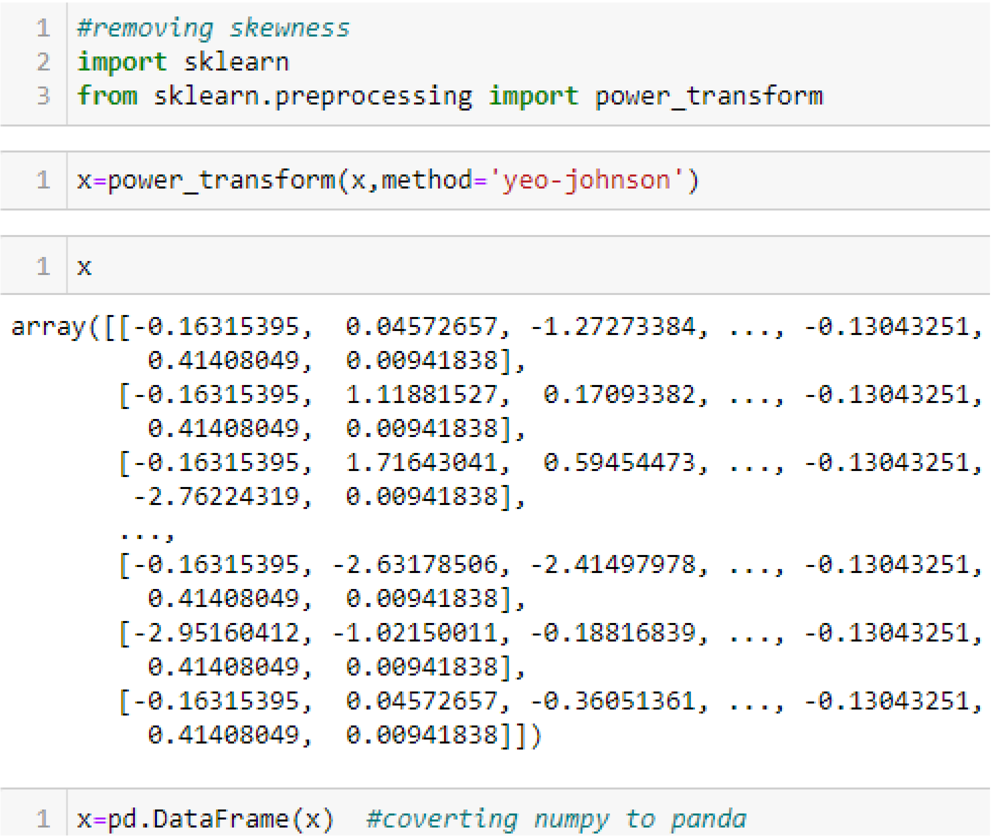
```python
#removing outliers
from scipy.stats import zscore
z=np.abs(zscore(df))
z
```

```
array([[0.85746723, 0.39033008, 0.74025498, ..., 0.4515006 , 0.51856492,
        0.073983  ],
       [0.11995382, 1.04145041, 0.20100893, ..., 0.4515006 , 0.51856492,
        0.20805471],
       [1.22622393, 0.39033008, 0.74025498, ..., 0.4515006 , 0.51856492,
        0.073983  ],
       ...,
       [0.48871052, 0.47783037, 0.94195438, ..., 0.4515006 , 0.51856492,
        1.36372146],
       [2.09258641, 0.47783037, 1.37758882, ..., 0.4515006 , 1.75259756,
        1.84875694],
       [0.11995382, 1.04145041, 0.20100893, ..., 2.2148365 , 0.51856492,
        0.62767181]])
```



```python
threshold=3
print(np.where(z>3))
```

```
(array([  66,  209,  215,  410,  415,  620,  692,  955, 1087, 1185, 1423,
       1459, 1508, 1539, 1546, 1662, 1745, 1806, 2144, 2145, 2395, 2473,
       2607, 2688, 2756, 2927, 2987, 3138, 3460, 3468, 3494, 3614, 3627,
       3662, 3665, 3761, 3854, 3859, 3863, 3890, 4115, 4145, 4146],
      dtype=int64), array([0, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
       6, 0, 6, 6, 6, 6, 0, 6, 0, 6, 6, 6, 0, 6, 6, 6, 6, 6, 0, 6, 6],
```



```python
#removing skewness
import sklearn
from sklearn.preprocessing import power_transform
```

```python
x=power_transform(x,method='yeo-johnson')
```

```python
x
```

```
array([[-0.16315395,  0.04572657, -1.27273384, ..., -0.13043251,
         0.41408049,  0.00941838],
       [-0.16315395,  1.11881527,  0.17093382, ..., -0.13043251,
         0.41408049,  0.00941838],
       [-0.16315395,  1.71643041,  0.59454473, ..., -0.13043251,
        -2.76224319,  0.00941838],
       ...,
       [-0.16315395, -2.63178506, -2.41497978, ..., -0.13043251,
         0.41408049,  0.00941838],
       [-2.95160412, -1.02150011, -0.18816839, ..., -0.13043251,
         0.41408049,  0.00941838],
       [-0.16315395,  0.04572657, -0.36051361, ..., -0.13043251,
         0.41408049,  0.00941838]])
```

```python
x=pd.DataFrame(x)  #coverting numpy to panda
```

Hardware and Software Requirements and Tools Used

For doing this project, the hardware used is a laptop with high end specification and a stable internet connection. While coming to software part, I had used anaconda navigator and in that I have used Jupyter notebook to do my python programming and analysis.
For using a csv file, Microsoft excel is needed. In Jupyter notebook, I had used lots of python libraries to carry out this project and I have mentioned below with proper justification:

1. Pandas- a library which is used to read the data, visualisation and analysis of data.
2. NumPy- used for working with array and various mathematical techniques.
3. Seaborn- visualization tool for plotting different types of plot.
4. Matplotlib- It provides an object-oriented API for embedding plots into applications.
5. zscore- technique to remove outliers.
6. skew ()- to treat skewed data using various transformation like sqrt, log, cube, boxcox, etc.
7. Standard scaler- I used this to scale my data before sending it to model.
8. train_test_split- to split the test and train data.
9. Then I used different classification algorithms to find out the best model for predictions.
10. joblib- library used to save the model in either pickle or obj file.

MODEL/S DEVELOPMENT AND EVALUATION

Identification of possible problem-solving approaches (methods)

From the given dataset it can be concluded that it is a Regression problem as the output column “Price” has continuous output. So,for further analysis of the problem, we have to import or call out the Regression related libraries in Python work frame.

The different libraries used for the problem solving are:
sklearn - Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

1. sklearn.linear_model
i. Linear Regression - Linear regression - is a linear model, e.g., a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

ii. Lasso - In statistics and machine learning, lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

iii. Ridge - The regression is a way to create a parsimonious model when the number of predictor variables in a set exceeds the number of observations, or when a data set has multicollinearity (correlations between predictor variables). Ridge regression is particularly useful to mitigate the problem of multicollinearity in linear regression, which commonly occurs in models with large numbers of parameters. In general, the method provides improved efficiency in parameter estimation problems in exchange for a tolerable amount of bias.

iv. Elastic Net - It is a popular type of regularized linear regression that combines two popular penalties, specifically the L1 and L2 penalty functions. Elastic net linear regression uses the penalties from both the lasso and ridge techniques to regularize regression models. The technique combines both the lasso and ridge regression methods by learning from their shortcomings to improve on the regularization of statistical models.

2. sklearn.tree –
Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a

model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
There are several advantages of using decision trees for predictive analysis:

- Decision trees can be used to predict both continuous and discrete values i.e., they work well for both regression and classification tasks.
- They require relatively less effort for training the algorithm.
- They can be used to classify non-linearly separable data.
- They're very fast and efficient compared to KNN and other algorithms.

Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree to go from observations about an item to conclusions about the item's target value.

Decision Tree Regressor - Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application. It is a tree-structured classifier with three types of nodes.

3. sklearn.ensemble

The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator.
The different types of ensemble techniques used in the model are:

i. Random Forest Regressor - It is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. A Random Forest's nonlinear nature can give it a leg up over linear algorithms, making it a great option. Random forest is a type of supervised learning algorithm that uses ensemble methods (bagging) to solve both regression and classification problems.

ii. AdaBoost Regressor - It is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction.

iii. 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.

4. sklearn.metrics - The sklearn. metrics module implements several losses, score, and utility functions to measure classification performance. Some metrics might require probability estimates of the positive class, confidence values, or binary decisions values. Important sklearn.metrics modules used in the project are:

- **mean_absolute_error**
- **mean_squared_error**
- **r2_score**

5. sklearn.model_selection –
i. GridSearchCV - It is a library function that is a member of sklearn's model_selection package. It helps to loop through predefined hyper parameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters. GridSearchCV combines an estimator with a grid search preamble to tune hyper-parameters. The method picks the optimal parameter from the grid search and uses it with the estimator selected by the user.

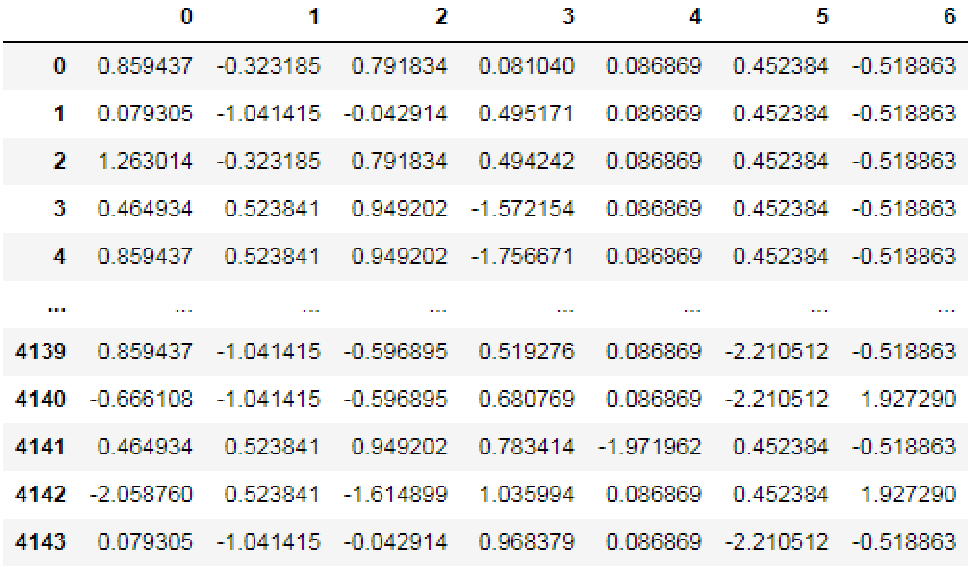
ii. cross_val_score - Cross validation helps to find out the over fitting and under fitting of the model. In the cross validation the model is made to run on different subsets of the dataset which will get multiple measures of the model. If we take 5 folds, the data will be divided into 5 pieces where each part being 20% of full dataset. While running the Cross validation the 1st part (20%) of the 5 parts will be kept out as a holdout set for validation and everything else is used for training data.

Testing of Identified Approaches

After completing the required pre-processing techniques for the model building data is separated as input and output columns before passing it to the train_test_split.

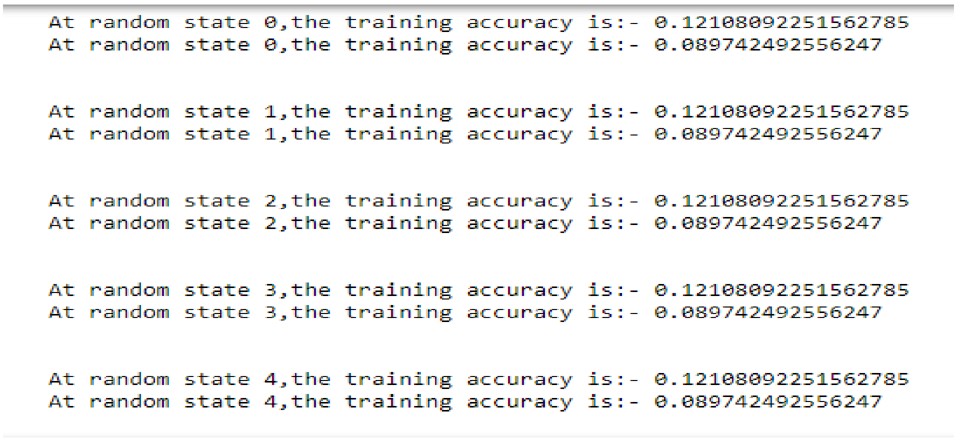
Scaling the data using Standard Scaler

For each value in a feature, StandardScaler subtracts the minimum value in the feature and then divides by the range. The range is the difference between the original maximum and original minimum. StandardScaler preserves the shape of the original distribution.



	0	1	2	3	4	5	6
0	0.859437	-0.323185	0.791834	0.081040	0.086869	0.452384	-0.518863
1	0.079305	-1.041415	-0.042914	0.495171	0.086869	0.452384	-0.518863
2	1.263014	-0.323185	0.791834	0.494242	0.086869	0.452384	-0.518863
3	0.464934	0.523841	0.949202	-1.572154	0.086869	0.452384	-0.518863
4	0.859437	0.523841	0.949202	-1.756671	0.086869	0.452384	-0.518863
...	...	...	...	...	...	...	...
4139	0.859437	-1.041415	-0.596895	0.519276	0.086869	-2.210512	-0.518863
4140	-0.666108	-1.041415	-0.596895	0.680769	0.086869	-2.210512	1.927290
4141	0.464934	0.523841	0.949202	0.783414	-1.971962	0.452384	-0.518863
4142	-2.058760	0.523841	-1.614899	1.035994	0.086869	0.452384	1.927290
4143	0.079305	-1.041415	-0.042914	0.968379	0.086869	-2.210512	-0.518863

CHECKING THE RANDOM STATE



```
At random state 0,the training accuracy is:- 0.1210809225l562785
At random state 0,the training accuracy is:- 0.089742492556247


At random state 1,the training accuracy is:- 0.12108092251562785
At random state 1,the training accuracy is:- 0.089742492556247


At random state 2,the training accuracy is:- 0.12108092251562785
At random state 2,the training accuracy is:- 0.089742492556247


At random state 3,the training accuracy is:- 0.12108092251562785
At random state 3,the training accuracy is:- 0.089742492556247


At random state 4,the training accuracy is:- 0.12108092251562785
At random state 4,the training accuracy is:- 0.089742492556247
```

Selecting random state as 4. We are getting same values in allthe random states.

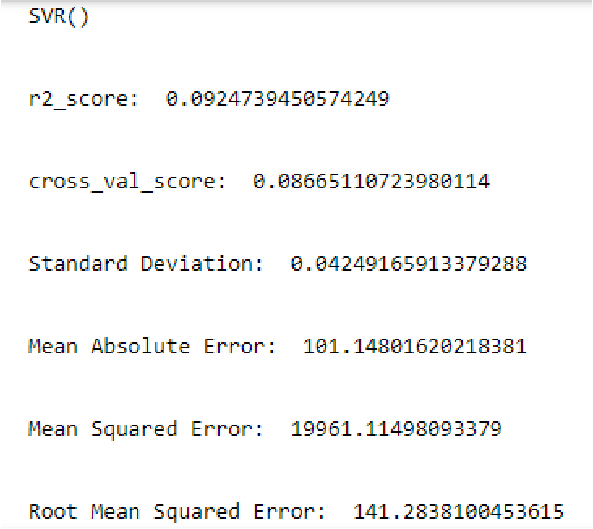
Train Test Split

The train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data.

FINDING THE BEST MODEL

We are using LinearRegressor, Lasso, Ridge, ElasticNet, Support vector, Decision Tree and Kneighbors Regressor.

We are calculating r2 score, cross_val_score, std, mean_absolute error, mean_squared error, root_mean_squared error for each models.



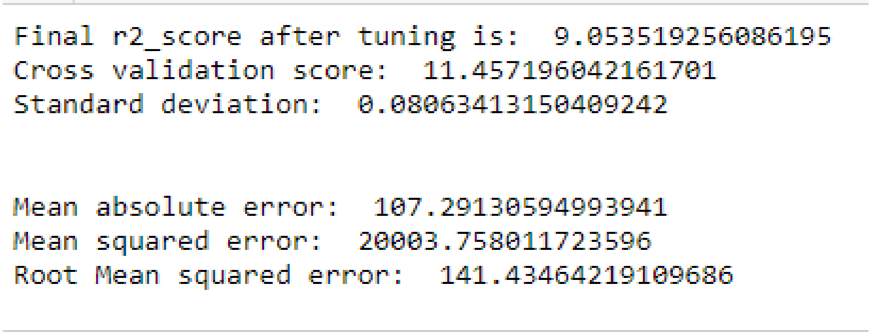
```
SVR()


r2_score:  0.0924739450574249


cross_val_score:  0.08665110723980114


Standard Deviation:  0.04249165913379288


Mean Absolute Error:  101.14801620218381


Mean Squared Error:  19961.11498093379


Root Mean Squared Error:  141.2838100453615
```

We can see that Support Vector Regression algorithms are performing well as compared to other models.

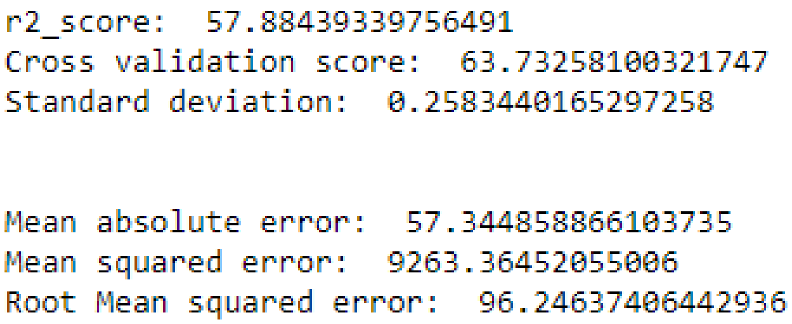
5. Hyperparameter Tuning:

There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as **Hyperparameters.** These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. You must select from a specific list of hyperparameters for a given model as it varies from model to model.

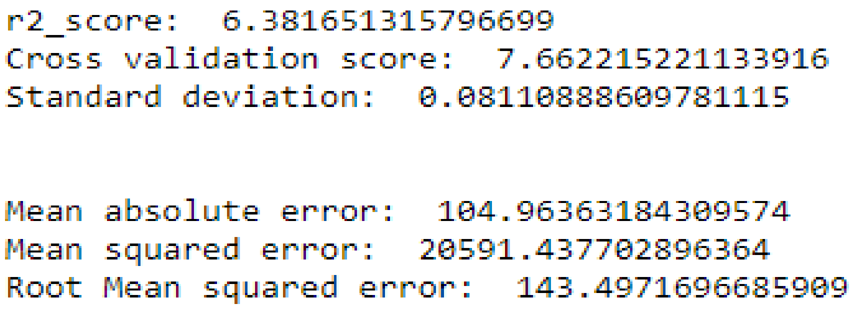
GridSearchCV is a function that comes in Scikit-learn (or SK-learn) model selection package. An important point here to note is that we need to have Scikit-learn library installed on the computer. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.



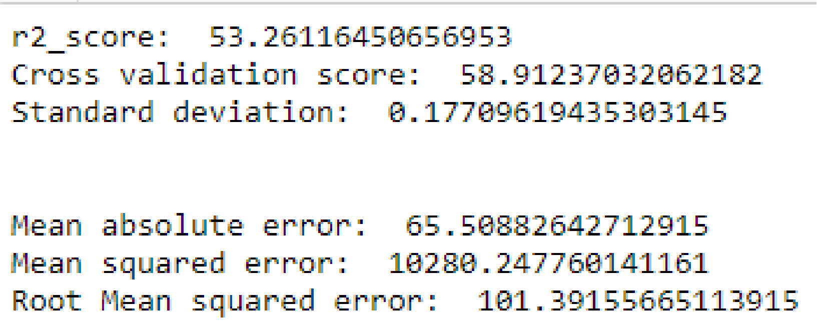
```
Final r2_score after tuning is:  9.053519256086195
Cross validation score:  11.457196042161701
Standard deviation:  0.08063413150409242


Mean absolute error:  107.29130594993941
Mean squared error:  20003.758011723596
Root Mean squared error:  141.43464219109686
```



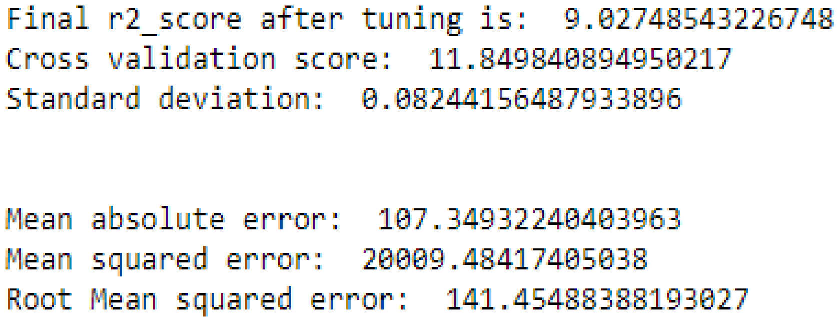
```
r2_score:  57.88439339756491
Cross validation score:  63.73258100321747
Standard deviation:  0.2583440165297258


Mean absolute error:  57.344858866103735
Mean squared error:  9263.36452055006
Root Mean squared error:  96.24637406442936
```



```
r2_score:  6.381651315796699
Cross validation score:  7.662215221133916
Standard deviation:  0.08110888609781115


Mean absolute error:  104.96363184309574
Mean squared error:  20591.437702896364
Root Mean squared error:  143.4971696685909
```



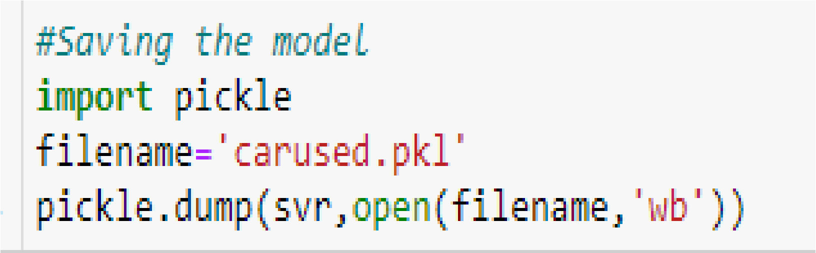
```
r2_score:  53.26116450656953
Cross validation score:  58.91237032062182
Standard deviation:  0.17709619435303145


Mean absolute error:  65.50882642712915
Mean squared error:  10280.247760141161
Root Mean squared error:  101.39155665113915
```



```
Final r2_score after tuning is:  9.02748543226748
Cross validation score:  11.849840894950217
Standard deviation:  0.08244156487933896


Mean absolute error:  107.34932240403963
Mean squared error:  20009.48417405038
Root Mean squared error:  141.45488388193027
```

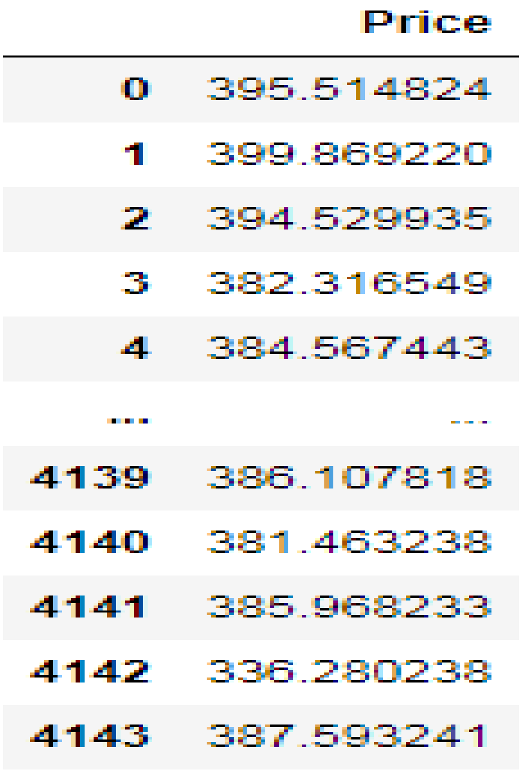
After applying Ensemble Techniques, we can see that SupportVectorRegressor is the best performing algorithm amongall other algorithms as it is giving a r2_score of 92.01 and cross validation score of 86.43

SAVING THE BEST MODEL



```
#Saving the model
import pickle
filename='carused.pkl'
pickle.dump(svr,open(filename,'wb'))
```

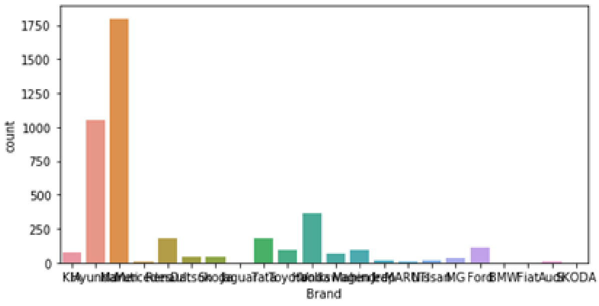
PREDICTING OVER TEST DATA

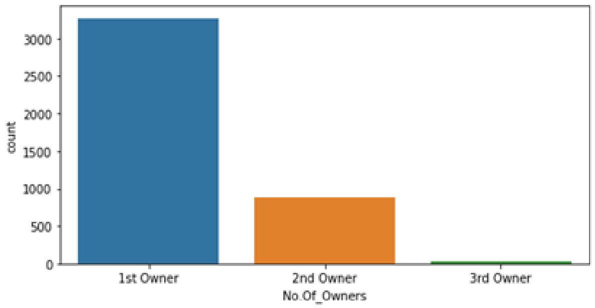


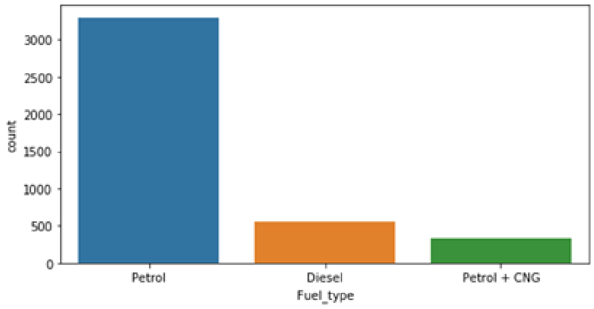
	Price
0	395.514824
1	399.869220
2	394.529935
3	382.316549
4	384.567443
...	...
4139	386.107818
4140	381.463238
4141	385.968233
4142	336.280238
4143	387.593241

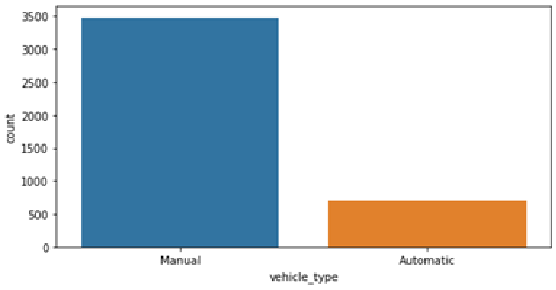
DATA VISUALIZATION

PLOTTING GRAPHS FOR CATEGORICAL DATA

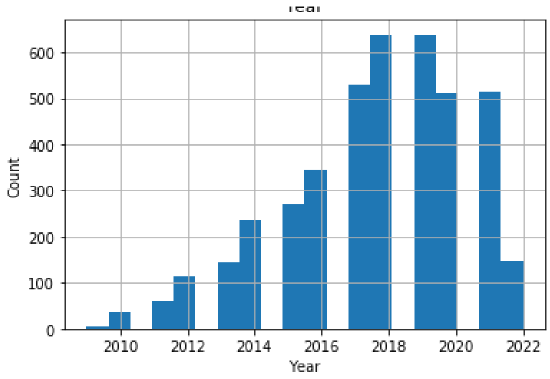








PLOTTING GRAPHS FOR CONTINUOUS DATA



Observations:

- 1. maximum cars to be selled are of year 2018 to 2020 of count almost 600 each.
- 2. 1st owner has the maximum count of 3268, then followed by 2nd owner count of 882 and 3rd owner count of 37.
- 3. maximum cars to be selled are manual with count of 3478 and rest all the 708 cars are automatic.
- 4. 3069 cars are petrol, 557 cars are diesel and rest 337 cars are petrol+cng.
- 5. maximum cars are of maruti, followed by hundai and least are SKODA and fiat.
- 6. alto, swift, baleno and grand models are maximum for selling.
- 7. Q, tiago, camry and carens models are very less for selling.

CONCLUSION

- The manufacturer like Land Rover, Benz, BMW cars are costliest used car in the market comparatively to other cars.
- The low kilometres driven and also if the manufacturing year is lesser on these brands those card sells in much higher rates or closest to the buying new car rates.
- The Diesel variant and Automatic shift variants are also costliest user car variants in the used car market.

Learning Outcomes of the Study in respect of Data Science

The above research will help our client to study about the latest used car market and with the help of the model built he can easily predict the price ranges of the cars, and also will helps him to understand based on what factors the Car Price is decided.

Limitations of this work and Scope for Future Work

The limitation of the study is that in the volatile changing market we have taken the data, to be more precise we have taken the data at the time of pandemic, so when the pandemic ends the market correction might happen slowly. So based on that again the deciding factors of the used car prize might change and we have shortlisted and taken these data from the important cities across India, if the seller is from the different city our model might fail to predict the accurate prize of that used car.

THANK YOU