FLIP ROBO

Used Car Price Prediction

Submitted By

Jessica Ghimeliya

ACKNOWLEDGMENT

I would like to express my deep sense of gratitude to my SME (Subject Matter Expert) **Miss. Sapna Verma** as well as **Flip Robo Technologies** who gave me the golden opportunity to do this data analysis project on **CAR PRICE PREDICTION**, which also helped me in doing lots of research and I came to know about so many new things.

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, and changes to the axle load norms, liquidity crunch, low-capacity utilization 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. Page 4 | 24

• 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 catalyze a big move towards electric vehicles (EVs). That could be the big positive for auto sector.

Conceptual Background of the Domain Problem:

The growing world of e-commerce is not just restricted to buying electronics and clothing but everything that you expect in a general store. Keeping the general store perspective aside and looking at the bigger picture, every day there are thousands or perhaps millions of deals happening in the digital marketplace. One of the most booming markets in the digital space is that of the automobile industry wherein the buying and selling of used cars take place. Sometimes we need to walk up to the dealer or individual sellers to get a used car price quote. However, buyers and sellers face a major stumbling block when it comes to their used car valuation or say their second   hand car valuation. Traditionally, you would go to a showroom and get your vehicle inspected before learning about the price. So instead of doing all these stuffs we can build machine learning model using different features of the used cars to predict the exact and valuable car price.

Review of Literature

This project is more about exploration, feature engineering and classification 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. 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 increasingly digital world.

Motivation for the Problem Undertaken

Based on the problem statement and the real time data scrapped from the OLX and Cars24 websites, I have understood how each independent feature helped me to understand the data as each feature provides a different kind of information. It is so interesting to work with different types of real time data in a single data set and perform root cause analysis to predict the price of the used car. Based on the analysis of the model of the car, kilometers driven, transmission type, fuel type etc. I would be able to model the price of used car as this model will then be used by the client to understand how exactly the prices vary with the variables. They can accordingly work on it and make some strategies to sell the used car and get some high returns. Furthermore, the model will be a good way for the client to understand the pricing dynamics of a used car.

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

In our scrapped dataset, our target variable "Used Car Price” is a continuous variable. Therefore, we will be handling this modeling problem as regression.

This project is done in two parts:

- Data Collection phase
- Model Building 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, OLA, CarDekho, Cars24 etc.) we need web scraping for this. We have to fetch data for different locations. The number of columns for data doesn’t have limit, it’s up to us and our creativity. Generally, these columns are Brand, model, variant, manufacturing year, driven kilometers, fuel, the number of owners, location and at last target variable Price of the car. This data is to give a hint about important variables in used car model. We can make changes to it, we can add or we can remove some columns, it completely depends on the website from which we are fetching the data. Trying to include all types of cars in our data for example- SUV, Sedans, Coupe, minivan, Hatchback.

- Model Building phase:

  After collecting the data, we need to build a machine learning model. Before model building do all data pre-processing steps. Trying different models with different hyper parameters and selecting the best model. Following the complete life cycle of data science including all the below steps mentioned:

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

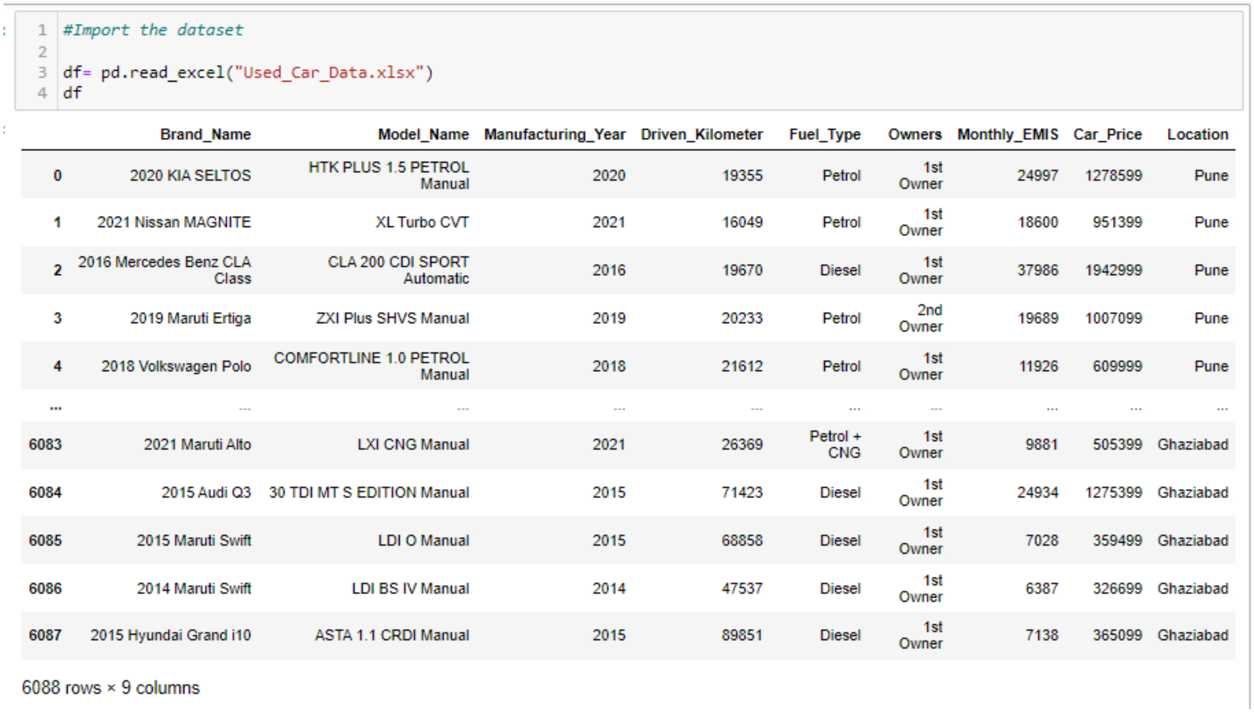
Data Sources and their formats

The dataset is in the form of Excel Sheet format and consists of 9 columns (8 features and 1 label) with 5000+ number of records as explained below:

- Used Car Brand – This show the car brand names
- Used Car Model - This show the car model names
- Year of Manufacture - Gives us the year in which the car was made
- Kilometers Driven - Number of kilometers the car the driven reflecting on the Odometer
- Fuel Type - Shows the fuel type used by the vehicle
- Owners - This show number of owners who sell the car
- Monthly EMIs - This show the price of monthly EMIs to buy a car
- Location – In which location these used car are sold
- Used Car Price - Lists the selling price of the used cars

We can see our dataset includes a target label "Car Price" column and the remaining feature columns can be used to determine or help in predicting the price of the used cars. Since price is a continuous value it makes this to be a Regression problem.

Loading the Dataset



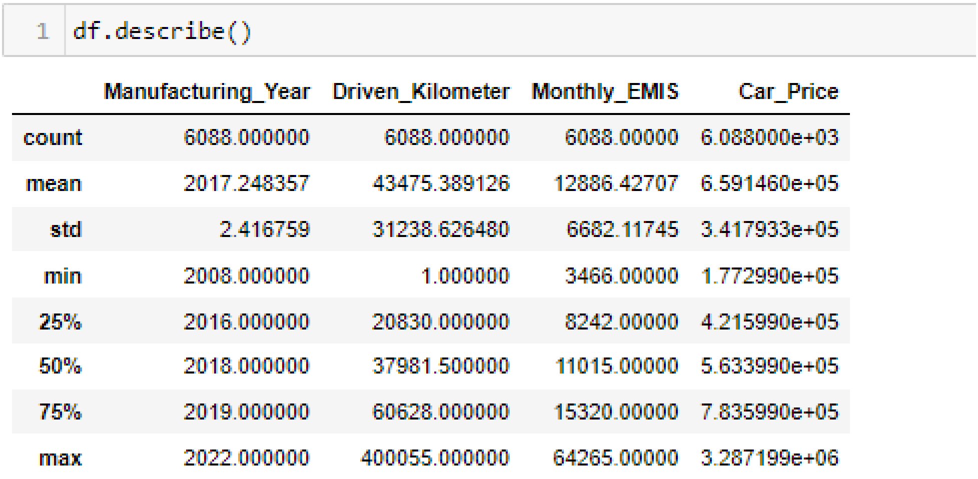
```
#Import the dataset

df= pd.read_excel("Used_Car_Data.xlsx")
df
```

	Brand_Name	Model_Name	Manufacturing_Year	Driven_Kilometer	Fuel_Type	Owners	Monthly_EMIS	Car_Price	Location
0	2020 KIA SELTOS	HTK PLUS 1.5 PETROL Manual	2020	19355	Petrol	1st Owner	24997	1278599	Pune
1	2021 Nissan MAGNITE	XL Turbo CVT	2021	16049	Petrol	1st Owner	18600	951399	Pune
2	2016 Mercedes Benz CLA Class	CLA 200 CDI SPORT Automatic	2016	19670	Diesel	1st Owner	37986	1942999	Pune
3	2019 Maruti Ertiga	ZXI Plus SHVS Manual	2019	20233	Petrol	2nd Owner	19689	1007099	Pune
4	2018 Volkswagen Polo	COMFORTLINE 1.0 PETROL Manual	2018	21612	Petrol	1st Owner	11926	609999	Pune
...	...	...	...	...	...	...	...	...	...
6083	2021 Maruti Alto	LXI CNG Manual	2021	26369	Petrol + CNG	1st Owner	9881	505399	Ghaziabad
6084	2015 Audi Q3	30 TDI MT S EDITION Manual	2015	71423	Diesel	1st Owner	24934	1275399	Ghaziabad
6085	2015 Maruti Swift	LDI O Manual	2015	68858	Diesel	1st Owner	7028	359499	Ghaziabad
6086	2014 Maruti Swift	LDI BS IV Manual	2014	47537	Diesel	1st Owner	6387	326699	Ghaziabad
6087	2015 Hyundai Grand i10	ASTA 1.1 CRDI Manual	2015	89851	Diesel	1st Owner	7138	365099	Ghaziabad

6088 rows × 9 columns

The complete dataset is imported in variable name ‘df’ and we can see there are some columns are need to alter them into numeric data because machine learning will not work with object data type but before that need to analyze the data.

Exploratory Data Analysis



```
df.describe()
```

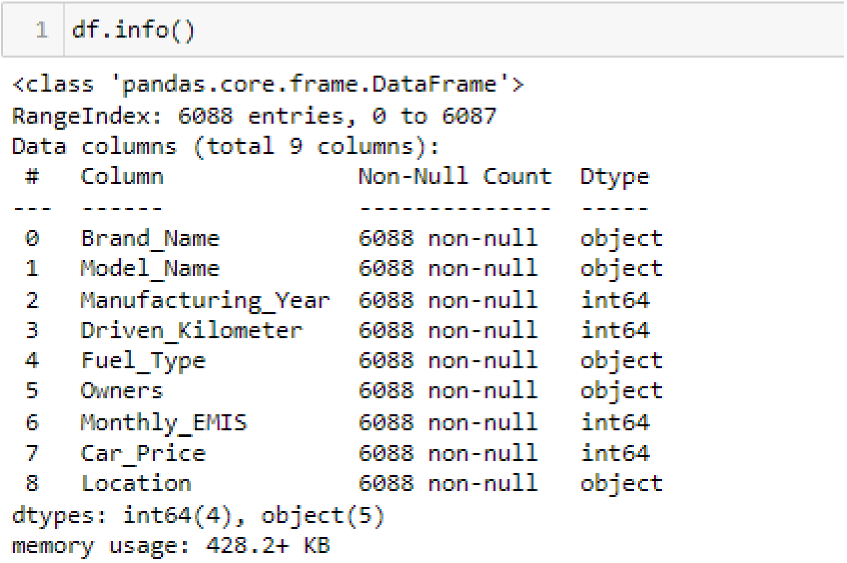
	Manufacturing_Year	Driven_Kilometer	Monthly_EMIS	Car_Price
count	6088.000000	6088.000000	6088.00000	6.088000e+03
mean	2017.248357	43475.389126	12886.42707	6.591460e+05
std	2.416759	31238.626480	6682.11745	3.417933e+05
min	2008.000000	1.000000	3466.00000	1.772990e+05
25%	2016.000000	20830.000000	8242.00000	4.215990e+05
50%	2018.000000	37981.500000	11015.00000	5.633990e+05
75%	2019.000000	60628.000000	15320.00000	7.835990e+05
max	2022.000000	400055.000000	64265.00000	3.287199e+06

Express more about the dataset:

a. As we can see in monthly EMIs column minimum value is 2812 and 75% of that entire column is 14561 but maximum value (100%) is very big i.e., 65376 as compare to all values so it will contain skewness.

b. Almost all columns are having skewness in it.

Similarly, we can conclude so many things by just observing at the describe function.



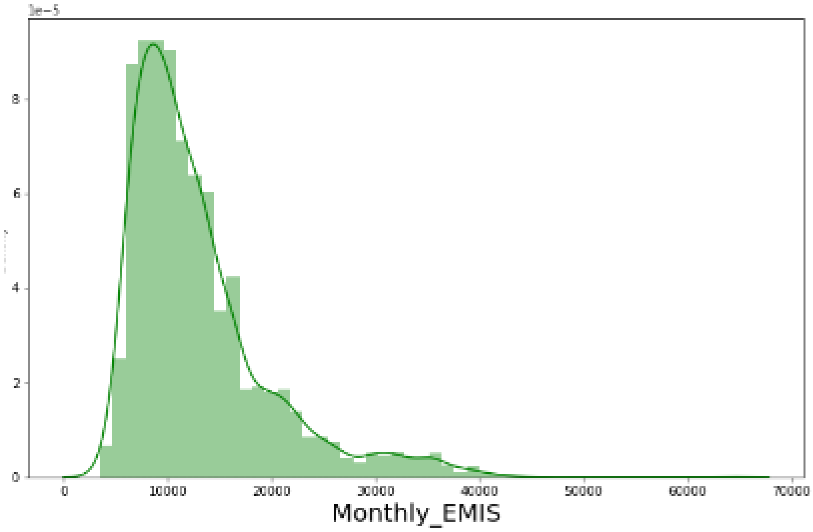
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6088 entries, 0 to 6087
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Brand_Name          6088 non-null   object
 1   Model_Name          6088 non-null   object
 2   Manufacturing_Year  6088 non-null   int64 
 3   Driven_Kilometer    6088 non-null   int64 
 4   Fuel_Type           6088 non-null   object
 5   Owners              6088 non-null   object
 6   Monthly_EMIS        6088 non-null   int64 
 7   Car_Price           6088 non-null   int64 
 8   Location            6088 non-null   object
dtypes: int64(4), object(5)
memory usage: 428.2+ KB
```

- There are 6088 Rows and 9 Columns in dataset
- We can see no columns are having null values.
- We can find here only four columns are having object data types and rest four columns are having integer data type.

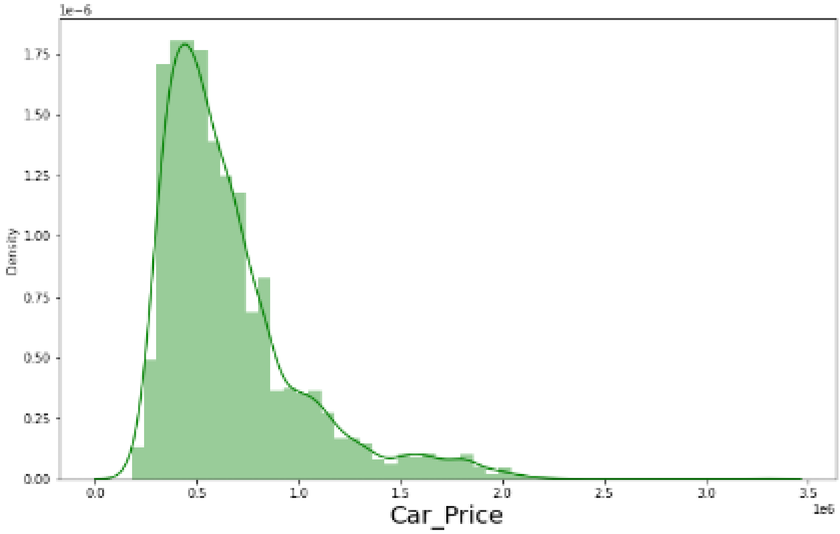
Data Inputs- Logic- Output Relationships (Visualization of data): Univariate analysis:

- What is easiest monthly pay EMI according to sites? Or most important factor for sale price?



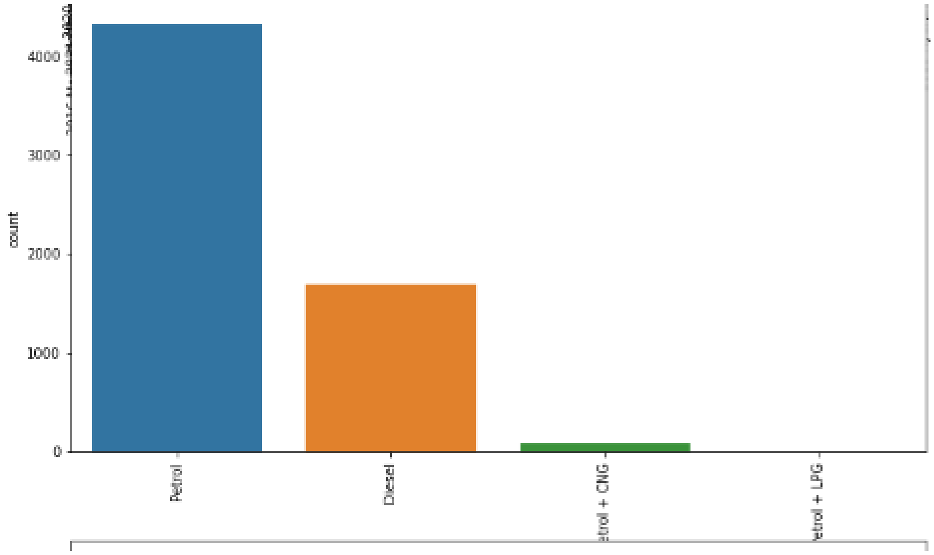
Maximum cars are having MONTHLY EMIS in between 10000rs to 20000rs with peak of 10000rs. As we know this is easy to pay for everyone who are capable of buying cars. This column is having skewness also.

- Car Price Range is?



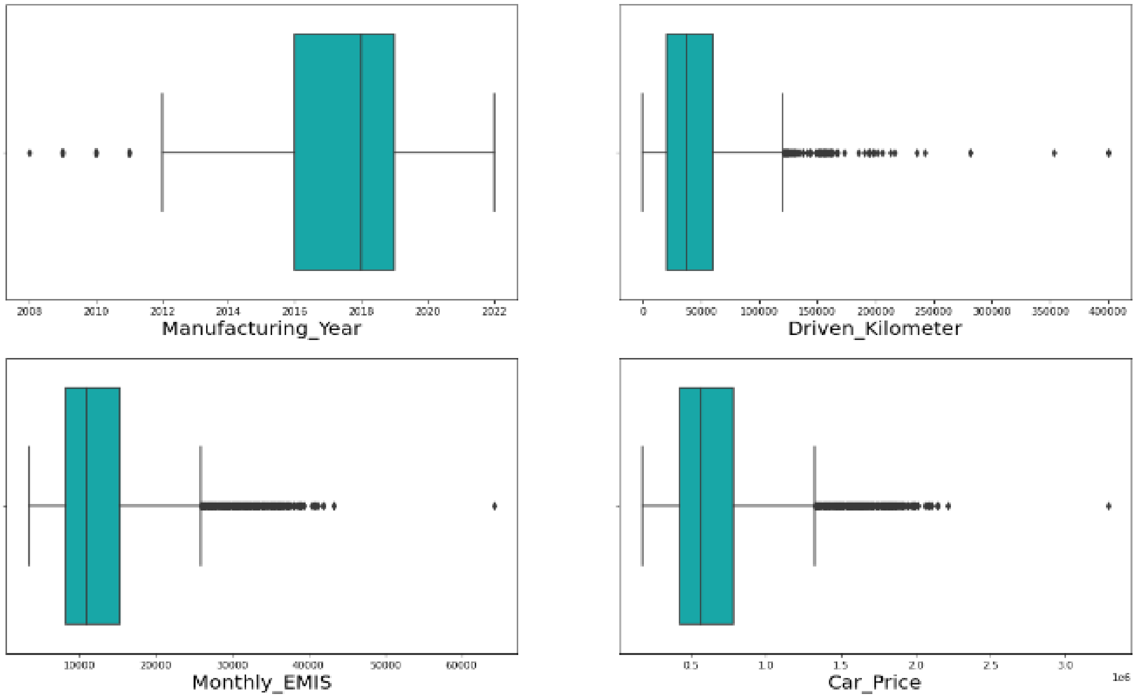
Maximum used cars prices range are in between 2lakhs to 7lakhs with peak of 4 - 5lakhs. It was start from less than 1lakh to the 30lakhs.

- Fuel type of cars?

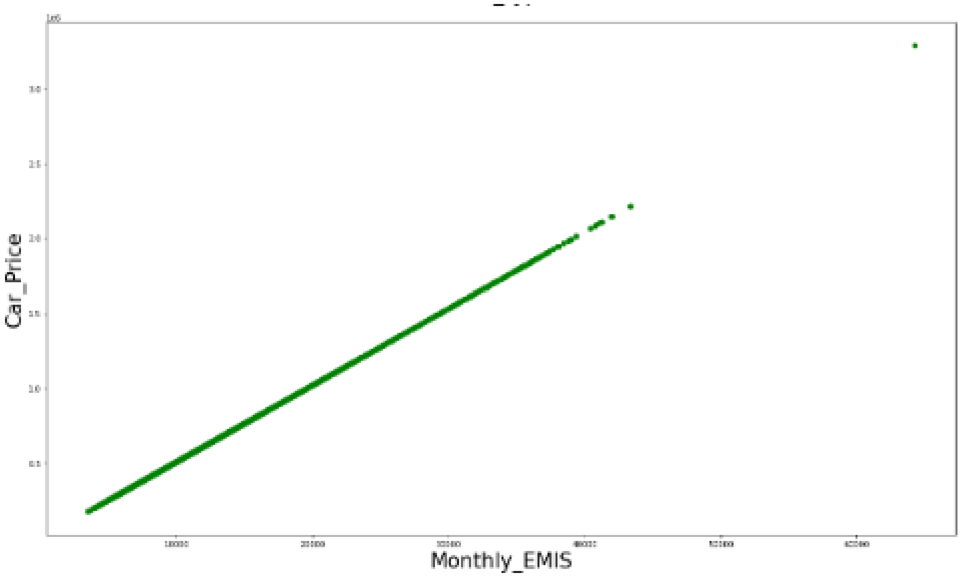


You can see that in FUEL TYPE has a high probability of petrol used cars because as we know petrol cars are less costly as compare to diesel cars and Petrol+CNG cars are having very less amount of cars. Similarly, Owners of cars having maximum first owners only following with 2nd owner and 5th and 4th are very least.

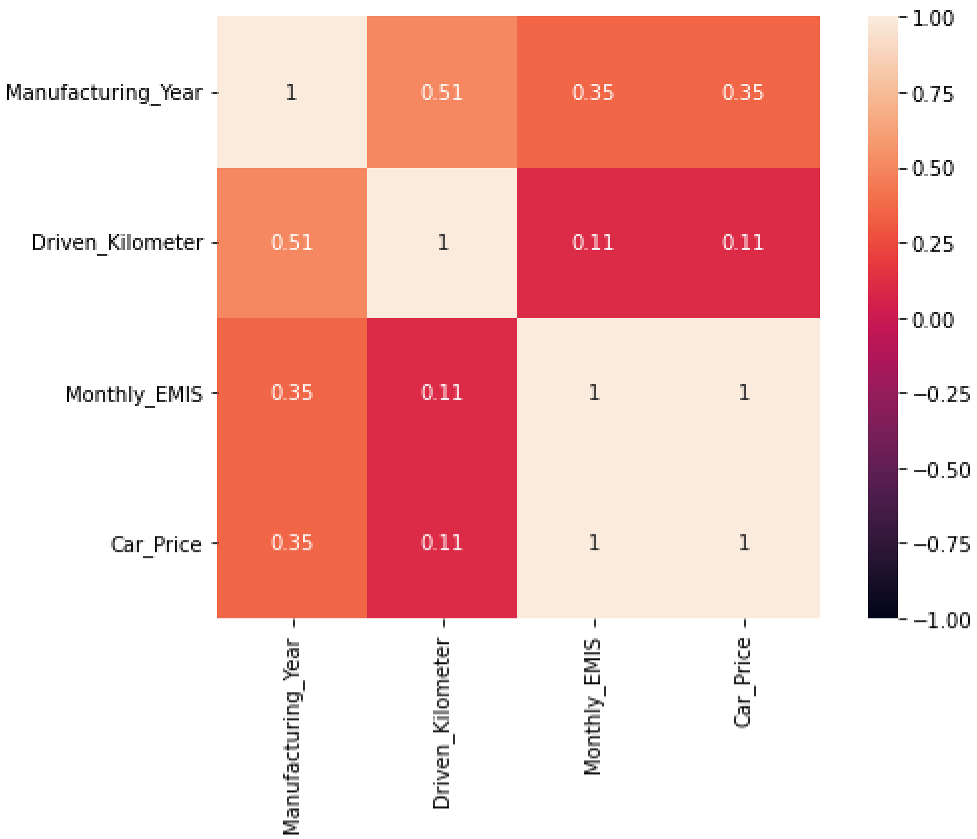
- Check is there outliers are present in dataset?



Driven Kilometers, Monthly EMIS and Manufacturing Year columns are having outliers which we need to treat that with Z score technique which removes the outliers.



Monthly EMIs is having very linear relationship with target variable i.e., Car price.



As shown in heat map Monthly EMIS is highly correlated with target variable i.e. Car price. Hence, it is an important variable to predict the Car price.

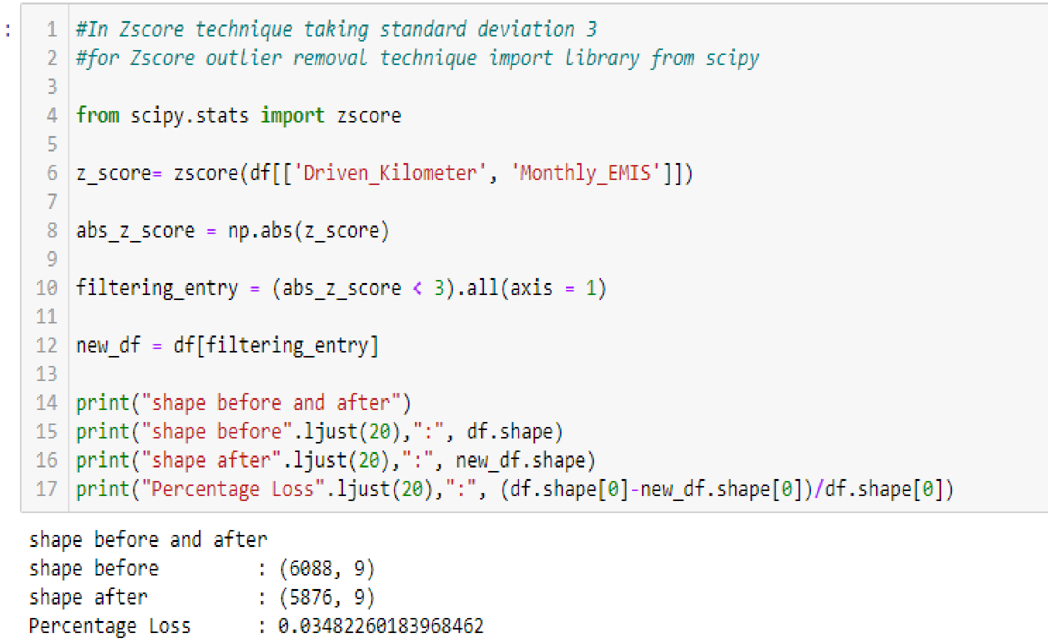
- Data Pre-processing Done

  - Encode the columns which are having object data type

    Encoding data is very important for model because data will not work with object data type. In this project I will encode the object columns with label encoder. Label encoder is work on alphabetical order and encodes the column. I am encoding Brand name, model name column.

  - Remove outliers using Z score technique

    We can use any technique to remove the outliers like z score technique or Inter Quartile range method but our first preference should be less data loss. We have to go with that technique in which less data will be lost.



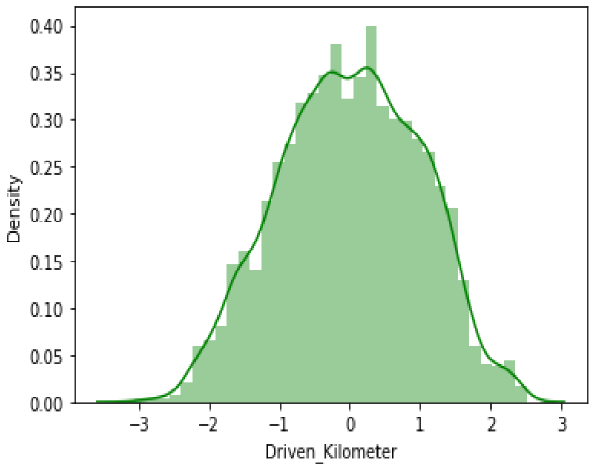
```
#In Zscore technique taking standard deviation 3
#for Zscore outlier removal technique import library from scipy

from scipy.stats import zscore

z_score= zscore(df[['Driven_Kilometer', 'Monthly_EMIS']])

abs_z_score = np.abs(z_score)

filtering_entry = (abs_z_score < 3).all(axis = 1)

new_df = df[filtering_entry]

print("shape before and after")
print("shape before".ljust(20),":", df.shape)
print("shape after".ljust(20),":", new_df.shape)
print("Percentage Loss".ljust(20),":", (df.shape[0]-new_df.shape[0])/df.shape[0])
```

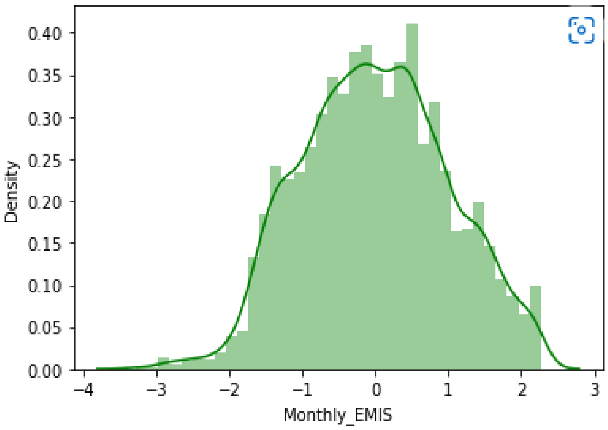
```
shape before and after
shape before         : (6088, 9)
shape after          : (5876, 9)
Percentage Loss      : 0.03482260183968462
```

Shown above 3.48% of data will loss after applying Z-score technique.

- Remove skewness

After removing skewness:





For continuous columns which are having skewness value greater than +0.5 to -0.5 then it is important to skewness from that.

- Hardware and Software Requirements and Tools Used

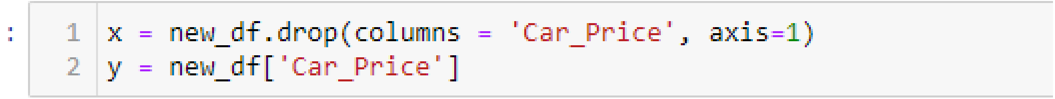
a. Notebook and Data: GitHub, Jupyter Notebook

b. Libraries: Numpy, pandas, Sklearn, scipy, joblib, Matplotlib, seaborn, statsmodels

Model/s Development and Evaluation

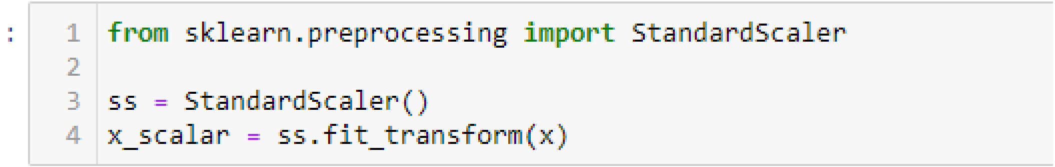
- ## Identification of possible problem-solving approaches (methods)

Now we will train several Machine Learning models and compare their results. We need to use the predictions on the training set to compare the algorithms with each other. Later on, we will use cross validation. After that I will compare the difference of accuracy score and cross validation score of all models. The model which is having minimum difference will be considered as a best model and further procedures will do on that model. Following are the basic steps of model building.



```python
x = new_df.drop(columns = 'Car_Price', axis=1)
y = new_df['Car_Price']
```

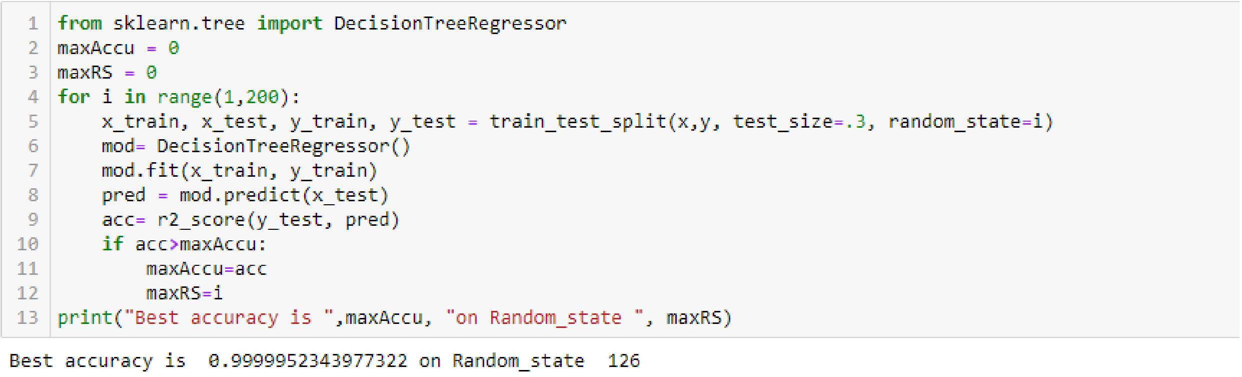
Scalling technique



```python
from sklearn.preprocessing import StandardScaler

ss = StandardScaler()
x_scalar = ss.fit_transform(x)
```

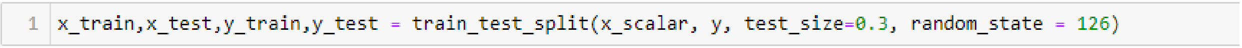
After scaling technique, the independent variables are getting normalized and standardized. Now, check the variance inflation factor to avoid multicollinearity problem. In this model I am having all the values if variance inflation factor is less than 10 so safe to proceed further.

- Testing of Identified Approaches (Algorithms)
- Find best random state: first of all, find the best random state



```python
from sklearn.tree import DecisionTreeRegressor
maxAccu = 0
maxRS = 0
for i in range(1,200):
    x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=.3, random_state=i)
    mod= DecisionTreeRegressor()
    mod.fit(x_train, y_train)
    pred = mod.predict(x_test)
    acc= r2_score(y_test, pred)
    if acc>maxAccu:
        maxAccu=acc
        maxRS=i
print("Best accuracy is ",maxAccu, "on Random_state ", maxRS)
```

```
Best accuracy is  0.9999952343977322 on Random_state  126
```



```python
x_train,x_test,y_train,y_test = train_test_split(x_scalar, y, test_size=0.3, random_state = 126)
```

Now, splitting our dataset by train test split with random state: 126

- Run and evaluate selected models

1. Linear regression model

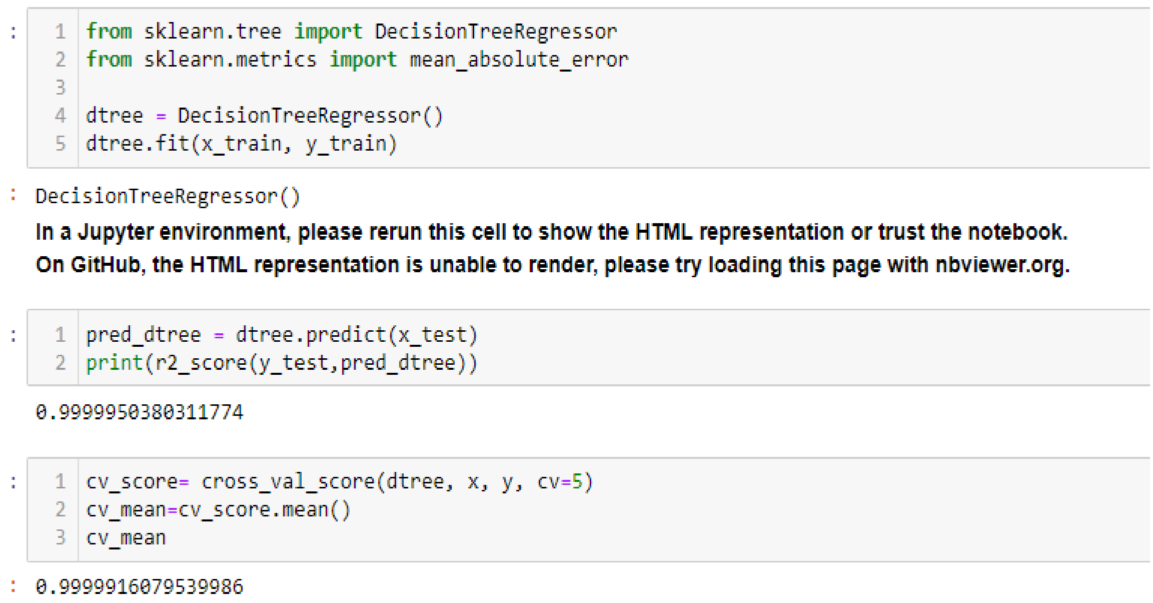
   Linear regression is an algorithm used to expect, or visualize, a relationship between two different features/variables. In linear regression tasks, there are two kinds of variables being examined: the dependent variable and the independent variable. After applying linear regression method, the output of r2 score is: 93.52% And cross validation score is: 93.28% The cross-validation score is used to find weather the model is having over fitting problem or not.

2. Random forest Regressor

   Random forest Regressor is a type of supervised machine learning algorithm. It gives the single result which is made with combination of multiple decision tree output. After applying Random Forest regression method, the output of r2 score is: 99.999% and cross validation score is: 99.998%

3. Decision Tree Regressor

   Decision tree uses the tree illustration to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. After applying Decision Tree regression method, the output of r2 score is: 99.999% and cross validation score is: 99.999%



```
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error

dtree = DecisionTreeRegressor()
dtree.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.

```
pred_dtree = dtree.predict(x_test)
print(r2_score(y_test,pred_dtree))
```

```
0.9999950380311774
```

```
cv_score= cross_val_score(dtree, x, y, cv=5)
cv_mean=cv_score.mean()
cv_mean
```

```
0.9999916079539986
```

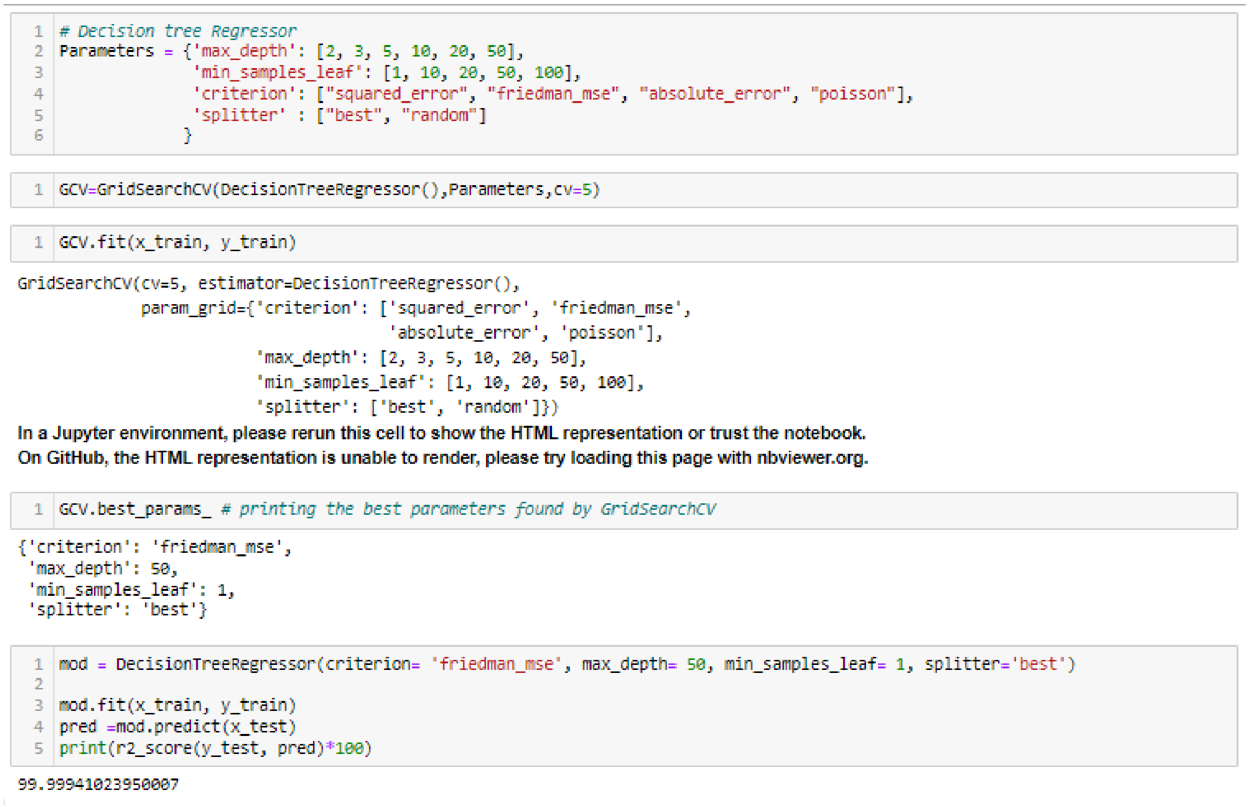
- Interpretation of the Results
- Comparison of all models:

Sr.No.	Algorithms	Accuracy score (a)	Cross-Validation Score (b)	Difference (a-b)
1.	Linear Regression	90.56%	90.14%	0.42%
2.	Random Forest Regressor	99.9997%	99.9993%	0.00032%
3.	Decision Tree Regressor	99.9995%	99.9993%	0.0003%

As shown in above table Decision Tree Regressor is having minimum difference, so Decision Tree Regressor is a best model. Now, I will work on best model with hyper parameter tuning.

Hyper Parameter tuning:

Grid search cross validation (GCV) in hyper parameter tuning will help us to finding the best hyper parameter and tune it with training data set and give best r2 score:



```
# Decision tree Regressor
Parameters = {'max_depth': [2, 3, 5, 10, 20, 50],
              'min_samples_leaf': [1, 10, 20, 50, 100],
              'criterion': ["squared_error", "friedman_mse", "absolute_error", "poisson"],
              'splitter' : ["best", "random"]
             }
```

```
GCV=GridSearchCV(DecisionTreeRegressor(),Parameters,cv=5)
```

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

```
GridSearchCV(cv=5, estimator=DecisionTreeRegressor(),
             param_grid={'criterion': ['squared_error', 'friedman_mse',
                                       'absolute_error', 'poisson'],
                         'max_depth': [2, 3, 5, 10, 20, 50],
                         'min_samples_leaf': [1, 10, 20, 50, 100],
                         'splitter': ['best', 'random']})
```

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.

```
GCV.best_params_ # printing the best parameters found by GridSearchCV
```

```
{'criterion': 'friedman_mse',
 'max_depth': 50,
 'min_samples_leaf': 1,
 'splitter': 'best'}
```

```
mod = DecisionTreeRegressor(criterion= 'friedman_mse', max_depth= 50, min_samples_leaf= 1, splitter='best')

mod.fit(x_train, y_train)
pred =mod.predict(x_test)
print(r2_score(y_test, pred)*100)
```

```
99.99941023950007
```

In above figure we can see the default hyper parameter which I was tuned with linear regression with GCV and then find the best parameters, now fit it in to model and find the r2 score:

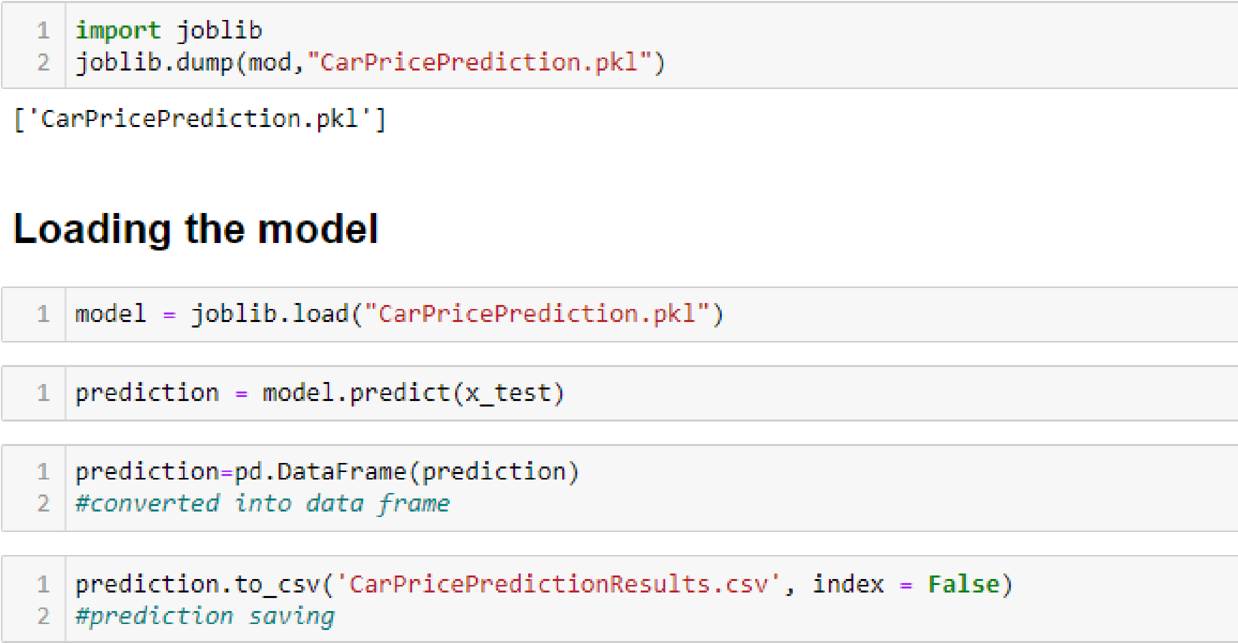
After hyper parameter tuning the r2 score is remains: 99.999%

Boosting of model: -

With the help of Gradient Boosting Regressor, I will boost my model and then my r2 score remain to: 99.999%

Saving and loading the model:

I save this model with name “CarPricePrediction.pkl”



```
import joblib
joblib.dump(mod,"CarPricePrediction.pkl")
```

```
['CarPricePrediction.pkl']
```

Loading the model

```
model = joblib.load("CarPricePrediction.pkl")
```

```
prediction = model.predict(x_test)
```

```
prediction=pd.DataFrame(prediction)
#converted into data frame
```

```
prediction.to_csv('CarPricePredictionResults.csv', index = False)
#prediction saving
```

CONCLUSION

- Key Findings and Conclusions of the Study

After the completion of this project, we got an insight on how to collect data, pre-processing the data, analyzing the data and building a model. First, we collected the used cars data from different websites like OLX, CarDekho, Cars 24, OLA etc and it was done by using Web Scraping. The framework used for web scraping was Beautiful Soup and Selenium, which has an advantage of automating our process of collecting data. We collected almost 5000+ of data which contained the selling price and other related features of used cars. Then the scrapped data was combined in a single data frame and saved in a csv file so that we can open it and analyze the data. We did data cleaning, data pre-processing steps like finding and handling null values, removing words from numbers, converting object to integer type, data visualization, handling outliers and skewness etc. After separating our train and test data, we started running different machine learning regression algorithms to find out the best performing model. We found that Extra Tree Regressor Algorithm was performing well according to their r2_score and cross validation scores. Then we performed hyper parameter Tuning technique using Grid Search CV for getting the best parameters and improving the score. In that Extra Tree Regressor Algorithm did not perform quite well as previously on the defaults but we finalized that model for further predictions as it was still better than the rest. We saved the final model in pkl format using the joblib library after getting a data frame of predicted and actual used car price details.

- Learning Outcomes of the Study in respect of Data Science

Visualization part helped me to understand the data as it provides graphical representation of huge data. It assisted me to understand the feature importance, outliers/skewness detection and to compare the independent-dependent features. Data cleaning is the most important part of model building and therefore before model building, we made sure the data is cleaned and scaled. We have generated multiple regression machine learning models to get the best model wherein we found Extra Trees Regressor Model being the best based on the metrics we have used.

- Limitations of this work and Scope for Future Work

The limitations we faced during this project were: The website was poorly designed because the scrapping took a lot of time and there were many issues in accessing to next page. Also need further practise in terms of various web scraping techniques. More negative correlated data were present than the positive correlated one's. Presence of outliers and skewness were detected and while dealing with them we had to lose a bit of valuable data. No information for handling these fast-paced websites were provided so that was consuming more time in web scraping part.

Future Work Scope: Current model is limited to used car data but this can further be improved for other sectors of automobiles by training the model accordingly. The overall score can also be improved further by training the model with more specific data.

REFERENCES:

1) https://www.google.com/
2) https://github.com/
3) https://www.kaggle.com/
4) https://towardsdatascience.com/
5) https://www.analyticsvidhya.com/