Project Name

CAR PRICE PREDICTION PROJECT REPORT

Submitted by:

Vikram Purohit

ACKNOWLEDGMENT

I would like to express my deep sense of gratitude to my SME (Subject Matter Expert) **Mr. Md. kashif** 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.

I have put in my all efforts while doing this project.

INTRODUCTION

• Business Problem Framing:

Impact of COVID-19 on Indian automotive sector: The Indian automotive sector was already struggling in FY20. before the Covid19 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 de-growth, 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 BSVI 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:

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 secondhand 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 a 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, kilometres 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 modelling 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, Car Dekho) .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 kilometres, fuel, 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 Visualisation 4. Model Building
5. Model Evaluation
6. Selecting the best model

Data Sources and their formats

The dataset is in the form of CSV (Comma Separated Value) format and consists of 10 columns (9 features and 1 label) with 5118 number of records as explained below:

- - Used Car Model - This shows the car model names
- - Year of Manufacture - Gives us the year in which the car was made
- - Kilometres Driven - Number of kilometres the car the driven reflecting on the Odometer
- - Fuel Type - Shows the fuel type used by the vehicle
- - Transmission Type - Gives us the manual or automatic gear shifting mechanism
- - Used Car Price - Lists the selling price of the used cars
  We can see our dataset includes a target label "Used 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.

Importing Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
df=pd.read_csv("car.csv")#load dataset
df.head()
```

	Unnamed: 0	Unnamed: 0.1	Brand	model	variant	year	fuel	transmission	km driven	No of owners	location	price
0	0	0	BMW	5 Series	523i	2010	Petrol	Automatic	69,000 km	2nd	Thane West, Thane, Maharashtra	₹ 10,25,000
1	1	1	Hyundai	Elite i20	Asta 1.2	2016	Petrol	Manual	27,884 km	-	Baner Road, Pune, Maharashtra	₹ 6,40,000
2	2	2	Renault	KWID	RXL	2016	Petrol	Manual	21,000 km	1st	Thillai Nagar, Tiruchirappalli, Tamil Nadu	₹ 3,00,000
3	3	3	Maruti Suzuki	Wagon R 1.0	1.0 LXi	2014	Petrol	Manual	8,350 km	1st	Govind Nagar, Nashik, Maharashtra	₹ 3,80,000
4	4	4	Mercedes-Benz	B Class	B180 Sport	2014	Petrol	Automatic	35,000 km	2nd	Adajan, Surat, Gujarat	₹ 12,50,000

Data Preprocessing Done

For the data pre-processing step, we checked through the dataframe for missing values, imputed records with “-“using various imputing techniques to handle them.

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

```
Brand           0
model           0
variant         0
year            0
fuel            0
transmission    0
km driven       0
No of owners    0
location        0
price           0
dtype: int64
```

Checked the datatype details for each column to understand the numeric ones and its further conversion process.

We also took a look at all the unique value present in each of the columns and then decided to deal with the imputation part accordingly.

The various data imputation performed on our data set are shown below with the code.

Data cleaning

modify Brand,model and variant column as its so much of noise

```
print(df['Brand'].nunique())
df['Brand'].value_counts()
```

```
2803

Maruti Suzuki              120
Hyundai                     95
Mahindra                    30
Toyota                      25
Volkswagen                  25
                          ...
2017 Mahindra TUV 300        1
2014 Nissan Terrano          1
2017 Hyundai Grand i10       1
2019 Hyundai i20             1
2011Nissan Micra XL          1
Name: Brand, Length: 2803, dtype: int64
```

```
#cleaning noise of Model column
Model=[]
for i in df['model']:
    i=i.split()
    if len(i)>=3:
        Model.append(i[1])
    else:
        Model.append(''.join(i))
len(Model)
```

```
4865
```

```
#cleaning unnecessary noise of variant column
Variant=[]
for i in df['variant']:
    i=i.split()
    if len(i)>=3:
        Variant.append(' '.join(i[2:]))
    else:
        Variant.append(' '.join(i))
Variant
```

```
['523i'
```

As there are so many locations,we name locations which are repeated less than 10 times we named them as others and other locations are as it is.

```
df['Location']=df['Location'].apply(lambda x:'others' if x in other else x)
```

```
df['Location'].unique()
```

```
array(['Thane', 'Tamil', 'others', 'Surat', 'Delhi', 'Ahmedabad',
       'Mumbai', 'Hyderabad', 'Pune', 'Bengaluru', 'chennai', 'kolkata'],
      dtype=object)
```

Converted categorical data to numeric by following steps,

converting catagorical feature to numeric

```
fuel=pd.get_dummies(df['fuel'],drop_first=True)
owners=pd.get_dummies(df['No of owners'],drop_first=True)
location=pd.get_dummies(df['Location'],drop_first=True)
brand=pd.get_dummies(df['Brand'],drop_first=True)
```

```
df=pd.concat([df,fuel,owners,location,brand],axis=1)
df=df.drop(['No of owners','fuel','Location','Brand'],axis=1)
df
```

Label Encoding

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
df['Model']=le.fit_transform(df['Model'])
df['Variant']=le.fit_transform(df['Variant'])
df['transmission']=le.fit_transform(df['transmission'])
df
```

Removed Outliers:

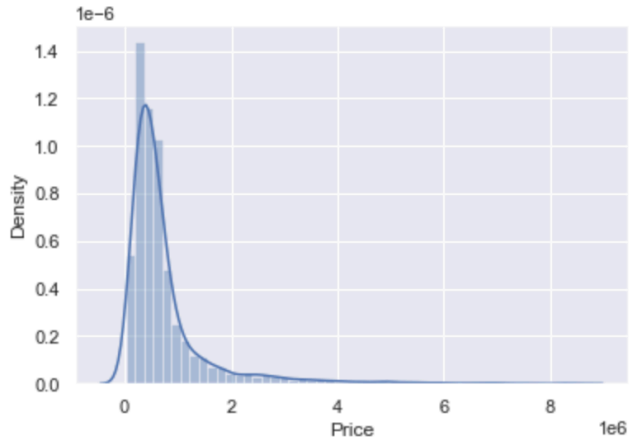
```
df=df[df['Price']<9000000]
df=df[df['Price']>20000]
```

```
df=df[df['Km Driven']<270000]
```

To handle the skewness, we made use of Log transformation technique ensuring that at least a bell shape curve closer to normal distribution is achieved.

```
sns.distplot(df['Price'])
```

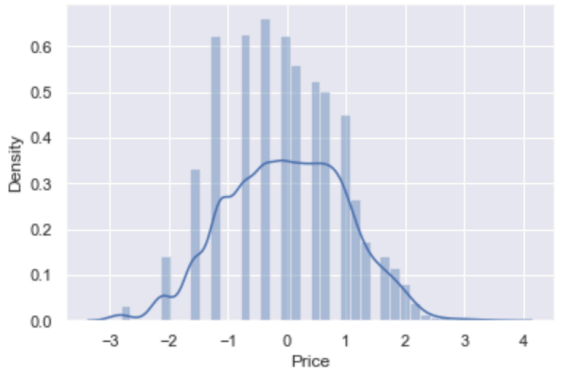
```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```




```
from sklearn.preprocessing import power_transform
df['Price']=power_transform(df,method='yeo-johnson')
```

```
sns.distplot(df['Price'])
```

```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```



Skewness removed

Hardware and Software Requirements and Tools Used

➢ *Hardware Used:*

- ▪ RAM: 8 GB

- ▪ CPU: AMD A8 Quad Core 2.2 Ghz▪GPU: AMD Redon R5 Graphics▪Software Tools used:

- ▪ Programming language: Python 3.0

- ▪ Distribution: Anaconda Navigator

- ▪ Browser-based language shell: Jupyter Notebook➢ *Libraries/Packages Used:*
  a. Pandas
  b. Numpy
  c. Matplotlib d. Seaborn
  e. Sklearn

Model/s Development and Evaluation
Identification of possible problem-solving approaches (methods)

1. Clean the dataset from unwanted scraped details.

2. Impute missing values with meaningful information.

3. Encoding the categorical data to get numerical input data.

4. Compare different models and identify the suitable model.

5. R2 score is used as the primary evaluation metric.

6. MSE and RMSE are used as secondary metrics.

7. Cross Validation Score was used to ensure there are no overfitting our under-fitting models.

Testing of Identified Approaches (Algorithms)

Libraries and Machine Learning Regression models that were used in this project are shown below.

Model building

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
```

```
#choosing best random_state
maxacc=0
maxrs=0
lr=LinearRegression()
for i  in range(1,100):
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1,random_state=i)
    lr.fit(x_train,y_train)
    y_pred=lr.predict(x_test)
    acc=r2_score(y_test,y_pred)
    if acc>maxacc:
        maxacc=acc
        maxrs=i
print('best accuracy score is',maxacc,'on random_state',maxrs)
```

```
best accuracy score is 0.9816934042928355 on random_state 45
```

All the regression machine learning algorithms used are:

- Ridge Regularization Model
- Lasso Regularization Model
- Elastic Net
- Decision Tree Regression Model
- Random Forest Regression Model
- Gradient Boosting Regression Model

- Run and Evaluate selected models
  We created a Regression Machine Leaning Model function incorporating the evaluation metrics so that we can get the required data for all the above models.

Code:

```
#importing different algorithms
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge
from sklearn.linear_model import ElasticNet
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
```

```
ls=Lasso()
rd=Ridge()
els=ElasticNet()
dt=DecisionTreeRegressor()
gb=GradientBoostingRegressor()
```

```
model=[lr,ls,rd,els,dt,gb]
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.05,random_state=74)
for i in model:
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    print(i)
    print(r2_score(y_test,y_pred))
    print(mean_squared_error(y_test,y_pred))
    print(mean_absolute_error(y_test,y_pred))
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1,random_state=45)
x_train['Km Driven']=sc.fit_transform(x_train[['Km Driven']])
x_test['Km Driven']=sc.transform(x_test[['Km Driven']])
```

Train and testing at the best random state:

Output:

```
LinearRegression()
0.974630072509835
0.028859840203100794
0.11666541312738035
Lasso()
-1.820728752899292e-05
1.1375817165302315
0.8834851440217798
Ridge()
0.9746269522639875
0.028863389673101755
0.11668044128142505
ElasticNet()
0.5266631036860709
0.5384495952992903
0.6046658726577079
DecisionTreeRegressor()
1.0
1.2490229505515725e-29
2.395888700055276e-15
GradientBoostingRegressor()
0.9999999970160663
3.394406550391458e-09
3.745270178955343e-05
```

```
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
y_Pred=rf.predict(x_test)
acc=r2_score(y_test,y_pred)
acc
```

```
1.0
```

Random Forest has given the best score so would be checking it with hyper parameter tuning and cross validation score for overfitting or under-fitting concepts.

Key Metrics for success in solving problem under consideration

RMSE Score:

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.

R2 Score:
The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. It is pronounced as R squared and is also known as the coefficient of determination. It works by measuring the amount of variance in the predictions explained by the dataset.

Cross Validation Score:
Cross-validation is a statistical method used to estimate the skill of machine learning models. It is commonly used in applied machine learning to compare and select a model for a given predictive modelling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods. The k-fold cross validation is a procedure used to estimate the skill of the model on new data. There are common tactics that you can use to select the value of k for your dataset (I have used 5-fold validation in this project). There are commonly used variations on cross-validation such as stratified and repeated that are available in scikit-learn.

Hyper Parameter Tuning:
In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

Final model score after plugging in the best parameter values:

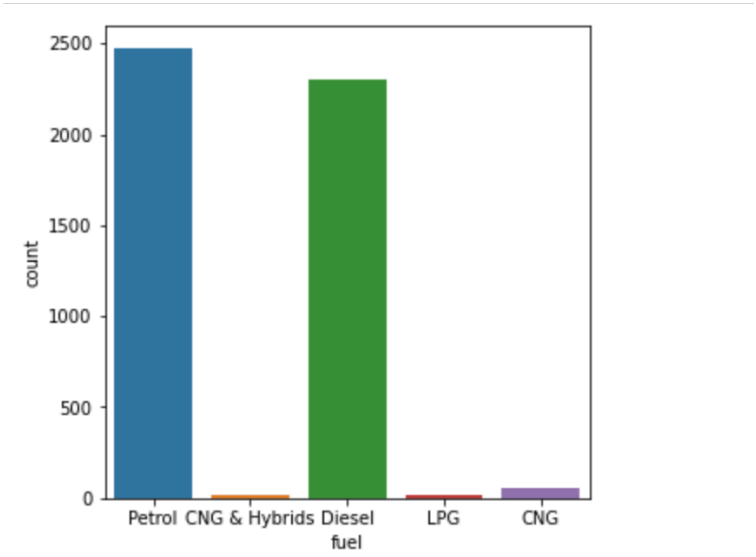
```
rf=RandomForestRegressor(n_estimators=200)
rf.fit(x_train,y_train)
rf.score(x_train,y_train)
y_pred=rf.predict(x_test)
print("Score of Model is",r2_score(y_test,y_pred))
print("Mean Absolute Error", mean_absolute_error(y_test,y_pred))
print("Root Mean Squared Error", (mean_squared_error(y_test,y_pred))**0.5 )
```

```
Score of Model is 0.9999941616755436
Mean Absolute Error 0.00019258282765056024
Root Mean Squared Error 0.0025771011299468276
```

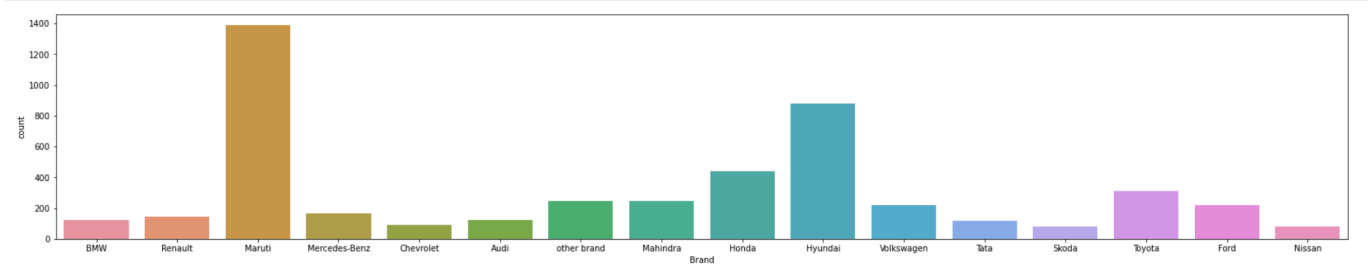
Hence random forest has given the best score and least error and no overfitting seen , best model to be saved.

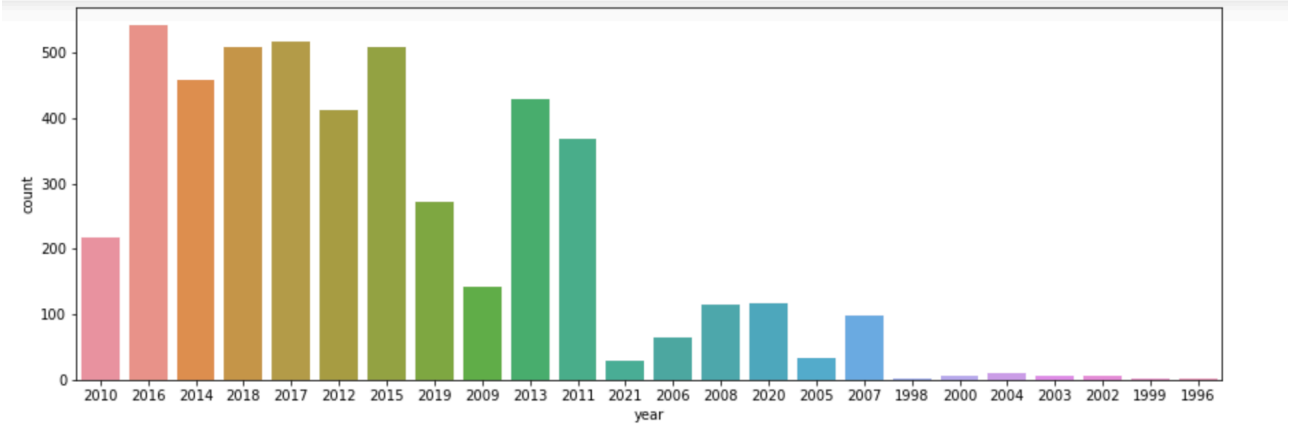
Visualizations

We generated count plots, bar plots, pair plots, heat-map and others to visualise the datapoint present in our column records.



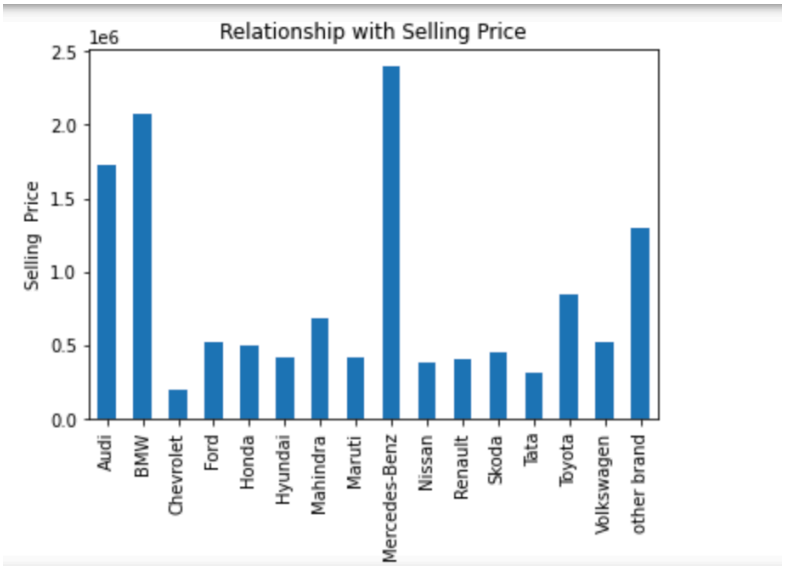
Diesel and Petrol cars are more in number.



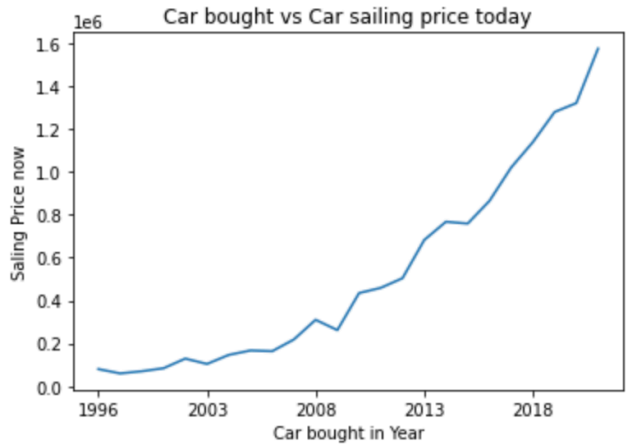


Maruti ,Hyundai and Honda cars are the highest in numbers for resale.

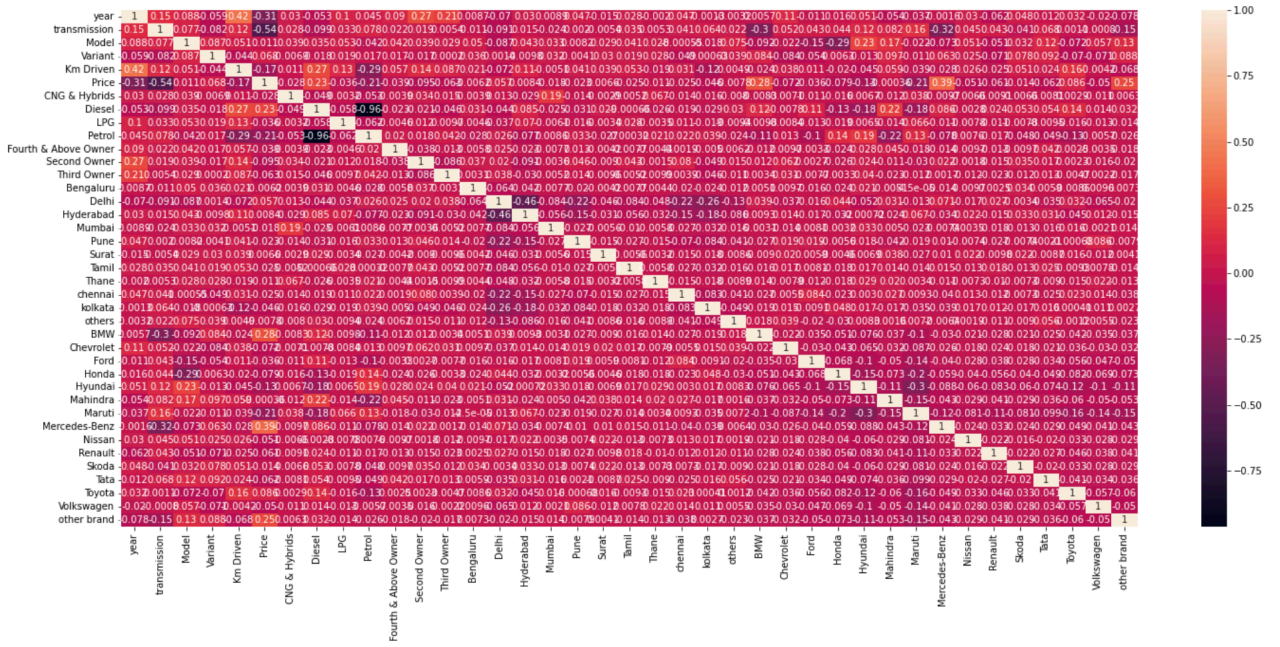
Selling price is dependent on vehicle age.



Selling price is high for luxury cars as new buying price is also high.



Selling price is promotional to buying year or age of vehicle.



Interpretation of the Results

We can see from the visuals that the features are impacting the price of used cars. There were categorical columns which we encoded using the ordinal encoder method instead of the one hot encoding to avoid the generation of large number of columns. Also, our target label stored continuous numeric data and therefore label encoder was out of the picture to be used.

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, analysing the data and building a model. First, we collected the used cars data from different websites like OLX, Car Dekho, 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 10000 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 analyse the data. We did data cleaning, data pre-processing steps like finding and handling null values, removing words from numbers, converting object to int 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 Hyperparameter 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 finalised 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 dataframe 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/