USED CAR PRICE PREDICTION

Submitted By: -

Prashant Pathak

Acknowledgement

First of all, I would like to thank my mentor Mr. Shwetank Mishra for giving me opportunity to work on this project.

For this project I needed data of used cars, So I went to Olx.com and collected used cars data from there using web scrapping techniques. I scraped the data in several parts. After that I made a big data set by taking small data sets using panda's techniques.

After that saved the dataset into csv format. After all these I cleaned the data one by one, analyzed it and prepared the prediction model.

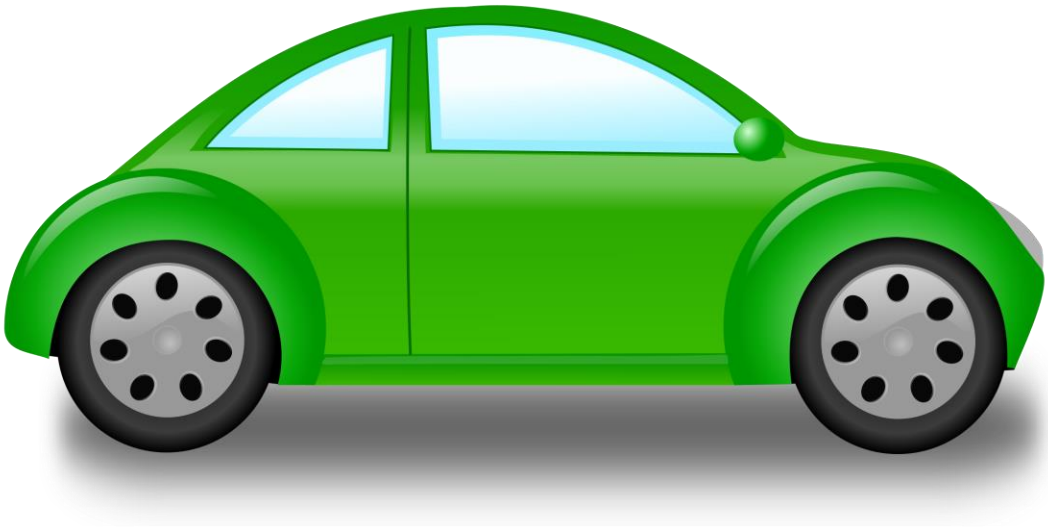
In all this my mentor supported me a lot I want to thank him again.

So now we'll understand all the process step by step:

Steps:

1. Problem Statement
2. Data Collection
3. Data Cleaning
4. Data Analysis
5. Preparer a best prediction model

Problem Statement:



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

Data Collection:

In this phase I went to the website of olx.com and started collecting data with the help of selenium library. Here I have collected the data in small parts.

In this data I give relevance to these following features:

Brand, Model, Variant, year, driven Km, fuel type, number of owners, location and price.

I have collected all this data from different areas of Uttar Pradesh.

In this way I collected data of more than 5000 cars.

Data collection links are below:

Small data collection file:

1. Used-Car-Price-Prediction-Project/Cars data WS 01 olx.ipynb at main · prashantpathakji/Used-Car-Price-Prediction-Project (github.com)
2. Used-Car-Price-Prediction-Project/Cars data WS 02 olx.ipynb at main · prashantpathakji/Used-Car-Price-Prediction-Project (github.com)
3. Used-Car-Price-Prediction-Project/Cars data WS 03 olx.ipynb at main · prashantpathakji/Used-Car-Price-Prediction-Project (github.com)
4. Used-Car-Price-Prediction-Project/Cars data WS 04 olx.ipynb at main · prashantpathakji/Used-Car-Price-Prediction-Project (github.com)

Making one large Data set using all small data sets:

1. Used-Car-Price-Prediction-Project/making large dataframe.ipynb at main · prashantpathakji/Used-Car-Price-Prediction-Project (github.com)

Data Cleaning:

In this phase I created a new file and imported the required libraries first and then I load the data and closely inspect each column.

I used these methods over here:

Shape, info (), unique (), is null () etc.

I got some result about my data from my observation.

Results:

(1). Unnamed: 0.1 and Unnamed: 0 are useless column.

(2). Manufacturing Year is float type column.

(3). Driven_KM and Price Columns are object type.

(4). Driven_KM column has many object with int.

(5). Price column has object with int.

(6). Fuel Type column has no issues.

(7). Brand column has no issues.

(8). Model with Variant column has unwanted Characters.

(9). Location column is not important because all cars from Up.

(10). No. of owner has wrong entry name.

After this I removed all the mistakes that I see in the data, using drop, replace, astype, isdigit, isnumeric etc. Methods.

After all these processes I had clean data. I saved again in csv format.

Data Analysis:

I did statistical and visual analysis of the data, and then I got some information about the data:

Name of data Set - car

Rows in data – 5285 (after cleaning)

Columns in data – 7 (after cleaning)

Number of categorical columns – 4

Number of numerical columns – 3

Name of targeted column - “Price”

Problem type – Regression (Supervised ML)

Statistical Report:

Statiscis Analysis:

```
In [32]: 1 car.describe()
```

Out[32]:

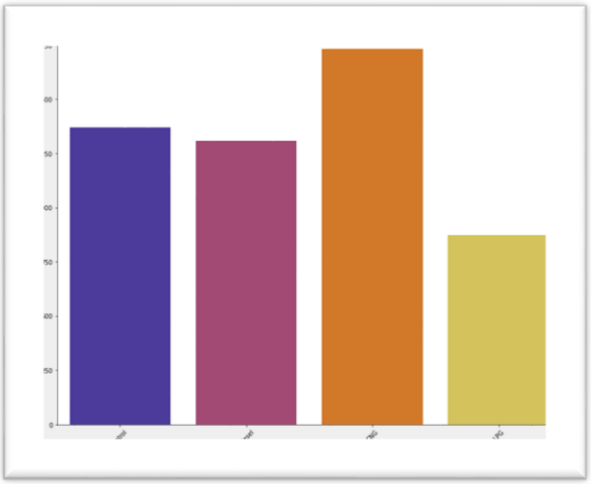
	Manufacturing Year	Driven KM	Price
count	5285.000000	5.285000e+03	5.285000e+03
mean	2012.575024	2.562246e+05	4.940251e+05
std	5.049293	4.021121e+05	7.544678e+05
min	1918.000000	0.000000e+00	1.500000e+04
25%	2009.000000	6.200000e+04	1.700000e+05
50%	2013.000000	9.000000e+04	2.900000e+05
75%	2016.000000	3.160000e+05	5.000000e+05
max	2023.000000	9.000000e+06	1.200000e+07

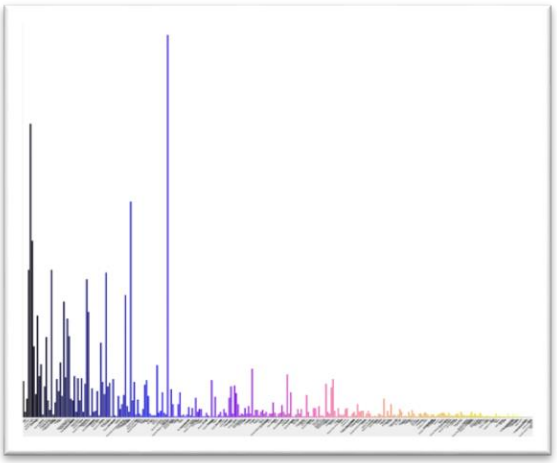
```
In [33]:  1  car.corr()
```

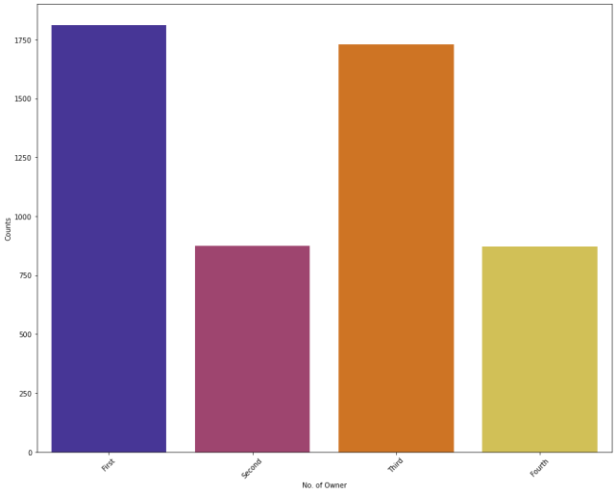
Out[33]:

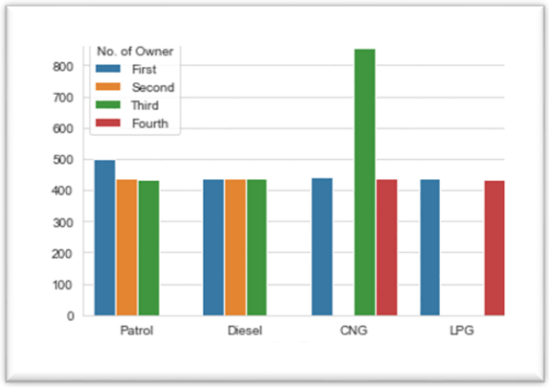
	Manufacturing Year	Driven KM	Price
Manufacturing Year	1.000000	-0.028727	0.351821
Driven KM	-0.028727	1.000000	0.041836
Price	0.351821	0.041836	1.000000

Some visualization Report:

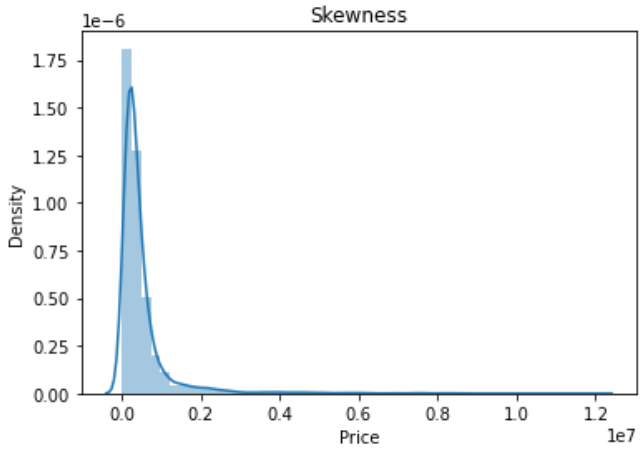


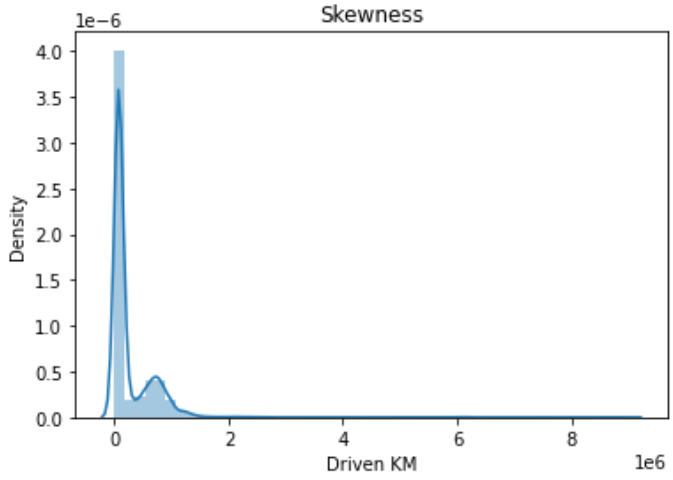


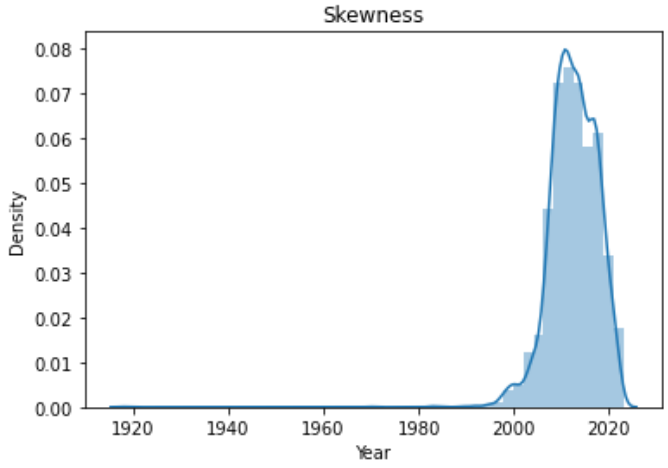




Skewness:


Skewness
1e−6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Price
1e7


Skewness
1e−6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Density
0
2
4
6
8
Driven KM
1e6


Skewness
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Density
1920
1940
1960
1980
2000
2020
Year

Result of visualization:

1. Maruti is bestselling car in the market.
2. The relevance of CNG cars is highest in the market.
3. Maruti Suzuki Omni, and Maruti Suzuki Wagon-R are Most selling car in the market.
4. First hand cars are most relevant according to market.
5. Year column is Left skewed.
6. Driven Km column is Right Skewed.
7. Price column is also Right Skewed.
8. CNG car by Third owner is most selling car.

Preparer a best prediction model:

I split the data into two parts x and y, and after this I import required modules for making a good predictor.

```
x=car.drop(columns='Price')
y=car["Price"]
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
```

```
ohe=OneHotEncoder()
ohe.fit(x[['Brand','Model with Variant','Fuel Type','No. of Owner']])
```

```
OneHotEncoder()
```

After this I built a pipeline, and fit the train and test data into it. And then I checked r-2 score on multiple random states.

Finally, 445 random state gave good result, and this is where I saved my model as a predictor using pickle.

In [62]:
```
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=np.argmax(scores))

    lr=LinearRegression()
    pipe=make_pipeline(column_trans,lr)
    pipe.fit(x_train,y_train)
    y_pred=pipe.predict(x_test)
    r2_score(y_test,y_pred)
```

Out[62]: 0.783304390142962

In [63]:
```
import pickle
```

In [64]:
```
pickle.dump(pipe,open('LinearRegressionModel.pkl','wb'))
```

So, this is the whole journey of this project if you want whole code of this project then follow this GitHub link;

[Used-Car-Price-Prediction-Project/Used Cars Price Predictions Project.ipynb at main · prashantpathakji/Used-Car-Price-Prediction-Project (github.com)]