

```
In [3]: import pandas as pd
import numpy as np
```

```
In [8]: d=pd.read_csv('https://github.com/YBI-Foundation/Dataset/raw/main/Car%20Price.csv')
```

```
In [9]: d.head()
```

Out[9]:

	Brand	Model	Year	Selling_Price	KM_Driven	Fuel	Seller_Type	Transmission	Owner
0	Maruti	Maruti 800 AC	2007	60000	70000	Petrol	Individual	Manual	First Owner
1	Maruti	Maruti Wagon R LXI Minor	2007	135000	50000	Petrol	Individual	Manual	First Owner
2	Hyundai	Hyundai Verna 1.6 SX	2012	600000	100000	Diesel	Individual	Manual	First Owner
3	Datsun	Datsun RediGO T Option	2017	250000	46000	Petrol	Individual	Manual	First Owner
4	Honda	Honda Amaze VX i-DTEC	2014	450000	141000	Diesel	Individual	Manual	Second Owner

```
In [10]: d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4340 entries, 0 to 4339
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Brand           4340 non-null   object
1   Model           4340 non-null   object
2   Year            4340 non-null   int64
3   Selling_Price   4340 non-null   int64
4   KM_Driven       4340 non-null   int64
5   Fuel            4340 non-null   object
6   Seller_Type     4340 non-null   object
7   Transmission    4340 non-null   object
8   Owner           4340 non-null   object
dtypes: int64(3), object(6)
memory usage: 305.3+ KB
```

```
In [11]: d.describe()
```

```
Out[11]:
```

	Year	Selling_Price	KM_Driven
count	4340.000000	4.340000e+03	4340.000000
mean	2013.090783	5.041273e+05	66215.777419
std	4.215344	5.785487e+05	46644.102194
min	1992.000000	2.000000e+04	1.000000
25%	2011.000000	2.087498e+05	35000.000000
50%	2014.000000	3.500000e+05	60000.000000
75%	2016.000000	6.000000e+05	90000.000000
max	2020.000000	8.900000e+06	806599.000000

```
In [12]: d[['Brand']].value_counts()
```

```
Out[12]:
```

Brand	
Maruti	1280
Hyundai	821
Mahindra	365
Tata	361
Honda	252
Ford	238
Toyota	206
Chevrolet	188
Renault	146
Volkswagen	107
Skoda	68
Nissan	64
Audi	60
BMW	39
Fiat	37
Datsun	37
Mercedes-Benz	35
Mitsubishi	6
Jaguar	6
Land	5
Ambassador	4
Volvo	4
Jeep	3
OpelCorsa	2
MG	2
Isuzu	1
Force	1
Daewoo	1
Kia	1

dtype: int64

```
In [13]: d[['Fuel']].value_counts()
```

```
Out[13]: Fuel
Diesel      2153
Petrol      2123
CNG         40
LPG         23
Electric     1
dtype: int64
```

```
In [15]: d[['Seller_Type']].value_counts()
```

```
Out[15]: Seller_Type
Individual      3244
Dealer          994
Trustmark Dealer 102
dtype: int64
```

```
In [16]: d[['Transmission']].value_counts()
```

```
Out[16]: Transmission
Manual      3892
Automatic   448
dtype: int64
```

```
In [17]: d[['Owner']].value_counts()
```

```
Out[17]: Owner
First Owner      2832
Second Owner     1106
Third Owner      304
Fourth & Above Owner 81
Test Drive Car   17
dtype: int64
```

```
In [19]: #d[['Fuel', 'Seller_Type', 'Transmission', 'Owner']].value_counts()
```

```
In [20]: d[['Model']].value_counts()
```

```
Out[20]: Model
Maruti Swift Dzire VDI      69
Maruti Alto 800 LXI         59
Maruti Alto LXi             47
Hyundai EON Era Plus        35
Maruti Alto LX              35
..
Mahindra KUV 100 G80 K4 Plus 1
Mahindra KUV 100 mFALCON D75 K8 1
Mahindra KUV 100 mFALCON D75 K8 AW 1
Mahindra KUV 100 mFALCON G80 K2 Plus 1
Volvo XC60 D5 Inscription    1
Length: 1491, dtype: int64
```

```
In [21]: d.columns
```

```
Out[21]: Index(['Brand', 'Model', 'Year', 'Selling_Price', 'KM_Driven', 'Fuel',  
              'Seller_Type', 'Transmission', 'Owner'],  
              dtype='object')
```

```
In [22]: d.shape
```

```
Out[22]: (4340, 9)
```

Encoding of Categorical Features

```
In [23]: d.replace({'Fuel':{'Petrol':0,'Diesel':1,'CNG':2,'LPG':3,'Electric':4}},inplace=True)
```

```
In [24]: d.replace({'Seller_Type':{'Individual':0,'Dealer':1,'Trustmark Dealer':2}},inplace=True)
```

```
In [25]: d.replace({'Transmission':{'Manual':0,'Automatic':1}},inplace=True)
```

```
In [58]: d.replace({'Owner':{'First Owner':0,'Second Owner':1,'Third Owner':2,'Fourth & Above':3}},inplace=True)
```

Define y and x variable

```
In [59]: y=d['Selling_Price']
```

```
In [60]: y.shape
```

```
Out[60]: (4340,)
```

```
In [61]: y
```

```
Out[61]: 0      60000  
         1     135000  
         2     600000  
         3     250000  
         4     450000  
         ...  
        4335    409999  
        4336    409999  
        4337    110000  
        4338     865000  
        4339     225000  
        Name: Selling_Price, Length: 4340, dtype: int64
```

```
In [62]: x=d[['Year','KM_Driven','Fuel','Seller_Type','Transmission','Owner']]
```

In [63]: `x.shape`

Out[63]: (4340, 6)

In [64]: `x`

Out[64]:

	Year	KM_Driven	Fuel	Seller_Type	Transmission	Owner
0	2007	70000	0	0	0	0
1	2007	50000	0	0	0	0
2	2012	100000	1	0	0	0
3	2017	46000	0	0	0	0
4	2014	141000	1	0	0	1
...
4335	2014	80000	1	0	0	1
4336	2014	80000	1	0	0	1
4337	2009	83000	0	0	0	1
4338	2016	90000	1	0	0	0
4339	2016	40000	0	0	0	0

4340 rows × 6 columns

Get Train Test Split

In [65]: `!pip install sklearn`

Requirement already satisfied: sklearn in c:\users\g.ravi prakash reddy\anaconda3\lib\site-packages (0.0)
 Requirement already satisfied: scikit-learn in c:\users\g.ravi prakash reddy\anaconda3\lib\site-packages (from sklearn) (1.1.0)
 Requirement already satisfied: numpy>=1.17.3 in c:\users\g.ravi prakash reddy\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.20.3)
 Requirement already satisfied: joblib>=1.0.0 in c:\users\g.ravi prakash reddy\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.1.0)
 Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\g.ravi prakash reddy\anaconda3\lib\site-packages (from scikit-learn->sklearn) (2.2.0)
 Requirement already satisfied: scipy>=1.3.2 in c:\users\g.ravi prakash reddy\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.7.1)

In [66]: `from sklearn.model_selection import train_test_split`

In [67]: `x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=252)`

```
In [68]: x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
Out[68]: ((3038, 6), (1302, 6), (3038,), (1302,))
```

Get Model Train

```
In [69]: from sklearn.linear_model import LinearRegression
```

```
In [70]: l=LinearRegression()  
l.fit(x_train,y_train)
```

```
Out[70]: 

▼ LinearRegression



LinearRegression()


```

Get Model Prediction

```
In [72]: y_pred=l.predict(x_test)
```

```
In [73]: y_pred.shape
```

```
Out[73]: (1302,)
```

```
In [74]: y_pred
```

```
Out[74]: array([502458.82786413, 646333.17428704, 521962.74075836, ...,  
                620183.32683781, 315403.8278857 , 731862.54196037])
```

Get Model Evaluation

```
In [75]: from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
```

```
In [76]: mean_squared_error(y_test,y_pred)
```

```
Out[76]: 193242972302.1957
```

```
In [77]: mean_absolute_error(y_test,y_pred)
```

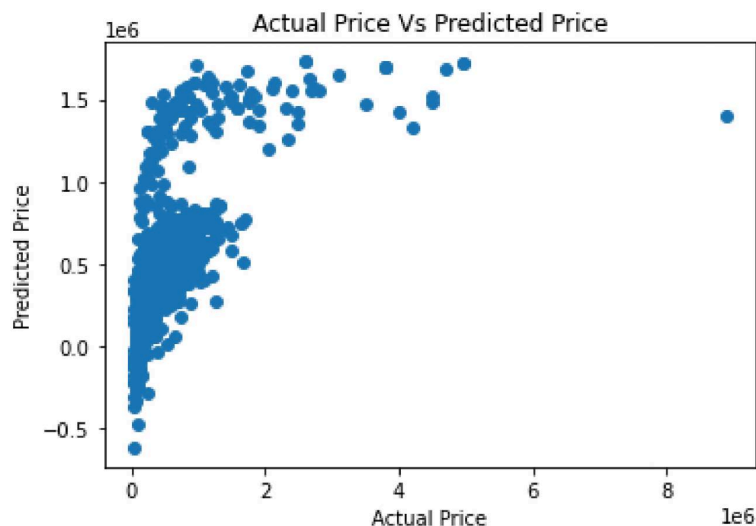
```
Out[77]: 228808.95522977927
```

```
In [78]: r2_score(y_test,y_pred)
```

```
Out[78]: 0.40755633943708336
```

Visualisation of Actual Vs Predicted Results

```
In [79]: import matplotlib.pyplot as p
p.scatter(y_test,y_pred)
p.xlabel('Actual Price')
p.ylabel('Predicted Price')
p.title("Actual Price Vs Predicted Price")
p.show()
```



Get Future Predictions

```
In [83]: d_new=d.sample(1)
```

```
In [84]: d_new
```

```
Out[84]:
```

	Brand	Model	Year	Selling_Price	KM_Driven	Fuel	Seller_Type	Transmission	Owner
3720	Hyundai	Hyundai i20 Active 1.2 S	2018	650000	30000	0	0	0	0

```
In [87]: d_new.shape
```

```
Out[87]: (1, 9)
```

```
In [89]: x_new=d_new.drop(['Brand','Model','Selling_Price'],axis=1)
```

```
In [90]: y_pred_new=l.predict(x_new)
```

```
In [91]: y_pred_new
```

```
Out[91]: array([495812.89330915])
```

In []: