```
[1]: import pandas as pd
     import numpy as np
```

```
[2]: df=pd.read_csv("https://raw.githubusercontent.com/YBI-Foundation/Dataset/main/
     ↪Car%20Price.csv")
```

```
[3]: df.head()
```

[3]:

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

	Seller_Type	Transmission	Owner
0	Individual	Manual	First Owner
1	Individual	Manual	First Owner
2	Individual	Manual	First Owner
3	Individual	Manual	First Owner
4	Individual	Manual	Second Owner

```
[4]: df.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

[5]: `df.describe()`

[5]:
	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

[6]: `df[["Brand"]].value_counts()`

[6]:
```
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

[7]: `df[['Model']].value_counts()`

[7]: 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

[8]: `df[['Fuel']].value_counts()`

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

[9]: `df[['Seller_Type']].value_counts()`

[9] : Seller_Type
       Individual   3244
       Dealer   994
       Trustmark Dealer   102
       dtype: int64

[10]: `df[['Transmission']].value_counts()`

[10] : Transmission
        Manual   3892
        Automatic   448
        dtype: int64

[11]: `df[['Owner']].value_counts()`

[11] : Owner
        First Owner   2832

```
            Second Owner          1106
            Third Owner            304
            Fourth & Above Owner    81
            Test Drive Car          17
            dtype: int64
```

[12]: `df.columns`

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

[13]: `df.shape`

[13]: (4340, 9)

[14]: `df.replace({"Fuel":{"Petrol":0,"Diesel":1,"CNG":2,"LPG":3,"Electric":`
      `     ↪4}},inplace=True)`

[15]: `df.replace({"Seller_Type":{"Individual":0,"Dealer":1,"Trustmark Dealer":`
      `     ↪2}},inplace=True)`

[16]: `df.replace({"Transmission":{"Manual":0,"Automatic":1}},inplace=True)`

[17]: `df.replace({"Owner":{"First Owner":0,"Second Owner":1,"Third Owner":2,"Fourth &`
      `     ↪Above Owner":3,"Test Drive Car":4}},inplace=True)`

[18]: `y=df["Selling_Price"]`

[19]: `y.shape`

[19]: (4340,)

[20]: `y`

```
[20]: 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
```

4

```
[21]: x=df[["Year","KM_Driven","Fuel","Seller_Type","Transmission","Owner"]]
```

```
[22]: x.shape
```

```
[22]: (4340, 6)
```

```
[23]: x
```

```
[23]:        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 x 6 columns]
```

```
[24]: from sklearn.model_selection import train_test_split
```

```
[25]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
      ↪3,random_state=2529)
```

```
[26]: x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
[26]: ((3038, 6), (1302, 6), (3038,), (1302,))
```

```
[27]: from sklearn.linear_model import LinearRegression
```

```
[28]: lr=LinearRegression()
```

```
[29]: lr.fit(x_train,y_train)
```

```
[29]: LinearRegression()
```

```
[30]: y_pred=lr.predict(x_test)
```

```
[31]: y_pred.shape
```

```
[31]: (1302,)
```

```
[32]: y_pred
```

5

```
[32]:  array([502458.82786413, 646333.17428704, 521962.74075836, ...,
              620183.32683781, 315403.82788569, 731862.54196037])
```

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

```
[34]:  mean_squared_error(y_test,y_pred)
```
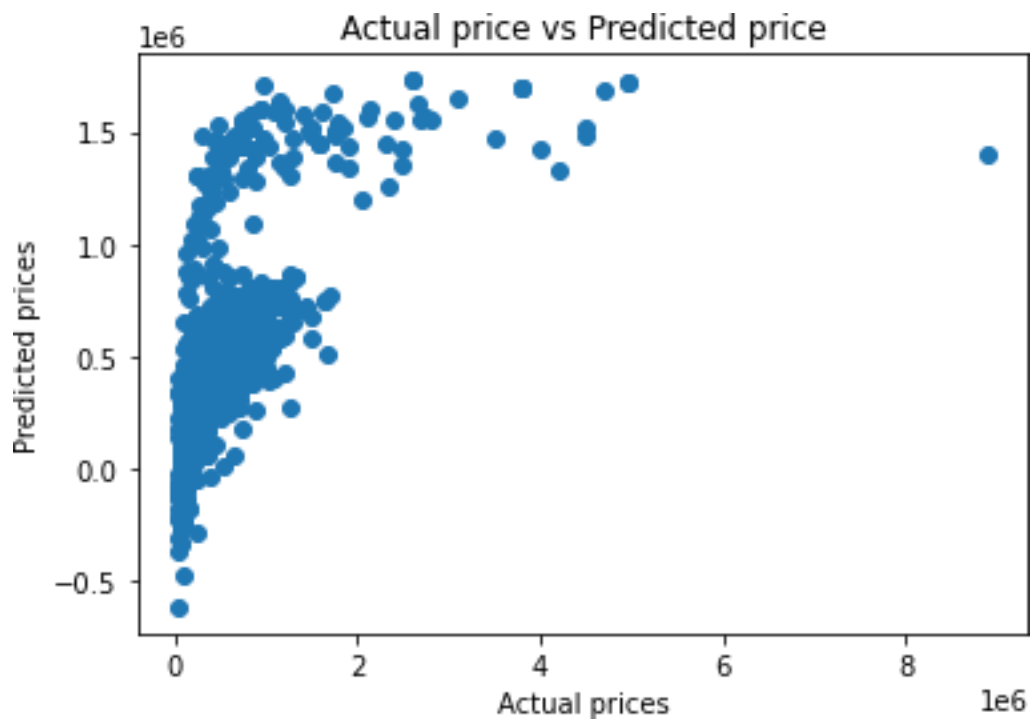
[34]: 193242972302.19577

```
[35]:  mean_absolute_error(y_test,y_pred)
```

[35]: 228808.95522977936

```
[36]:  r2_score(y_test,y_pred)
```

[36]: 0.40755633943708325

```
[37]:  import matplotlib.pyplot as plt
       plt.scatter(y_test,y_pred)
       plt.xlabel("Actual prices")
       plt.ylabel("Predicted prices")
       plt.title("Actual price vs Predicted price")
       plt.show()
```

```python
[38]: df_new=df.sample(1)
```

```python
[39]: df_new
```

```
[39]:         Brand               Model  Year  Selling_Price  KM_Driven  Fuel  \
      425  Maruti  Maruti Wagon R VXI  2004         130000      46000     0

          Seller_Type  Transmission  Owner
      425            0             0      0
```

```python
[40]: df_new.shape
```

```
[40]: (1, 9)
```

```python
[41]: x_new=df_new.drop(["Brand","Model","Selling_Price"],axis=1)
```

```python
[42]: y_pred_new=lr.predict(x_new)
```

```python
[43]: y_pred_new
```

```
[43]: array([-67004.63380739])
```

```python
[ ]:
```