

```
In [1]: import pandas as pd

In [2]: import numpy as np

In [4]: df = pd.read_csv(r'https://github.com/YBI-Foundation/Dataset/raw/main/Car%20Price.csv')

In [5]: df.head()

Out[5]:
   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 [6]: 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

In [7]: df = df.dropna()

In [8]: df.describe()

Out[8]:
      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 [9]: df[['Brand']].value_counts()

Out[9]:
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 [10]: df[['Model']].value_counts()

Out[10]:
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 [11]: df[['Fuel']].value_counts()

Out[11]:
Fuel
Diesel      2153
Petrol      2123
CNG         40
LPG         23
Electric     1
dtype: int64

In [12]: df[['Seller_Type']].value_counts()

Out[12]:
Seller_Type
Individual      3244
Dealer          994
Trustmark Dealer 182
dtype: int64

In [13]: df[['Transmission']].value_counts()

Out[13]:
Transmission
Manual      3892
Automatic   448
dtype: int64

In [14]: df[['Owner']].value_counts()

Out[14]:
Owner
First Owner      2832
Second Owner     1106
Third Owner       384
Fourth & Above Owner 81
Test Drive Car    17
dtype: int64

In [15]: #df[['Fuel', 'Seller_Type', 'Transmission', 'Owner']].value_counts()

In [16]: df.columns

Out[16]:
Index(['Brand', 'Model', 'Year', 'Selling_Price', 'KM_Driven', 'Fuel',
      'Seller_Type', 'Transmission', 'Owner'],
      dtype='object')

In [17]: df.shape

Out[17]:
(4340, 9)

In [18]: df.replace({'Fuel':{'Petrol':0, 'Diesel':1, 'CNG':2, 'LPG': 3, 'Electric':4}},inplace=True)

In [19]: df.replace({'Seller_Type':{'Individual':0, 'Dealer':1, 'Trustmark Dealer':2}},inplace=True)

In [20]: df.replace({'Transmission':{'Manual':0, 'Automatic':1}},inplace=True)

In [22]: df.replace({'Owner':{'First Owner':0, 'Second Owner':1, 'Third Owner': 2, 'Fourth & Above Owner':3, 'Test Drive Car':4}},inplace=True)

In [23]: #x = pd.get_dummies (x, columns=['Fuel', 'Seller_Type', 'Transmission', 'Owner'], drop_first=True)

In [24]: y =df['Selling_Price']

In [25]: y.shape

Out[25]:
(4340,)

In [26]: y

Out[26]:
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 [29]: x = df[['Year', 'KM_Driven', 'Fuel', 'Seller_Type', 'Transmission', 'Owner']]

In [30]: x.shape

Out[30]:
(4340, 6)

In [31]: x

Out[31]:
   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

In [32]: from sklearn.model_selection import train_test_split

In [33]: x_train,x_test,y_train,y_test = train_test_split(x,y, test_size = 0.3, random_state=72529)

In [34]: x_train.shape,x_test.shape,y_train.shape,y_test.shape

Out[34]:
((3038, 6), (1302, 6), (3038,), (1302,))

In [35]: from sklearn.linear_model import LinearRegression

In [36]: lr = LinearRegression()

In [37]: lr.fit(x_train,y_train)

Out[37]:
LinearRegression
LinearRegression()

In [38]: y_pred = lr.predict(x_test)

In [39]: y_pred.shape

Out[39]:
(1302,)

In [40]: y_pred

Out[40]:
array([ 742588.23909873, -83568.1480097 ,  85231.88045487, ...,
        407530.24057807, 1442317.08662422,  284146.56288867])

In [41]: from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

In [42]: mean_squared_error(y_test,y_pred)

Out[42]:
219430033060.6229

In [43]: mean_absolute_error(y_test,y_pred)

Out[43]:
237679.2980671438

In [44]: r2_score(y_test,y_pred)

Out[44]:
0.4161217261989766

In [45]: import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual Prices')
plt.ylabel('Prediction Prices')
plt.title("Actual Prices vs Prediction Prices")
plt.show()

Actual Prices vs Prediction Prices

Prediction Prices
1e6
15
10
0.5
0.0
0
2
4
6
8
1e6
Actual Prices

In [46]: df_new = df.sample(1)

In [47]: df_new

Out[47]:
   Brand      Model  Year  Selling_Price  KM_Driven  Fuel  Seller_Type  Transmission  Owner
175  Maruti  Maruti 800 AC   2007         95000     100000     0         0         0         1

In [48]: df_new.shape

Out[48]:
(1, 9)

In [49]: x_new = df_new.drop(['Brand','Model','Selling_Price'], axis = 1)

In [50]: y_pred_new = lr.predict(x_new)

In [51]: y_pred_new

Out[51]:
array([18137.28551677])

In [ ]:
```