

SUMESH R -20104169

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [2]: df = pd.read_csv("12_mobile_prices_2023.csv")
df
```

Out[2]:

	Phone Name	Rating ?/5	Number of Ratings	RAM	ROM/Storage	Back/Rare Camera	Front Camera	Battery	Processor	Price in INR
0	POCO C50 (Royal Blue, 32 GB)	4.2	33,561	2 GB RAM	32 GB ROM	8MP Dual Camera	5MP Front Camera	5000 mAh	Mediatek Helio A22 Processor, Upto 2.0 GHz Pro...	₹5,649
1	POCO M4 5G (Cool Blue, 64 GB)	4.2	77,128	4 GB RAM	64 GB ROM	50MP + 2MP	8MP Front Camera	5000 mAh	Mediatek Dimensity 700 Processor	₹11,999
2	POCO C51 (Royal Blue, 64 GB)	4.3	15,175	4 GB RAM	64 GB ROM	8MP Dual Rear Camera	5MP Front Camera	5000 mAh	Helio G36 Processor	₹6,999
3	POCO C55 (Cool Blue, 64 GB)	4.2	22,621	4 GB RAM	64 GB ROM	50MP Dual Rear Camera	5MP Front Camera	5000 mAh	Mediatek Helio G85 Processor	₹7,749
4	POCO C51 (Power Black, 64 GB)	4.3	15,175	4 GB RAM	64 GB ROM	8MP Dual Rear Camera	5MP Front Camera	5000 mAh	Helio G36 Processor	₹6,999
...
1831	Infinix Note 7 (Forest Green, 64 GB)	4.3	25,582	4 GB RAM	64 GB ROM	48MP + 2MP + AI Lens Camera	16MP Front Camera	5000 mAh	MediaTek Helio G70 Processor	₹14,999
1832	Infinix Note 7 (Bolivia	4.3	25,582	4 GB RAM	64 GB ROM	48MP + 2MP + AI	16MP Front Camera	5000 mAh	MediaTek Helio G70 Processor	₹14,999

	Phone Name	Rating ?/5	Number of Ratings	RAM	ROM/Storage	Back/Rare Camera	Front Camera	Battery	Processor	Price in INR
	Blue, 64 GB)					Lens Camera				
1833	Infinix Note 7 (Aether Black, 64 GB)	4.3	25,582	4 GB RAM	64 GB ROM	48MP + 2MP + 2MP + AI Lens Camera	16MP Front Camera	5000 mAh	MediaTek Helio G70 Processor	₹14,999
1834	Infinix Zero 8i (Silver Diamond, 128 GB)	4.2	7,117	8 GB RAM	128 GB ROM	48MP + 8MP + 2MP + AI Lens Camera	16MP + 8MP Dual Front Camera	4500 mAh	MediaTek Helio G90T Processor	₹18,999
1835	Infinix S5 (Quetzal Cyan, 64 GB)	4.3	15,701	4 GB RAM	64 GB ROM	16MP + 5MP + 2MP + Low Light Sensor	32MP Front Camera	4000 mAh	Helio P22 (MTK6762) Processor	₹10,999

1836 rows × 11 columns

In [3]:

df.head()

Out[3]:

	Phone Name	Rating ?/5	Number of Ratings	RAM	ROM/Storage	Back/Rare Camera	Front Camera	Battery	Processor	Price in INR	Score
0	POCO C50 (Royal Blue, 32 GB)	4.2	33,561	2 GB RAM	32 GB ROM	8MP Dual Camera	5MP Front Camera	5000 mAh	Mediatek Helio A22 Processor, Upto 2.0 GHz Pro...	₹5,649	20
1	POCO M4 5G (Cool Blue, 64 GB)	4.2	77,128	4 GB RAM	64 GB ROM	50MP + 2MP	8MP Front Camera	5000 mAh	Mediatek Dimensity 700 Processor	₹11,999	20
2	POCO C51 (Royal Blue, 64 GB)	4.3	15,175	4 GB RAM	64 GB ROM	8MP Dual Rear Camera	5MP Front Camera	5000 mAh	Helio G36 Processor	₹6,999	20
3	POCO C55 (Cool Blue, 64 GB)	4.2	22,621	4 GB RAM	64 GB ROM	50MP Dual Rear Camera	5MP Front Camera	5000 mAh	Mediatek Helio G85 Processor	₹7,749	20
4	POCO C51 (Power	4.3	15,175	4 GB RAM	64 GB ROM	8MP Dual Rear Camera	5MP Front Camera	5000 mAh	Helio G36 Processor	₹6,999	20

Phone Name	Rating ?/5	Number of Ratings	RAM	ROM/Storage	Back/Rare Camera	Front Camera	Battery	Processor	Price in INR	Date of Scraping
------------	------------	-------------------	-----	-------------	------------------	--------------	---------	-----------	--------------	------------------

Data cleaning and pre processing

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1836 entries, 0 to 1835
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Phone Name            1836 non-null   object
1   Rating ?/5            1836 non-null   float64
2   Number of Ratings     1836 non-null   object
3   RAM                   1836 non-null   object
4   ROM/Storage           1662 non-null   object
5   Back/Rare Camera      1827 non-null   object
6   Front Camera          1435 non-null   object
7   Battery               1826 non-null   object
8   Processor             1781 non-null   object
9   Price in INR          1836 non-null   object
10  Date of Scraping      1836 non-null   object
dtypes: float64(1), object(10)
memory usage: 157.9+ KB
```

In [5]:

```
df.describe()
```

Out[5]:

	Rating ?/5
count	1836.000000
mean	4.210512
std	0.543912
min	0.000000
25%	4.200000
50%	4.300000
75%	4.400000
max	4.800000

In [6]:

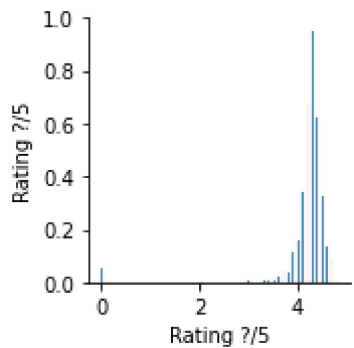
```
df.columns
```

```
Out[6]: Index(['Phone Name', 'Rating ?/5', 'Number of Ratings', 'RAM', 'ROM/Storage',
              'Back/Rare Camera', 'Front Camera', 'Battery', 'Processor',
              'Price in INR', 'Date of Scraping'],
              dtype='object')
```

EDA and VISUALIZATION

```
In [7]: sns.pairplot(df)
```

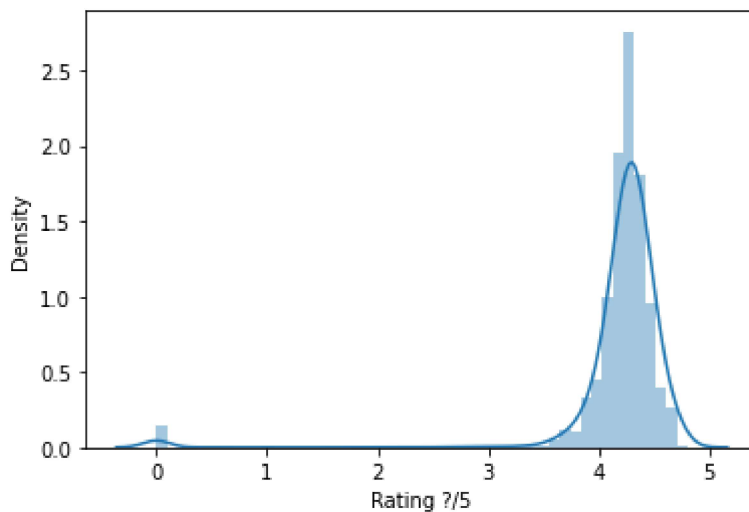
```
Out[7]: <seaborn.axisgrid.PairGrid at 0x2b127135c70>
```



```
In [8]: sns.distplot(df["Rating ?/5"])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

```
Out[8]: <AxesSubplot:xlabel='Rating ?/5', ylabel='Density'>
```



```
In [9]: df1 = df[['Phone Name', 'Rating ?/5', 'Number of Ratings', 'RAM', 'ROM/Storage',  
                'Back/Rare Camera', 'Front Camera', 'Battery', 'Processor',  
                'Price in INR', 'Date of Scraping']]
```

```
In [10]: sns.heatmap(df1.corr())
```

```
Out[10]: <AxesSubplot:>
```



```
In [11]: x = df1[['Rating ?/5','Rating ?/5']]
         y = df1['Rating ?/5']
```

split the data into training and test data

```
In [12]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)
```

```
In [13]: lr = LinearRegression()
         lr.fit(x_train, y_train)
```

Out[13]: LinearRegression()

```
In [14]: lr.intercept_
```

Out[14]: 8.881784197001252e-16

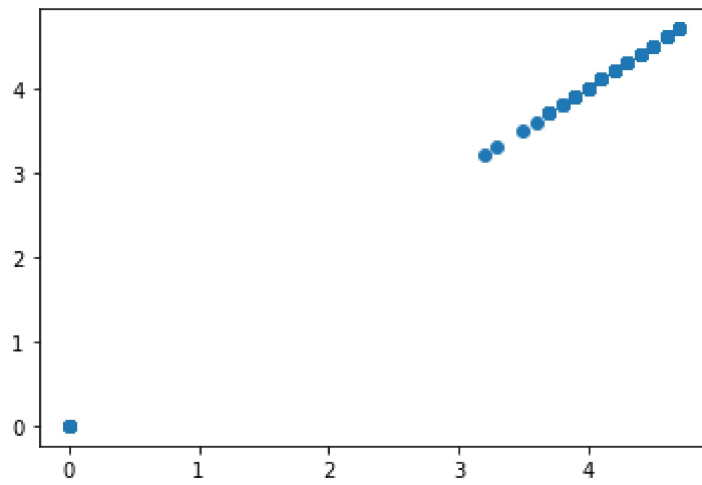
```
In [15]: coeff = pd.DataFrame(lr.coef_, x.columns, columns =['Co-efficient'])
         coeff
```

```
Out[15]:
```

	Co-efficient
Rating ?/5	0.5
Rating ?/5	0.5

```
In [16]: prediction = lr.predict(x_test)
         plt.scatter(y_test, prediction)
```

Out[16]: <matplotlib.collections.PathCollection at 0x2b127c3ff10>



```
In [17]: lr.score(x_test,y_test)
```

```
Out[17]: 1.0
```