Project Name:- Rating Prediction Project.

Importing Libraries

Loading Dataset

In [5]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
df=pd.read_csv("C:\\Users\ADMIN\Desktop\\varshu.csv")
print(df)
```

```
                                               Prod_Name Prod_Rating  \
0          Acer V196HQL 18.5 inch LED Backlit LCD Monitor         4.3
1      ZEBRONICS 18.5 inch HD TN Panel HDMI, VGA Inpu...         4.1
2      acer 53.8 inch Full HD LED Backlit IPS Panel M...           5
3      BenQ 53.8 inch Full HD LED Backlit IPS Panel F...           5
4      SAMSUNG 57 inch Full HD VA Panel with HAS,3-si...         4.4
...                                                  ...         ...
19994    Enter WIDE 17.3 inch HD Monitor (SXGA E MO A05)         5.1
19995  DELL E-SERIES 51.5 inch Full HD LED Backlit VA...         3.8
19996  LG 18.5 inch HD LED Backlit TN Panel Monitor (...         4.1
19997           Enter 17.3 inch HD Monitor (E-MO-A05(W))         3.5
19998             ZEBSTER 19 inch HD Monitor (ZEBRONICS)         3.8

      Prod_Review
0          16,059
1           3,155
2           1,493
3           4,566
4             105
...           ...
19994         174
19995           1
19996          59
19997           1
19998           0

[19999 rows x 3 columns]
```

In [6]:

```
df
```

Out[6]:

	Prod_Name	Prod_Rating	Prod_Review
0	Acer V196HQL 18.5 inch LED Backlit LCD Monitor	4.3	16,059
1	ZEBRONICS 18.5 inch HD TN Panel HDMI, VGA Inpu...	4.1	3,155
2	acer 53.8 inch Full HD LED Backlit IPS Panel M...	5	1,493
3	BenQ 53.8 inch Full HD LED Backlit IPS Panel F...	5	4,566
4	SAMSUNG 57 inch Full HD VA Panel with HAS,3-si...	4.4	105
...	...	...	...
19994	Enter WIDE 17.3 inch HD Monitor (SXGA E MO A05)	5.1	174
19995	DELL E-SERIES 51.5 inch Full HD LED Backlit VA...	3.8	1
19996	LG 18.5 inch HD LED Backlit TN Panel Monitor (...	4.1	59
19997	Enter 17.3 inch HD Monitor (E-MO-A05(W))	3.5	1
19998	ZEBSTER 19 inch HD Monitor (ZEBRONICS)	3.8	0

19999 rows × 3 columns

Checking the Additional_info column and having the count of unique types of values.

In [5]:

```
df["Prod_Rating"].value_counts()
```

Out[5]:

```
5      3251
4.1    2026
4.3    1753
4.5    1740
4      1737
4.4    1626
3.5    1474
3.9    1246
3.7    1009
3.8     924
3.6     784
3.3     505
3.4     503
3       298
5.9     184
3.1     184
5.5     152
5.3      81
4.6      76
5.8      71
1        64
4.7      63
5.6      52
5.4      27
1.5      26
5.7      26
1.9      22
1.8      21
4.9      20
4.8      12
5.1      10
1.4       9
38        6
0         3
8         3
4..5      2
431       2
2.8       1
Name: Prod_Rating, dtype: int64
```

Information about the dataset

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19999 entries, 0 to 19998
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Prod_Name    19993 non-null  object
 1   Prod_Rating  19993 non-null  object
 2   Prod_Review  19953 non-null  object
dtypes: object(3)
memory usage: 468.9+ KB
```

In [5]:

```
df.sample(5)
```

Out[5]:

	Prod_Name	Prod_Rating	Prod_Review
13017	GLWO Y68 Smart Watches for Men Women, Bluetoot...	4	0
16323	(Renewed) boAt Airdopes 455 Bluetooth Truly Wi...	3	138
2872	Samsung Galaxy M55 5G (ICY Blue, 8GB RAM, 158G...	4.5	15,548
9551	Redmi 7 (Comet Blue, 5GB RAM, SD 635, 35GB Sto...	4	76,467
16418	Apple iPhone 13 (515 GB) - Green	5	7,396

In [6]:

```
df.head(5)
```

Out[6]:

	Prod_Name	Prod_Rating	Prod_Review
0	Acer V196HQL 18.5 inch LED Backlit LCD Monitor	4.3	16,059
1	ZEBRONICS 18.5 inch HD TN Panel HDMI, VGA Inpu...	4.1	3,155
2	acer 53.8 inch Full HD LED Backlit IPS Panel M...	5	1,493
3	BenQ 53.8 inch Full HD LED Backlit IPS Panel F...	5	4,566
4	SAMSUNG 57 inch Full HD VA Panel with HAS,3-si...	4.4	105

In [6]:

```
df.tail(5)
```

Out[6]:

	Prod_Name	Prod_Rating	Prod_Review
19994	Enter WIDE 17.3 inch HD Monitor (SXGA E MO A05)	5.1	174
19995	DELL E-SERIES 51.5 inch Full HD LED Backlit VA...	3.8	1
19996	LG 18.5 inch HD LED Backlit TN Panel Monitor (...	4.1	59
19997	Enter 17.3 inch HD Monitor (E-MO-A05(W))	3.5	1
19998	ZEBSTER 19 inch HD Monitor (ZEBRONICS)	3.8	0

In [7]:

```
df.columns
```

Out[7]:

```
Index(['Prod_Name', 'Prod_Rating', 'Prod_Review'], dtype='object')
```

In [8]:

```
df.shape
```

Out[8]:

```
(19999, 3)
```

In [11]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19999 entries, 0 to 19998
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Prod_Name    19993 non-null  object
 1   Prod_Rating  19993 non-null  object
 2   Prod_Review  19953 non-null  object
dtypes: object(3)
memory usage: 468.9+ KB
```

Now while using the IsNull function and sum function we will gonna see the number of null values in our dataset

In [10]:

```
df.isnull()
```

Out[10]:

	Prod_Name	Prod_Rating	Prod_Review
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
...	...	...	...
19994	False	False	False
19995	False	False	False
19996	False	False	False
19997	False	False	False
19998	False	False	False

19999 rows × 3 columns

Now while using the IsNull function and sum function we will gonna see the number of null values in our dataset

In [11]:

```
df.isnull().sum()
```

Out[11]:

```
Prod_Name       6
Prod_Rating     6
Prod_Review    46
dtype: int64
```

In [12]:

```
df.isnull().sum().sum()
```

Out[12]:

```
58
```

Now while using the IsNull function we will gonna see the number of null values in our dataset

In [8]:

```
df.isnull().head()
```

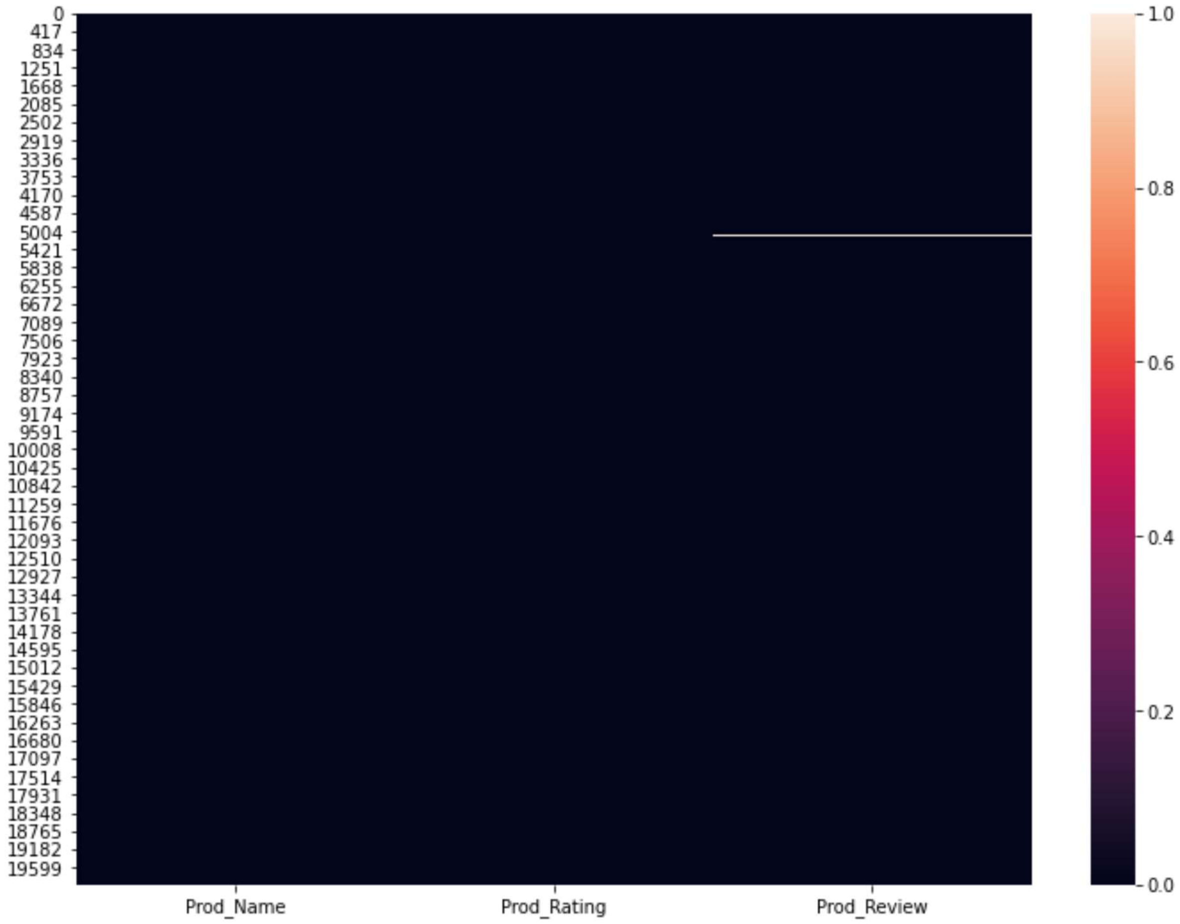
Out[8]:

	Prod_Name	Prod_Rating	Prod_Review
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False

Isnull Heatmap

In [14]:

```
fig, ax = plt.subplots(figsize=(12,9))
sns.heatmap(df.isnull(), ax=ax);
```



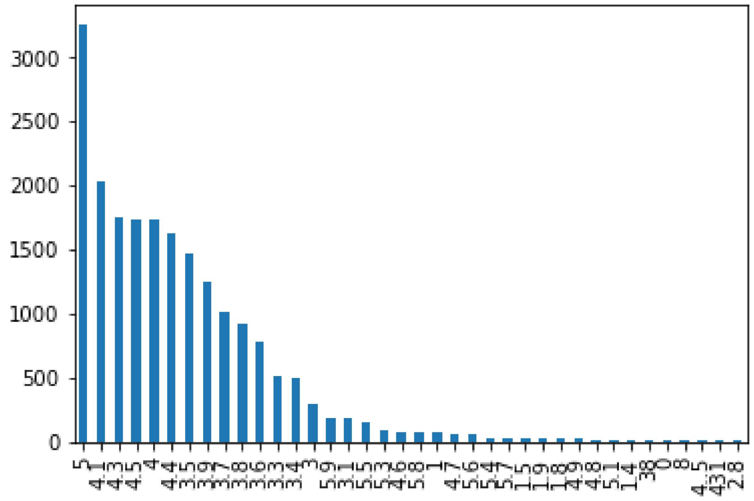
Normal Distribution Curve

In [15]:

```
df['Prod_Rating'].value_counts().plot(kind='bar')
```

Out[15]:

```
<AxesSubplot:>
```



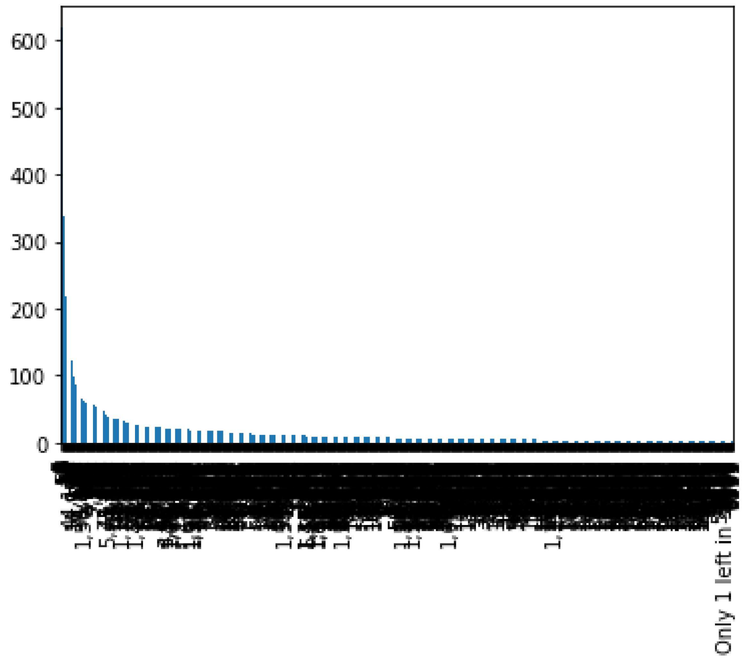
In the above line plot, Prod_Rating the feature is proportional to the Rise feature

In [16]:

```
df['Prod_Review'].value_counts().plot(kind='bar')
```

Out[16]:

```
<AxesSubplot:>
```



In the above line plot, Prod_Review the feature is proportional to the down feature

Correlation

In [17]:

```
df.corr()
```

Out[17]:

—

In [47]:

```
df.skew()
```

Out[47]:

```
Series([], dtype: float64)
```

Exploratory Data Analysis (EDA)

Now here we will be looking at the kind of columns data has.

In [7]:

```
df.columns
```

Out[7]:

```
Index(['Prod_Name', 'Prod_Rating', 'Prod_Review'], dtype='object')
```

To know more about the dataset

In [26]:

```
df.describe()
```

Out[26]:

	Prod_Name	Prod_Rating	Prod_Review
count	19993	19993	19953
unique	4922	38	1275
top	Men Digital Watch	5	5
freq	197	3251	620

Checking the different Prod_Rating

In [27]:

```
df['Prod_Rating'].nunique()
```

Out[27]:

```
38
```

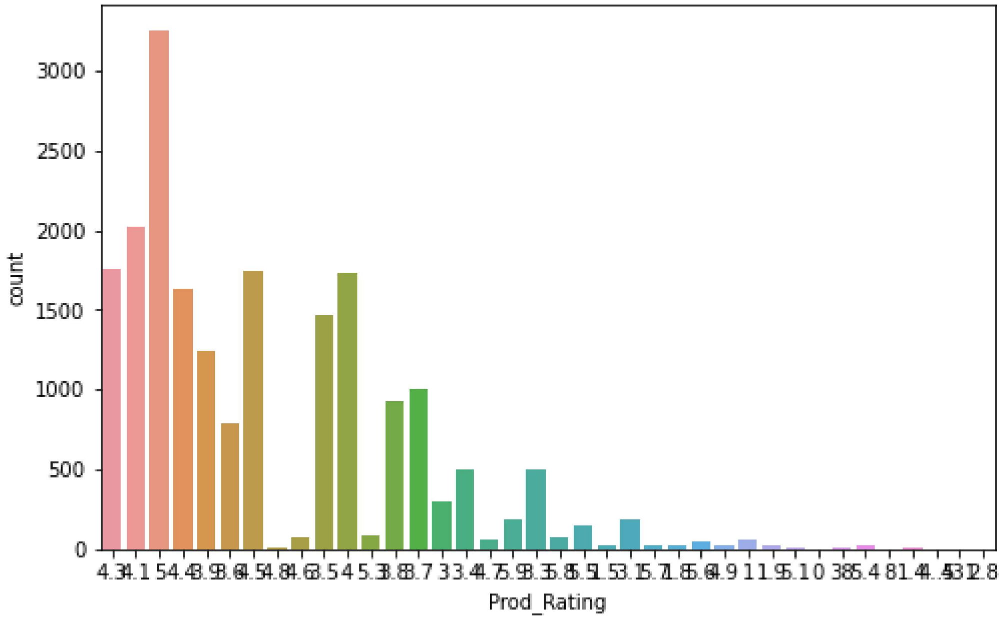
In [28]:

```
print(df['Prod_Rating'].unique())
```

```
['4.3' '4.1' '5' '4.4' '3.9' '3.6' '4.5' '4.8' '4.6' '3.5' '4' '5.3' '3.8'
 '3.7' '3' '3.4' '4.7' '5.9' '3.3' '5.8' '5.5' '1.5' nan '3.1' '5.7' '1.8'
 '5.6' '4.9' '1' '1.9' '5.1' '0' '38' '5.4' '8' '1.4' '4..5' '431' '2.8']
```

In [29]:

```
plt.figure(figsize=(8,5))
sns.countplot(x=df.Prod_Rating);
```



In [ ]:

```
x=df['Prod_Review'].values
y=df['Prod_Rating'].values
```

In [ ]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [36]:

```
from pandas import read_csv
# load dataset
url = 'C:\\Users\ADMIN\Desktop\\varshu.csv'
dataframe = read_csv(url, header=None)
# split into input and output elements
data = dataframe.values
X, y = data[:, :-1], data[:, -1]
print(X.shape, y.shape)
```

```
(20000, 2) (20000,)
```

In [37]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(13400, 2) (6600, 2) (13400,) (6600,)
```

load dataset

summarize shape

In [41]:

```
from pandas import read_csv
url = 'C:\\Users\ADMIN\Desktop\\varshu.csv'
df = read_csv(url, header=None)
print(df.shape)
```

```
(20000, 3)
```

split into inputs and outputs

In [42]:

```
X, y = data[:, :-1], data[:, -1]
print(X.shape, y.shape)
```

```
(20000, 2) (20000,)
```

splitting X and y into training and testing sets

In [43]:

```
from sklearn.datasets import load_iris
iris = load_iris()

# store the feature matrix (X) and response vector (y)
X = iris.data
y = iris.target

# splitting X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=1)

# training the model on training set
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)

# making predictions on the testing set
y_pred = gnb.predict(X_test)

# comparing actual response values (y_test) with predicted response values (y_pred)
from sklearn import metrics
print("Gaussian Naive Bayes model accuracy(in %):", metrics.accuracy_score(y_test, y_pre
```

```
Gaussian Naive Bayes model accuracy(in %): 95.0
```

Necessary imports Instantiating logistic regression classifier Print the tuned parameters and score

In [44]:

```

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV

# Creating the hyperparameter grid
c_space = np.logspace(-5, 8, 15)
param_grid = {'C': c_space}

# Instantiating logistic regression classifier
logreg = LogisticRegression()

# Instantiating the GridSearchCV object
logreg_cv = GridSearchCV(logreg, param_grid, cv = 5)

logreg_cv.fit(X, y)

# Print the tuned parameters and score
print("Tuned Logistic Regression Parameters: {}".format(logreg_cv.best_params_))
print("Best score is {}".format(logreg_cv.best_score_))
```

```
Tuned Logistic Regression Parameters: {'C': 31.622776601683793}
Best score is 0.9800000000000001
```

Conclusion

This is the best model Logistic Regression for Ratings Prediction Project. We also suggest that people take into consideration the features that were deemed as most important as seen in the previous section; this might help them estimate the Ratings Prediction Project.

Thank you