In [23]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [24]:
```
'https://github.com/abdelrahmansamir1/Women-s-E-Commerce-Clothing-Reviews/raw/mai
```

In [25]:
```
d=d.drop('Unnamed: 0',axis=1)
```

In [26]:
```
d.head()
```

Out[26]:

	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	
0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	In
1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	D
2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	D
3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	
4	847	47	Flattering shirt	This shirt is very flattering to all due to th...	5	1	6	General	Tops	E

In [27]:

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23486 entries, 0 to 23485
Data columns (total 10 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Clothing ID              23486 non-null  int64 
 1   Age                      23486 non-null  int64 
 2   Title                    19676 non-null  object
 3   Review Text              22641 non-null  object
 4   Rating                   23486 non-null  int64 
 5   Recommended IND          23486 non-null  int64 
 6   Positive Feedback Count  23486 non-null  int64 
 7   Division Name            23472 non-null  object
 8   Department Name          23472 non-null  object
 9   Class Name               23472 non-null  object
dtypes: int64(5), object(5)
memory usage: 1.8+ MB
```

In [28]:

```
d.shape
```

Out[28]:

```
(23486, 10)
```

In [29]:

```
d.isna().sum()
```

Out[29]:

```
Clothing ID                   0
Age                           0
Title                      3810
Review Text                 845
Rating                        0
Recommended IND               0
Positive Feedback Count       0
Division Name                14
Department Name              14
Class Name                   14
dtype: int64
```

In [36]:

```
import numpy as np
d[d['Review Text']==""]=np.NaN
```

In [37]:

```
d['Review Text'].fillna('No Review',inplace=True)
```

```
In [38]: d.isna().sum()
```

```
Out[38]: Clothing ID                   0
         Age                           0
         Title                      3810
         Review Text                   0
         Rating                        0
         Recommended IND               0
         Positive Feedback Count       0
         Division Name                14
         Department Name              14
         Class Name                   14
         dtype: int64
```

```
In [39]: d['Review Text']
```

```
Out[39]: 0        Absolutely wonderful - silky and sexy and comf...
         1        Love this dress!  it's sooo pretty.  i happene...
         2        I had such high hopes for this dress and reall...
         3        I love, love, love this jumpsuit. it's fun, fl...
         4        This shirt is very flattering to all due to th...
                                        ...
         23481    I was very happy to snag this dress at such a ...
         23482    It reminds me of maternity clothes. soft, stre...
         23483    This fit well, but the top was very see throug...
         23484    I bought this dress for a wedding i have this ...
         23485    This dress in a lovely platinum is feminine an...
         Name: Review Text, Length: 23486, dtype: object
```

Define Target(y) and Feature(x)

```
In [40]: d.columns
```

```
Out[40]: Index(['Clothing ID', 'Age', 'Title', 'Review Text', 'Rating',
                'Recommended IND', 'Positive Feedback Count', 'Division Name',
                'Department Name', 'Class Name'],
               dtype='object')
```

```
In [41]: x=d['Review Text']
```

```
In [42]: y=d['Rating']
```

```
In [44]: d['Rating'].value_counts()
```

```
Out[44]: 5.0    13131
         4.0     5077
         3.0     2871
         2.0     1565
         1.0      842
         Name: Rating, dtype: int64
```

Train Test Split

```
In [45]: from sklearn.model_selection import  train_test_split
```

```
In [48]: x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7,stratify=y,ranc
```

```
In [49]: x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
Out[49]: ((16440,), (7046,), (16440,), (7046,))
```

Get Feature Text Conversion to Tokens

```
In [53]: from sklearn.feature_extraction.text import CountVectorizer
```

```
In [56]: cv=CountVectorizer(lowercase=True,analyzer='word',ngram_range=(2,3),stop_words='e
```

```
In [57]: x_train=cv.fit_transform(x_train)
```

```
In [58]: cv.get_feature_names_out()
```

```
Out[58]: array(['10 12', '10 bought', '10 fit', ..., 'yellow color', 'yoga pants',
                'zipper little'], dtype=object)
```

```
In [59]: x_train.toarray()
```

```
Out[59]: array([[0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                ...,
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

```
In [60]: x_test=cv.fit_transform(x_test)
```

```
In [61]: cv.get_feature_names_out()
```

```
Out[61]: array(['10 12', '10 dress', '10 fit', ..., 'years come', 'years old',
                'yoga pants'], dtype=object)
```

In [62]:
```
x_test.toarray()
```

Out[62]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

Get Model Train

In [64]:
```
from sklearn.naive_bayes import MultinomialNB
```

In [65]:
```
m=MultinomialNB()
```

In [66]:
```
m.fit(x_train,y_train)
```

Out[66]:
```
▾ MultinomialNB
MultinomialNB()
```

Get Model Prediction

In [67]:
```
y_pred=m.predict(x_test)
```

In [68]:
```
y_pred.shape
```

Out[68]:
```
(7046,)
```

In [69]:
```
y_pred
```

Out[69]:
```
array([1., 5., 5., ..., 5., 5., 5.])
```

Probability of Each Predicted Class

In [70]:
```
m.predict_proba(x_test)
```

Out[70]:
```
array([[0.71118473, 0.02625165, 0.15465118, 0.01496876, 0.09294369],
       [0.02416867, 0.04769471, 0.35268622, 0.16185007, 0.41360034],
       [0.03582725, 0.06660584, 0.12226277, 0.21618005, 0.55912409],
       ...,
       [0.02320281, 0.08950939, 0.08962183, 0.16719203, 0.63047394],
       [0.01167675, 0.00202714, 0.08539004, 0.34347398, 0.55743209],
       [0.03959824, 0.05612822, 0.00688869, 0.1560574 , 0.74132745]])
```

Get Model Evaluation

```
In [71]: from sklearn.metrics import confusion_matrix,classification_report
```

```
In [72]: print(confusion_matrix(y_test,y_pred))
```

```
[[  15   13   45   36  144]
 [  43   43   86   85  213]
 [ 116   78  113  166  388]
 [ 166  108  194  336  719]
 [ 371  272  349  722 2225]]
```

```
In [73]: print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

         1.0       0.02      0.06      0.03       253
         2.0       0.08      0.09      0.09       470
         3.0       0.14      0.13      0.14       861
         4.0       0.25      0.22      0.23      1523
         5.0       0.60      0.56      0.58      3939

    accuracy                           0.39      7046
   macro avg       0.22      0.21      0.21      7046
weighted avg       0.42      0.39      0.40      7046
```

Recategories Ratings as Poor(0) and Good(1)

```
In [75]: d['Rating'].value_counts()
```

```
Out[75]: 5.0    13131
         4.0     5077
         3.0     2871
         2.0     1565
         1.0      842
         Name: Rating, dtype: int64
```

Re-Rating as 1,2,3 as 0 and 4,5 as 1

```
In [77]: d.replace({'Rating':{1:0,2:0,3:0,4:1,5:1}},inplace=True)
```

```
In [79]: y=d['Rating']
```

```
In [80]: x=d['Review Text']
```

Train Test Split

In [81]:
```
from sklearn.model_selection import  train_test_split
```

In [82]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7,stratify=y,ran
```

In [83]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[83]:
```
((16440,), (7046,), (16440,), (7046,))
```

Get Feature Text Conversion to Tokens

In [84]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [85]:
```
cv=CountVectorizer(lowercase=True,analyzer='word',ngram_range=(2,3),stop_words='
```

In [86]:
```
x_train=cv.fit_transform(x_train)
```

In [87]:
```
x_test=cv.fit_transform(x_test)
```

Get Model Re-Train

In [88]:
```
from sklearn.naive_bayes import MultinomialNB
```

In [89]:
```
m=MultinomialNB()
```

In [90]:
```
m.fit(x_train,y_train)
```

Out[90]:
```
▾ MultinomialNB
MultinomialNB()
```

Get Model Prediction

In [91]:
```
y_pred=m.predict(x_test)
```

In [92]:
```
y_pred.shape
```

Out[92]:
```
(7046,)
```

```
In [93]: y_pred

Out[93]: array([1., 1., 1., ..., 1., 1., 1.])
```

Get Model Evaluation

```
In [94]: from sklearn.metrics import confusion_matrix,classification_report
```

```
In [95]: print(confusion_matrix(y_test,y_pred))

[[ 449 1134]
 [ 989 4474]]
```

```
In [99]: print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

         0.0       0.31      0.28      0.30      1583
         1.0       0.80      0.82      0.81      5463

    accuracy                           0.70      7046
   macro avg       0.56      0.55      0.55      7046
weighted avg       0.69      0.70      0.69      7046
```

```
In [ ]:
```