Women Cloth Reviews Prediction with Multi Nomial Naïve Bayes

The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.

Import Library

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

Import Dataset

```
df = pd.read_csv('https://github.com/YBIFoundation/ProjectHub/raw/main/Women%2
```

```
df.head()
```

	Clothing ID	Age	Title	Review	Rating	Recommended	Positive Feedback	Division	Department	Category
0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses
3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	Pants

```
df.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             22641 non-null  object
 4   Rating             23486 non-null  int64 
 5   Recommended        23486 non-null  int64 
 6   Positive Feedback  23486 non-null  int64 
 7   Division           23472 non-null  object
 8   Department         23472 non-null  object
 9   Category           23472 non-null  object
dtypes: int64(5), object(5)
memory usage: 1.8+ MB
```

```
df.shape
```

```
(23486, 10)
```

Missing Values

Remove missing values in Reviews columns with No Review text

```
df.isna().sum()
```

```
Clothing ID             0
Age                     0
Title                3810
Review                845
Rating                  0
Recommended             0
Positive Feedback       0
Division               14
Department             14
Category               14
dtype: int64
```

```
df[df['Review']==""]=np.NaN
```

```
df['Review'].fillna("No Review",inplace=True)
```

```
df.isna().sum()
```

```
Clothing ID             0
Age                     0
Title                3810
Review                  0
Rating                  0
```

```
Recommended             0
Positive Feedback       0
Division               14
Department             14
Category               14
dtype: int64
```

```
df['Review']
```

```
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, Length: 23486, dtype: object
```

Define Target (y) and Feature (X)

```
df.columns
```

```
Index(['Clothing ID', 'Age', 'Title', 'Review', 'Rating', 'Recommended',
       'Positive Feedback', 'Division', 'Department', 'Category'],
      dtype='object')
```

```
X = df['Review']
```

```
y = df['Rating']
```

```
df['Rating'].value_counts()
```

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

Train Test Split

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, st
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((16440,), (7046,), (16440,), (7046,))
```

Get Feature Text Conversion to Tokens

```
from sklearn.feature_extraction.text import CountVectorizer
```

```
 cv= CountVectorizer(lowercase = True, analyzer='word', ngram_r
```

```
X_train = cv.fit_transform(X_train)
```

```
cv.get_feature_names_out()
```

```
array(['10 12', '110 lbs', '115 lbs', '12 14', '120 lbs', '125 lbs',
       '130 lbs', '135 lbs', '140 lbs', '145 lbs', '150 lbs', '26 waist',
       '36 hips', 'able wear', 'absolutely beautiful',
       'absolutely gorgeous', 'absolutely love', 'absolutely love dress',
       'addition wardrobe', 'agree previous', 'agree reviewer',
       'agree reviewers', 'arm holes', 'athletic build', 'based reviews',
       'bathing suit', 'beautiful color', 'beautiful colors',
       'beautiful dress', 'beautiful fabric', 'beautiful fit',
       'beautiful love', 'beautiful person', 'beautiful quality',
       'beautiful sweater', 'better fit', 'better person', 'big usually',
       'bit big', 'bit boxy', 'bit large', 'bit long', 'bit longer',
       'bit loose', 'bit short', 'bit shorter', 'bit small', 'bit snug',
       'bit tight', 'black pants', 'black white', 'blue color',
       'blue motif', 'body type', 'body types', 'bought black',
       'bought blue', 'bought colors', 'bought dress', 'bought green',
       'bought large', 'bought medium', 'bought regular', 'bought sale',
       'bought shirt', 'bought size', 'bought small', 'bought usual',
       'bought white', 'bought xs', 'bra size', 'bra straps',
       'broad shoulders', 'bust area', 'came colors', 'cami underneath',
       'casual dress', 'casual look', 'caught eye', 'chest area',
       'cloth stone', 'cold water', 'color beautiful', 'color design',
       'color dress', 'color fabric', 'color fit', 'color gorgeous',
       'color great', 'color just', 'color looks', 'color love',
       'color lovely', 'color nice', 'color perfect', 'color pretty',
       'color really', 'color soft', 'color style', 'colors beautiful',
       'colors great', 'colors vibrant', 'comfortable cute',
       'comfortable dress', 'comfortable easy', 'comfortable fabric',
       'comfortable fit', 'comfortable flattering', 'comfortable great',
       'comfortable looks', 'comfortable love', 'comfortable material',
       'comfortable soft', 'comfortable stylish', 'comfortable wear',
       'compliments time', 'compliments time wear', 'compliments wear',
       'compliments wore', 'coral color', 'cream color', 'cut flattering',
       'cute comfortable', 'cute design', 'cute dress', 'cute flattering',
       'cute great', 'cute love', 'cute shirt', 'cuter person',
       'date night', 'deal breaker', 'decided order', 'decided try',
       'definitely recommend', 'definitely runs', 'definitely size',
       'definitely worth', 'denim jacket', 'did fit', 'did like',
       'did look', 'did size', 'did work', 'didn feel', 'didn fit',
       'didn like', 'didn look', 'didn realize', 'didn think',
       'didn want', 'didn work', 'different colors', 'does fit',
       'does look', 'does model', 'does run', 'does run large',
       'does run little', 'doesn look', 'doesn make', 'doesn work',
       'don know', 'don like', 'don love', 'don mind', 'don need',
       'don think', 'don want', 'don wear', 'drapes beautifully',
       'drapes nicely', 'dress absolutely', 'dress adorable',
       'dress beautiful', 'dress bit', 'dress bought', 'dress color',
       'dress comfortable', 'dress cute', 'dress definitely', 'dress did',
       'dress does', 'dress dress', 'dress fabric', 'dress fall',
       'dress fit', 'dress fits', 'dress flattering', 'dress gorgeous',
       'dress great', 'dress just', 'dress like', 'dress little',
       'dress look', 'dress looked', 'dress looks', 'dress love',
       'dress lovely', 'dress makes', 'dress material', 'dress nice',
       'dress online', 'dress ordered', 'dress perfect', 'dress pretty',
       'dress really', 'dress runs', 'dress runs large', 'dress saw',
       'dress short', 'dress size', 'dress store', 'dress summer',
       'dress super', 'dress wear', 'dress work', 'dressing room',
       'dry clean', 'early fall', 'easily dressed', 'easy dress',
       'easy wear', 'elastic waist', 'embroidery beautiful',
```

```
X_train.toarray()
```

```
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]])
```

```
X_test = cv.transform(X_test)
```

```
cv.get_feature_names_out()
```

```
array(['10 12', '110 lbs', '115 lbs', '12 14', '120 lbs', '125 lbs',
       '130 lbs', '135 lbs', '140 lbs', '145 lbs', '150 lbs', '26 waist',
       '36 hips', 'able wear', 'absolutely beautiful',
       'absolutely gorgeous', 'absolutely love', 'absolutely love dress',
       'addition wardrobe', 'agree previous', 'agree reviewer',
       'agree reviewers', 'arm holes', 'athletic build', 'based reviews',
       'bathing suit', 'beautiful color', 'beautiful colors',
       'beautiful dress', 'beautiful fabric', 'beautiful fit',
       'beautiful love', 'beautiful person', 'beautiful quality',
       'beautiful sweater', 'better fit', 'better person', 'big usually',
       'bit big', 'bit boxy', 'bit large', 'bit long', 'bit longer',
       'bit loose', 'bit short', 'bit shorter', 'bit small', 'bit snug',
       'bit tight', 'black pants', 'black white', 'blue color',
       'blue motif', 'body type', 'body types', 'bought black',
       'bought blue', 'bought colors', 'bought dress', 'bought green',
       'bought large', 'bought medium', 'bought regular', 'bought sale',
       'bought shirt', 'bought size', 'bought small', 'bought usual',
       'bought white', 'bought xs', 'bra size', 'bra straps',
       'broad shoulders', 'bust area', 'came colors', 'cami underneath',
       'casual dress', 'casual look', 'caught eye', 'chest area',
       'cloth stone', 'cold water', 'color beautiful', 'color design',
       'color dress', 'color fabric', 'color fit', 'color gorgeous',
       'color great', 'color just', 'color looks', 'color love',
       'color lovely', 'color nice', 'color perfect', 'color pretty',
       'color really', 'color soft', 'color style', 'colors beautiful',
       'colors great', 'colors vibrant', 'comfortable cute',
       'comfortable dress', 'comfortable easy', 'comfortable fabric',
       'comfortable fit', 'comfortable flattering', 'comfortable great',
       'comfortable looks', 'comfortable love', 'comfortable material',
       'comfortable soft', 'comfortable stylish', 'comfortable wear',
       'compliments time', 'compliments time wear', 'compliments wear',
       'compliments wore', 'coral color', 'cream color', 'cut flattering',
       'cute comfortable', 'cute design', 'cute dress', 'cute flattering',
       'cute great', 'cute love', 'cute shirt', 'cuter person',
       'date night', 'deal breaker', 'decided order', 'decided try',
       'definitely recommend', 'definitely runs', 'definitely size',
       'definitely worth', 'denim jacket', 'did fit', 'did like',
       'did look', 'did size', 'did work', 'didn feel', 'didn fit',
       'didn like', 'didn look', 'didn realize', 'didn think',
       'didn want', 'didn work', 'different colors', 'does fit',
       'does look', 'does model', 'does run', 'does run large',
       'does run little', 'doesn look', 'doesn make', 'doesn work',
       'don know', 'don like', 'don love', 'don mind', 'don need',
       'don think', 'don want', 'don wear', 'drapes beautifully',
       'drapes nicely', 'dress absolutely', 'dress adorable',
       'dress beautiful', 'dress bit', 'dress bought', 'dress color',
       'dress comfortable', 'dress cute', 'dress definitely', 'dress did',
       'dress does', 'dress dress', 'dress fabric', 'dress fall',
       'dress fit', 'dress fits', 'dress flattering', 'dress gorgeous',
       'dress great', 'dress just', 'dress like', 'dress little',
       'dress look', 'dress looked', 'dress looks', 'dress love',
       'dress lovely', 'dress makes', 'dress material', 'dress nice',
       'dress online', 'dress ordered', 'dress perfect', 'dress pretty',
       'dress really', 'dress runs', 'dress runs large', 'dress saw',
       'dress short', 'dress size', 'dress store', 'dress summer',
       'dress super', 'dress wear', 'dress work', 'dressing room',
       'dry clean', 'early fall', 'easily dressed', 'easy dress',
       'easy wear', 'elastic waist', 'embroidery beautiful',
```

```
X_test.toarray()
```

```
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]])
```

Get Model Train

The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.

```
from sklearn.naive_bayes import MultinomialNB
```

```
model = MultinomialNB()
```

```
model.fit(X_train, y_train)
```

```
MultinomialNB()
```

Get Model Prediction

```
y_pred = model.predict(X_test)
```

```
y_pred.shape
```

```
(7046,)
```

```
y_pred
```

```
array([5., 5., 5., ..., 5., 5., 5.])
```

Get Probability of Each Predicted Class

```
model.predict_proba(X_test)
```

```
array([[2.01586401e-02, 1.78804627e-01, 7.57793454e-02, 2.81062761e-01,
        4.44194626e-01],
       [1.11435859e-01, 1.08171814e-01, 1.69334873e-01, 1.63435724e-01,
        4.47621730e-01],
       [3.58272506e-02, 6.66058394e-02, 1.22262774e-01, 2.16180049e-01,
        5.59124088e-01],
       ...,
       [7.70892022e-02, 1.46522491e-01, 2.06072782e-01, 2.24938598e-01,
        3.45376927e-01],
       [1.00692716e-03, 1.58459813e-03, 3.76282076e-03, 4.04244108e-02,
        9.53221243e-01],
       [1.19174157e-04, 2.69385436e-05, 6.36511937e-04, 4.08820183e-02,
        9.58335357e-01]])
```

Get Model Evaluation

```
from sklearn.metrics import confusion_matrix, classification_report
```

```
print(confusion_matrix(y_test, y_pred))
```

```
[[  21   23   39   21  149]
 [  11   62  105   50  242]
 [  20   63  192  161  425]
 [  20   38  117  274 1074]
 [  10   40   92  304 3493]]
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

         1.0       0.26      0.08      0.13       253
         2.0       0.27      0.13      0.18       470
         3.0       0.35      0.22      0.27       861
         4.0       0.34      0.18      0.23      1523
         5.0       0.65      0.89      0.75      3939

    accuracy                           0.57      7046
   macro avg       0.37      0.30      0.31      7046
weighted avg       0.51      0.57      0.52      7046
```

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

```
df['Rating'].value_counts()
```

```
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, 4 as 0 and 5 as 1

```
df.replace({'Rating' : { 1 : 0, 2 : 0, 3 : 0, 4 : 0, 5 : 1}}, inplace = True)
```

```
y = df['Rating']
```

```
X = df['Review']
```

Train Test Split

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, st
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((16440,), (7046,), (16440,), (7046,))
```

Get Feature Text Conversion to Tokens

```
from sklearn.feature_extraction.text import CountVectorizer
```

```
cv= CountVectorizer(lowercase = True, analyzer='word', ngram_range=(2, 3), sto
```

```
X_train = cv.fit_transform(X_train)
```

```
X_test = cv.transform(X_test)
```

Get Model Re-Train

```
from sklearn.naive_bayes import MultinomialNB
```

```
model = MultinomialNB()
```

```
model.fit(X_train, y_train)
```

```
MultinomialNB()
```

Get Model Prediction

```
y_pred = model.predict(X_test)
```

```
y_pred.shape
```

```
(7046,)
```

```
y_pred
```

```
array([1., 0., 1., ..., 1., 0., 1.])
```

Get Model Evaluation

```
from sklearn.metrics import confusion_matrix, classification_report
```

```
print(confusion_matrix(y_test, y_pred))
```

```
[[1721 1386]
 [ 703 3236]]
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

         0.0       0.71      0.55      0.62      3107
         1.0       0.70      0.82      0.76      3939

    accuracy                           0.70      7046
   macro avg       0.71      0.69      0.69      7046
weighted avg       0.70      0.70      0.70      7046
```

✓ 0s completed at 2:09 PM ● ×