CountVectorizer

June 21, 2022

[1]:
```
import re , pandas as pd , numpy , string , nltk
```

[2]:
```
stopwords = nltk.corpus.stopwords.words('english')
lem = nltk.WordNetLemmatizer()
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\sriva\AppData\Roaming\nltk_data…
[nltk_data]   Package omw-1.4 is already up-to-date!
```

[2]:
```
True
```

[3]:
```
df = pd.read_csv('news.csv')
df = df.drop(['Unnamed: 0'] , axis = 1)
df
```

[3]:

	title	text	label
0	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello…	FAKE
1	Watch The Exact Moment Paul Ryan Committed Pol…	Google Pinterest Digg Linkedin Reddit Stumbleu…	FAKE
2	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon…	REAL
3	Bernie supporters on Twitter erupt in anger ag…	- Kaydee King (@KaydeeKing) November 9, 2016 T…	FAKE
4	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners…	REAL
…	…	…	…
6330	State Department says it can't find emails fro…	The State Department told the Republican Natio…	REAL
6331	The 'P' in PBS Should Stand for 'Plutocratic' …	The 'P' in PBS Should Stand for 'Plutocratic' …	FAKE

```
6332   Anti-Trump Protesters Are Tools of the Oligar…  FAKE
6333  ADDIS ABABA, Ethiopia -President Obama convene…  REAL
6334  Jeb Bush Is Suddenly Attacking Trump. Here's W…  REAL

[6335 rows x 3 columns]
```

[4]:
```
def clean_text(text):
    text="".join([word.lower()for word in text if word not in string.
 ↪punctuation])
    tokens=re.split('\W+',text)
    text=[lem.lemmatize(word)for word in tokens if word not in stopwords]
    return text
```

[5]:
```
text = []
for i in df['text']:
    text.append(clean_text(i))
```

[6]:
```
len(text[0])
```

[6]:
```
671
```

[7]:
```
df['clean_data'] = pd.Series(text)
```

[8]:
```
df
```

[8]:
```
                                                  title  \
0                           You Can Smell Hillary's Fear
1      Watch The Exact Moment Paul Ryan Committed Pol…
2            Kerry to go to Paris in gesture of sympathy
3      Bernie supporters on Twitter erupt in anger ag…
4       The Battle of New York: Why This Primary Matters
…                                                    …
6330  State Department says it can't find emails fro…
6331  The 'P' in PBS Should Stand for 'Plutocratic' …
6332  Anti-Trump Protesters Are Tools of the Oligarc…
6333  In Ethiopia, Obama seeks progress on peace, se…
6334  Jeb Bush Is Suddenly Attacking Trump. Here's W…

                                                  text label  \
0      Daniel Greenfield, a Shillman Journalism Fello…  FAKE
1      Google Pinterest Digg Linkedin Reddit Stumbleu…  FAKE
2      U.S. Secretary of State John F. Kerry said Mon…  REAL
3      - Kaydee King (@KaydeeKing) November 9, 2016 T…  FAKE
4      It's primary day in New York and front-runners…  REAL
…                                                    …     …
6330  The State Department told the Republican Natio…  REAL
6331  The 'P' in PBS Should Stand for 'Plutocratic' …  FAKE
```

```
6332   Anti-Trump Protesters Are Tools of the Oligar…  FAKE
6333  ADDIS ABABA, Ethiopia -President Obama convene…  REAL
6334  Jeb Bush Is Suddenly Attacking Trump. Here's W…  REAL

                                                clean_data
0     [daniel, greenfield, shillman, journalism, fel…
1     [google, pinterest, digg, linkedin, reddit, st…
2     [u, secretary, state, john, f, kerry, said, mo…
3     [, kaydee, king, kaydeeking, november, 9, 2016…
4     [primary, day, new, york, frontrunners, hillar…
…                                                   …
6330  [state, department, told, republican, national…
6331  [p, pb, stand, plutocratic, pentagon, posted, …
6332  [, antitrump, protester, tool, oligarchy, refo…
6333  [addis, ababa, ethiopia, president, obama, con…
6334  [jeb, bush, suddenly, attacking, trump, here, …

[6335 rows x 4 columns]
```

[ ]:

[9]:
```
from sklearn.feature_extraction.text import CountVectorizer

count_vect=CountVectorizer(analyzer=clean_text)
X_counts=count_vect.fit_transform(df['text'])
```

[10]:
```
X_counts.toarray()
```

[10]:
```
array([[1, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0],
       [1, 0, 0, …, 0, 0, 0],
       …,
       [2, 0, 0, …, 0, 0, 0],
       [1, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0]], dtype=int64)
```

[11]:
```
count_vect.get_feature_names()[0]
```

```
C:\Users\sriva\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will
be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

[11]:
```
''
```

[12]:
```
#count_vect.get_feature_names()

df2 = pd.DataFrame(X_counts.toarray())
df2.columns= [count_vect.get_feature_names()]
df2
```

[12]:

	0	00	000	0000	0000000031	000000031	0000035	00001400	00006	…	\
0	1	0	0	0	0	0	0	0	0	…	0
1	0	0	0	0	0	0	0	0	0	…	0
2	1	0	0	0	0	0	0	0	0	…	0
3	1	0	0	0	0	0	0	0	0	…	0
4	0	0	0	0	0	0	0	0	0	…	0
…	..	..	..	..	…	…	…	…	…	…	
6330	0	0	0	0	0	0	0	0	0	…	0
6331	0	0	0	0	0	0	0	0	0	…	0
6332	2	0	0	0	0	0	0	0	0	…	0
6333	1	0	0	0	0	0	0	0	0	…	0
6334	0	0	0	0	0	0	0	0	0	…	0

	ade								
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
…	..	..	..	…	..	..	…	…	..		
6330	0	0	0	0	0	0	0	0	0
6331	0	0	0	0	0	0	0	0	0
6332	0	0	0	0	0	0	0	0	0
6333	0	0	0	0	0	0	0	0	0
6334	0	0	0	0	0	0	0	0	0

[6335 rows x 76543 columns]

[13]:
```
df2.sum()
```

[13]:
```
           1836
0           545
00            9
000           3
0000          5
           …
             1
             1
1
 ade         1
             2
```

```
Length: 76543, dtype: int64
```

[14]:
```
y = df.iloc[:,2]
y
```

[14]:
```
0       FAKE
1       FAKE
2       REAL
3       FAKE
4       REAL
        …
6330    REAL
6331    FAKE
6332    FAKE
6333    REAL
6334    REAL
Name: label, Length: 6335, dtype: object
```

[15]:
```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
y = le.fit_transform(y)
y
```

[15]:
```
array([0, 0, 1, …, 0, 1, 1])
```

[16]:
```
df2.values
```

[16]:
```
array([[1, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0],
       [1, 0, 0, …, 0, 0, 0],
       …,
       [2, 0, 0, …, 0, 0, 0],
       [1, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0]], dtype=int64)
```

[17]:
```
x = df2.values

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
 ↪20,random_state=0)
```

[18]:
```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train,y_train)
```

[18]:
```
GaussianNB()
```

[19]:
```
y_pred = classifier.predict(x_test)
```

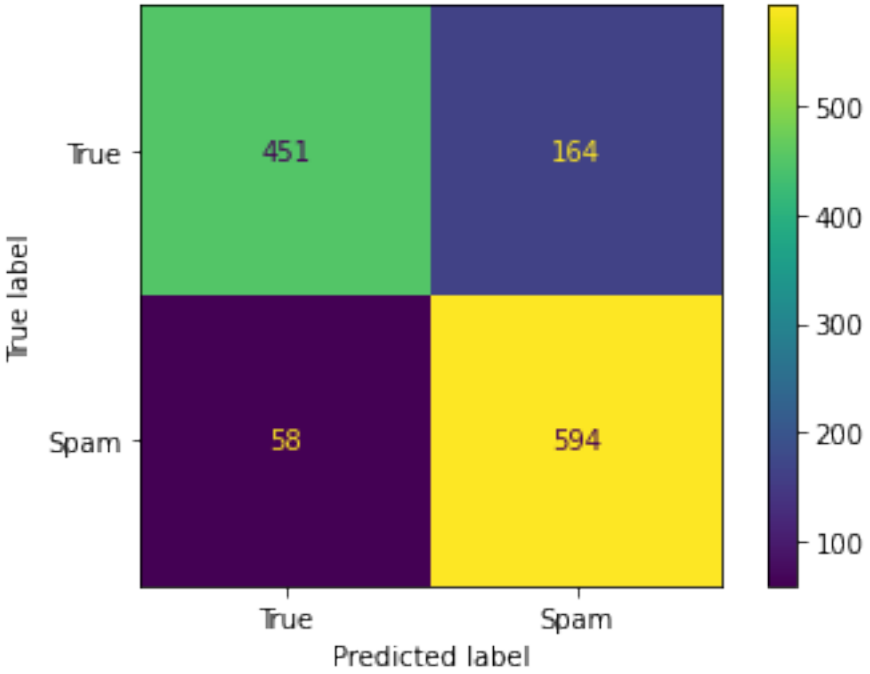
[23]:
```
from sklearn.metrics import confusion_matrix ,ConfusionMatrixDisplay
cm = confusion_matrix(y_test,y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm ,␣
 ↪display_labels=['True','Spam'])
```

[24]:
```
import seaborn as sns
cm
```

[24]:
```
array([[451, 164],
       [ 58, 594]], dtype=int64)
```

[27]:
```
disp.plot()
```

[27]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1bbd65f6f50>
```



[28]:
```
from sklearn.metrics import classification_report

cr = classification_report(y_test, y_pred)
```

[29]:
```
print(cr)
```

	precision	recall	f1-score	support
0	0.89	0.73	0.80	615
1	0.78	0.91	0.84	652
accuracy			0.82	1267
macro avg	0.83	0.82	0.82	1267
weighted avg	0.83	0.82	0.82	1267

[ ]: