TFIDF

June 21, 2022

[1]:
```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
```

[2]:
```
df = pd.read_csv('news.csv')
df
```

[2]:
```
      Unnamed: 0                                                title  \
0           8476                       You Can Smell Hillary's Fear
1          10294  Watch The Exact Moment Paul Ryan Committed Pol…
2           3608        Kerry to go to Paris in gesture of sympathy
3          10142  Bernie supporters on Twitter erupt in anger ag…
4            875   The Battle of New York: Why This Primary Matters
…            …                                                  …
6330        4490  State Department says it can't find emails fro…
6331        8062  The 'P' in PBS Should Stand for 'Plutocratic' …
6332        8622  Anti-Trump Protesters Are Tools of the Oligarc…
6333        4021  In Ethiopia, Obama seeks progress on peace, se…
6334        4330  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

[6335 rows x 4 columns]
```

[3]:
```
import nltk
stopwords = nltk.corpus.stopwords.words('english')
```

[21]:
```
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(df['text'])
```

[22]:
```
tfidf.get_feature_names_out()
```

[22]:
```
array(['00', '000', '0000', …, '  ', '    ', ' ade'], dtype=object)
```

[23]:
```
df2 = pd.DataFrame(X.toarray())
df2.columns = [tfidf.get_feature_names_out()]
df2
```

[23]:

	00	000	0000	000000031	00000031	000035	00006	0001	0001pt	0002	\
0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.028139	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
…	…	…	…	…	…	…	…	…	…	…	
6330	0.0	0.029299	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6331	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6332	0.0	0.007522	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6333	0.0	0.011331	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6334	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

	…										ade
0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
…	…	…	…	…	…	…	…	…	…	…	…
6330	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6331	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6332	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6333	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6334	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[6335 rows x 67659 columns]

[24]:
```
df2.sum()
```

[24]:
```
00            3.560481
000          37.380637
0000          0.097361
000000031     0.026123
00000031      0.053099
                …
```

```
            0.041217
              0.041217
             0.041217
            0.041217
 ade          0.032057
Length: 67659, dtype: float64
```

[25]:
```
y = df.iloc[:,3]
y
```

[25]:
```
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
```

[26]:
```
from sklearn.preprocessing import LabelEncoder

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

[26]:
```
array([0, 0, 1, …, 0, 1, 1])
```

[27]:
```
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)
```

[28]:
```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train,y_train)
```

[28]:
```
GaussianNB()
```

[29]:
```
y_pred = classifier.predict(x_test)
```

[30]:
```
y_pred
```

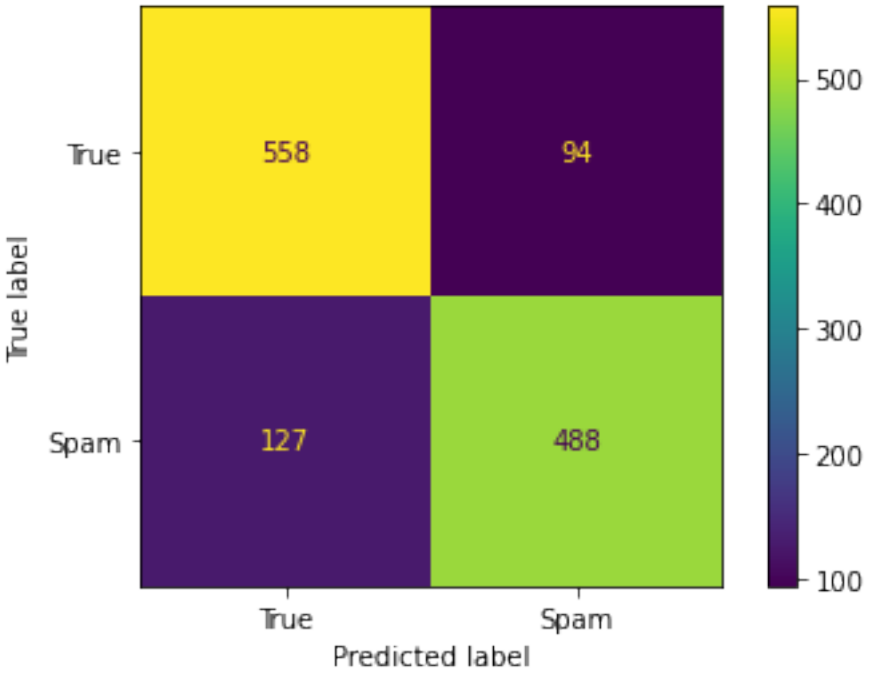
[30]:
```
array([1, 0, 0, …, 0, 1, 0])
```

[31]:
```
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
cm = confusion_matrix(y_test,y_pred, labels=[1,0])
disp = ConfusionMatrixDisplay(confusion_matrix=cm ,␣
 ↪display_labels=['True','Spam'])
```

[32]:
```
import seaborn as sns
```

[33]:
```
disp.plot()
```

[33]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1cbdde83460>
```



[34]:
```
from sklearn.metrics import classification_report

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

```
              precision    recall  f1-score   support

           0       0.84      0.79      0.82       615
           1       0.81      0.86      0.83       652

    accuracy                           0.83      1267
   macro avg       0.83      0.82      0.83      1267
```

```
weighted avg       0.83      0.83      0.83      1267
```

[ ]:

[ ]:

[ ]: