Exercise 3-2

September 17, 2023

[1]:
```
## DSC 550
## Carlos Cano
## Activity 3.2
```

[ ]:
```
## Part 1: Using the TextBlob Sentiment Analyzer
```

[19]:
```
## 1) Import the movie review data as a data frame and ensure that the data is␣
 ↪loaded properly.
```

[3]:
```
import pandas as pd
from textblob import TextBlob
from sklearn.metrics import accuracy_score
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import nltk
import re
```

[3]:
```
## Import movie review data
```

[4]:
```
## Data fields
## id - Unique ID of each review
## sentiment - Sentiment of the review; 1 for positive reviews and 0 for␣
 ↪negative reviews
## review - Text of the review
```

[7]:
```
df = pd.read_csv('labeledTrainData.tsv', sep = '\t')
df.head()
```

[7]:
```
       id  sentiment                                             review
0  5814_8          1  With all this stuff going down at the moment w...
1  2381_9          1  \The Classic War of the Worlds\" by Timothy Hi...
2  7759_3          0  The film starts with a manager (Nicholas Bell)...
3  3630_4          0  It must be assumed that those who praised this...
4  9495_8          1  Superbly trashy and wondrously unpretentious 8...
```

[8]:
```
## 2) How many of each positive and negative reviews are there?
```

[9]:
```
Pos_Neg = df['sentiment'].value_counts()
Pos_Neg
```

```
[9]: 1    12500
     0    12500
     Name: sentiment, dtype: int64
```

```
[20]: df['sentiment'].replace({0: 'negative', 1: 'positive'}, inplace=True)
      df.head()
```

```
[20]:       id sentiment                                             review
      0  5814_8  positive  With all this stuff going down at the moment w...
      1  2381_9  positive  \The Classic War of the Worlds\" by Timothy Hi...
      2  7759_3  negative  The film starts with a manager (Nicholas Bell)...
      3  3630_4  negative  It must be assumed that those who praised this...
      4  9495_8  positive  Superbly trashy and wondrously unpretentious 8...
```

```
[25]: display(df['sentiment'].value_counts())
```

```
positive    12500
negative    12500
Name: sentiment, dtype: int64
```

```
[10]: ## 3) Use TextBlob to classify each movie review as positive or negative. Assume␣
       ↪that a polarity score greater than or equal to zero is a positive sentiment␣
       ↪and less than 0 is a negative sentiment.
```

```
[21]: def textblob_analyzer(df):
          results = []
          def getPolarity(text):
              return TextBlob(text).sentiment.polarity
          for review in df:
              polarity = getPolarity(review)
              if polarity >= 0:
                  results.append('positive')
              else:
                  results.append('negative')
          return results
```

```
[12]: ## 4) Check the accuracy of this model. Is this model better than random␣
       ↪guessing?
```

```
[26]: sentiments_textblob = textblob_analyzer(df['review'].iloc[:25000])
```

```
[28]: print('TextBlob Accuracy is',round(accuracy_score(df['sentiment'].iloc[:25000],␣
       ↪sentiments_textblob)*100, 2),'%')
```

```
TextBlob Accuracy is 68.52 %
```

```
[ ]: ## 5) For up to five points extra credit, use another prebuilt text sentiment␣
      ↪analyzer, e.g., VADER, and repeat steps (3) and (4).
```

[30]:
```
## 5-3)
```

[51]:
```
def varder_analyzer(df):
    results = []
    for review in df:
        SIA_obj = SentimentIntensityAnalyzer()
        sentiment = SIA_obj.polarity_scores(review)
        if sentiment['compound'] >= 0:
            results.append('positive')
        else:
            results.append('negative')
    return results
```

[29]:
```
## 5-4)
```

[40]:
```
sentiments_vader = varder_analyzer(df['review'].iloc[:25000])
```

[50]:
```
print('Vader Accuracy is',round(accuracy_score(df['sentiment'].iloc[:25000],␣
 ↪sentiments_vader)*100, 2),'%')
```

```
Vader Accuracy is 69.4 %
```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[42]:
```
## Part 2: Prepping Text for a Custom Model
```

[53]:
```
## Import Test Data
```

[4]:
```
df2 = pd.read_csv('testData.tsv', sep = '\t')
df2.head()
```

[4]:
```
         id                                             review
0  12311_10  Naturally in a film who's main themes are of m...
1    8348_2  This movie is a disaster within a disaster fil...
2    5828_4  All in all, this is a movie for kids. We saw i...
3    7186_2  Afraid of the Dark left me with the impression...
4   12128_7  A very accurate depiction of small time mob li...
```

[5]:
```
## 1) Convert all text to lowercase letters.
```

[6]:
```
df2['review'] = df2['review'].str.lower()
df2.head()
```

[6]:
```
         id                                             review
0  12311_10  naturally in a film who's main themes are of m...
1    8348_2  this movie is a disaster within a disaster fil...
2    5828_4  all in all, this is a movie for kids. we saw i...
3    7186_2  afraid of the dark left me with the impression...
4   12128_7  a very accurate depiction of small time mob li...
```

[7]:
```
## 2) Remove punctuation and special characters from the text.
```

[8]:
```
df2['review'] = df2['review'].str.replace('\W',' ',regex = True)
df2['review'] = df2['review'].str.replace('\d+', '',regex = True)
df2['review'] = df2['review'].str.replace('_', '',regex = True)
df2.head()
```

[8]:
```
         id                                             review
0  12311_10  naturally in a film who s main themes are of m...
1    8348_2  this movie is a disaster within a disaster fil...
2    5828_4  all in all  this is a movie for kids  we saw i...
3    7186_2  afraid of the dark left me with the impression...
4   12128_7  a very accurate depiction of small time mob li...
```

[9]:
```
## 3) Remove stop words.
```

[20]:
```
from nltk.corpus import stopwords
stop = set(stopwords.words('english'))
```

[29]:
```
df2['review'] = df2['review'].apply(lambda x: ' '.join([word for word in x.
 ↪split() if word not in (stop)]))
df2.head()
```

[29]:
```
         id                                             review
0  12311_10  naturally film main themes mortality nostalgia...
1    8348_2  movie disaster within disaster film full great...
2    5828_4  movie kids saw tonight child loved one point k...
3    7186_2  afraid dark left impression several different ...
4   12128_7  accurate depiction small time mob life filmed ...
```

[30]:
```
## from nltk.corpus import brown
## english = set(nltk.corpus.words.words())
```

[ ]:
```
## df2['review'] = df2['review'].apply(lambda x: ' '.join(w for w in nltk.
 ↪wordpunct_tokenize() if w.lower() in words or not w.isalpha())
## df2.head()
```

[ ]:
```
## df2['review'] = df2['review'].apply(lambda x: ' '.join([word for word in x.
 ↪split() if word not in (english)]))
## df2.head()
```

[31]:

```
## 4) Apply NLTK's PorterStemmer.
```

[32]:

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
ps = PorterStemmer()
```

[33]:

```
def stem_sentences(sentence):
    tokens = sentence.split()
    stemmed_tokens = [ps.stem(token) for token in tokens]
    return ' '.join(stemmed_tokens)
```

[34]:

```
df2['review'] = df2['review'].apply(stem_sentences)
df2.head()
```

[34]:

```
         id                                             review
0  12311_10  natur film main theme mortal nostalgia loss in...
1    8348_2  movi disast within disast film full great acti...
2    5828_4  movi kid saw tonight child love one point kid ...
3    7186_2  afraid dark left impress sever differ screenpl...
4   12128_7  accur depict small time mob life film new jers...
```

[35]:

```
## 5) Create a bag-of-words matrix from your stemmed text (output from (4)),␣
↪where each row is a word-count vector for a single movie review (see sections␣
↪5.3 & 6.8 in the Machine Learning with Python Cookbook). Display the␣
↪dimensions of your bag-of-words matrix. The number of rows in this matrix␣
↪should be the same as the number of rows in your original data frame.
```

[36]:

```
cleaned_review = df2['review']
cleaned_review
```

[36]:

```
0        natur film main theme mortal nostalgia loss in...
1        movi disast within disast film full great acti...
2        movi kid saw tonight child love one point kid ...
3        afraid dark left impress sever differ screenpl...
4        accur depict small time mob life film new jers...
                               ...
24995    soni pictur classic look soni got right harri ...
24996    alway felt ms merkerson never gotten role fit ...
24997    disappoint movi familiar case read mark fuhrma...
24998    open sequenc fill black white shot reminisc go...
24999    great horror film peopl want vomit retch gore ...
Name: review, Length: 25000, dtype: object
```

[ ]:

```

```

[40]:

```
from sklearn.feature_extraction.text import CountVectorizer
count = CountVectorizer(stop_words = 'english')
bag_of_words_vec = count.fit_transform(cleaned_review)
```

```
[41]: df_bow = pd.DataFrame(bag_of_words_vec.toarray(), columns = count.
       ↪get_feature_names_out())
      df_bow
```

```
[41]:        aa  aaa  aaaaaaaaaaaahhhhhhhhhhhhhh  aaaaaaaargh  aaaaaaahhhhhhggg  \
      0       0    0                           0            0                 0
      1       0    0                           0            0                 0
      2       0    0                           0            0                 0
      3       0    0                           0            0                 0
      4       0    0                           0            0                 0
      ...    ..  ...                         ...          ...               ...
      24995   0    0                           0            0                 0
      24996   0    0                           0            0                 0
      24997   0    0                           0            0                 0
      24998   0    0                           0            0                 0
      24999   0    0                           0            0                 0

             aaaaagh  aaaaah  aaaaahhhh  aaaaargh  aaaaarrrrrrgggggghhhhhh  ...  \
      0            0       0          0         0                        0  ...
      1            0       0          0         0                        0  ...
      2            0       0          0         0                        0  ...
      3            0       0          0         0                        0  ...
      4            0       0          0         0                        0  ...
      ...        ...     ...        ...       ...                      ...  ...
      24995        0       0          0         0                        0  ...
      24996        0       0          0         0                        0  ...
      24997        0       0          0         0                        0  ...
      24998        0       0          0         0                        0  ...
      24999        0       0          0         0                        0  ...

             überwoman  ünel  ünfaith  üzümcü  ýs  þorleifsson  þór  żmijewski  \
      0              0     0        0       0   0            0    0          0
      1              0     0        0       0   0            0    0          0
      2              0     0        0       0   0            0    0          0
      3              0     0        0       0   0            0    0          0
      4              0     0        0       0   0            0    0          0
      ...          ...   ...      ...     ...  ..          ...  ...        ...
      24995          0     0        0       0   0            0    0          0
      24996          0     0        0       0   0            0    0          0
      24997          0     0        0       0   0            0    0          0
      24998          0     0        0       0   0            0    0          0
      24999          0     0        0       0   0            0    0          0

             ℷℵgnore  gnoregnoregnoregnore all
      0          0      0
      1          0      0
      2          0      0
```

```
3          0      0
4          0      0
...      ...    ...
24995      0      0
24996      0      0
24997      0      0
24998      0      0
24999      0      0

[25000 rows x 49074 columns]
```

[42]:
```
df_bow.describe()
```

[42]:
```
                 aa          aaa  aaaaaaaaaaaahhhhhhhhhhhhh   aaaaaaaargh  \
count  25000.000000  25000.00000                25000.000000  25000.000000
mean       0.000440      0.00024                    0.000040      0.000040
std        0.027565      0.01549                    0.006325      0.006325
min        0.000000      0.00000                    0.000000      0.000000
25%        0.000000      0.00000                    0.000000      0.000000
50%        0.000000      0.00000                    0.000000      0.000000
75%        0.000000      0.00000                    0.000000      0.000000
max        3.000000      1.00000                    1.000000      1.000000

       aaaaaaahhhhhhggg       aaaaagh        aaaaah     aaaaahhhh  \
count      25000.000000  25000.000000  25000.000000  25000.000000
mean           0.000040      0.000040      0.000040      0.000040
std            0.006325      0.006325      0.006325      0.006325
min            0.000000      0.000000      0.000000      0.000000
25%            0.000000      0.000000      0.000000      0.000000
50%            0.000000      0.000000      0.000000      0.000000
75%            0.000000      0.000000      0.000000      0.000000
max            1.000000      1.000000      1.000000      1.000000

           aaaaargh  aaaaarrrrrrgggggghhhhhh  ...     überwoman          ünel  \
count  25000.000000             25000.000000  ...  25000.000000  25000.000000
mean       0.000040                 0.000040  ...      0.000040      0.000040
std        0.006325                 0.006325  ...      0.006325      0.006325
min        0.000000                 0.000000  ...      0.000000      0.000000
25%        0.000000                 0.000000  ...      0.000000      0.000000
50%        0.000000                 0.000000  ...      0.000000      0.000000
75%        0.000000                 0.000000  ...      0.000000      0.000000
max        1.000000                 1.000000  ...      1.000000      1.000000

            ünfaith        üzümcü            ýs   þorleifsson           þór  \
count  25000.000000  25000.000000  25000.000000  25000.000000  25000.000000
mean       0.000040      0.000040      0.000040      0.000040      0.000040
std        0.006325      0.006325      0.006325      0.006325      0.006325
```

```
min        0.000000     0.000000     0.000000     0.000000     0.000000
25%        0.000000     0.000000     0.000000     0.000000     0.000000
50%        0.000000     0.000000     0.000000     0.000000     0.000000
75%        0.000000     0.000000     0.000000     0.000000     0.000000
max        1.000000     1.000000     1.000000     1.000000     1.000000

          żmijewski           אב
count  25000.000000  25000.000000  25000.000000
mean       0.000040      0.000040      0.000040
std        0.006325      0.006325      0.006325
min        0.000000      0.000000      0.000000
25%        0.000000      0.000000      0.000000
50%        0.000000      0.000000      0.000000
75%        0.000000      0.000000      0.000000
max        1.000000      1.000000      1.000000

[8 rows x 49074 columns]
```

[48]:

```
## 6) Create a term frequency-inverse document frequency (tf-idf) matrix from␣
 ↪your stemmed text, for your movie reviews (see section 6.9 in the Machine␣
 ↪Learning with Python Cookbook). Display the dimensions of your tf-idf matrix.␣
 ↪These dimensions should be the same as your bag-of-words matrix.
```

[43]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

[44]:

```
tr_idf_model  = TfidfVectorizer()
tf_idf_vector = tr_idf_model.fit_transform(df_bow)
```

[45]:

```
print(type(tf_idf_vector), tf_idf_vector.shape)
```

```
<class 'scipy.sparse._csr.csr_matrix'> (49074, 49074)
```

[46]:

```
tf_idf_array = tf_idf_vector.toarray()

print(tf_idf_array)
```

```
[[1. 0. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 0. 1. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 0. 0. 1.]]
```

[49]:

```
words_set = tr_idf_model.get_feature_names()

#print(words_set)
```

```
/Users/carloscano/opt/anaconda3/lib/python3.9/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)
```

[50]:

```
df_tf_idf = pd.DataFrame(tf_idf_array, columns = words_set)

df_tf_idf
```

[50]:

```
        aa  aaa  aaaaaaaaaaaahhhhhhhhhhhhhh  aaaaaaaargh  aaaaaaahhhhhggg  \
0      1.0  0.0                         0.0          0.0              0.0
1      0.0  1.0                         0.0          0.0              0.0
2      0.0  0.0                         1.0          0.0              0.0
3      0.0  0.0                         0.0          1.0              0.0
4      0.0  0.0                         0.0          0.0              1.0
...    ...  ...                         ...          ...              ...
49069  0.0  0.0                         0.0          0.0              0.0
49070  0.0  0.0                         0.0          0.0              0.0
49071  0.0  0.0                         0.0          0.0              0.0
49072  0.0  0.0                         0.0          0.0              0.0
49073  0.0  0.0                         0.0          0.0              0.0

       aaaaagh  aaaaah  aaaaahhhh  aaaaargh  aaaaarrrrrrgggggghhhhhh  ...  \
0          0.0     0.0        0.0       0.0                      0.0  ...
1          0.0     0.0        0.0       0.0                      0.0  ...
2          0.0     0.0        0.0       0.0                      0.0  ...
3          0.0     0.0        0.0       0.0                      0.0  ...
4          0.0     0.0        0.0       0.0                      0.0  ...
...        ...     ...        ...       ...                      ...  ...
49069      0.0     0.0        0.0       0.0                      0.0  ...
49070      0.0     0.0        0.0       0.0                      0.0  ...
49071      0.0     0.0        0.0       0.0                      0.0  ...
49072      0.0     0.0        0.0       0.0                      0.0  ...
49073      0.0     0.0        0.0       0.0                      0.0  ...

       überwoman  ünel  ünfaith  üzümcü   ýs  þorleifsson  þór  żmijewski  \
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
2            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
4            0.0   0.0      0.0     0.0  0.0          0.0  0.0        0.0
...          ...   ...      ...     ...  ...          ...  ...        ...
49069        0.0   0.0      0.0     0.0  0.0          1.0  0.0        0.0
49070        0.0   0.0      0.0     0.0  0.0          0.0  1.0        0.0
49071        0.0   0.0      0.0     0.0  0.0          0.0  0.0        1.0
49072        0.0   0.0      0.0     0.0  0.0          0.0  0.0        0.0
49073        0.0   0.0      0.0     0.0  0.0          0.0  0.0        0.0
```

```
       אב
0       0.0    0.0
1       0.0    0.0
2       0.0    0.0
3       0.0    0.0
4       0.0    0.0
...     ...    ...
49069   0.0    0.0
49070   0.0    0.0
49071   0.0    0.0
49072   1.0    0.0
49073   0.0    1.0

[49074 rows x 49074 columns]
```