Rodriguez_Felipe_DSC550_Week3

June 25, 2023

Part 1: Using the TextBlob Sentiment Analyzer

Import the movie review data as a data frame and ensure that the data is loaded properly.

[14]:
```
import pandas as pd
```

[15]:
```
# Creates first dataset from TSV file
data=pd.read_csv('labeledTrainData.tsv',sep='\t')
data['review'] = data['review'].apply(str)
data
```

[15]:

	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…
…	…	…	…
24995	3453_3	0	It seems like more consideration has gone into…
24996	5064_1	0	I don't believe they made this film. Completel…
24997	10905_3	0	Guy is a loser. Can't get girls, needs to buil…
24998	10194_3	0	This 30 minute documentary Buñuel made in the …
24999	8478_8	1	I saw this movie as a child and it broke my he…

[25000 rows x 3 columns]

How many of each positive and negative reviews are there?

[16]:
```
# Gets Value Counts of each positive and negative review
reviews = data['sentiment'].value_counts()
reviews = reviews.rename(index={1:'Positive Reviews', 0:'Negative Reviews'})
# Sets value counts as Dataframe
reviews = pd.DataFrame(reviews)
# Renames Column to Count
reviews = reviews.rename(columns = {'sentiment':'Count'})
reviews
```

[16]:

	Count
Positive Reviews	12500

```
Negative Reviews  12500
```

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.

[17]:
```
from textblob import TextBlob
```

[18]:
```
# Creates New Column in Data that contains text blob sentiment
data['text_blob_sentiment'] = data['review'].apply(lambda review:␣
 ↪TextBlob(review).sentiment)
```

[19]:
```
# Divides Text blob sentiment into two columns
data[['Polarity','Subjectivity']] = pd.DataFrame(data['text_blob_sentiment'].
 ↪tolist(),index=data.index)
```

[20]:
```
# Removes text_blob_sentiment
data = data.drop(columns=['text_blob_sentiment'])
data
```

[20]:
```
            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…
…          …          …                                                 …
24995   3453_3          0  It seems like more consideration has gone into…
24996   5064_1          0  I don't believe they made this film. Completel…
24997  10905_3          0  Guy is a loser. Can't get girls, needs to buil…
24998  10194_3          0  This 30 minute documentary Buñuel made in the …
24999   8478_8          1  I saw this movie as a child and it broke my he…

       Polarity  Subjectivity
0      0.001277      0.606746
1      0.256349      0.531111
2     -0.053941      0.562933
3      0.134753      0.492901
4     -0.024842      0.459818
…           …             …
24995  0.102083      0.542857
24996  0.090813      0.462371
24997  0.145256      0.484103
24998  0.065625      0.504514
24999  0.239295      0.735897

[25000 rows x 5 columns]
```

[21]:
```
# Loop through polarity to count postive versus negative
def polarity_count(polarity):
    if polarity < 0:
        return 0
    else:
        return 1
    return data
```

[22]:
```
# Creates column
data['TextBlobSentiment'] = data['Polarity'].apply(polarity_count)
```

[23]:
```
data
```

[23]:

```
            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…
…          …        …                                                 …
24995   3453_3          0  It seems like more consideration has gone into…
24996   5064_1          0  I don't believe they made this film. Completel…
24997  10905_3          0  Guy is a loser. Can't get girls, needs to buil…
24998  10194_3          0  This 30 minute documentary Buñuel made in the …
24999   8478_8          1  I saw this movie as a child and it broke my he…

       Polarity  Subjectivity  TextBlobSentiment
0      0.001277      0.606746                  1
1      0.256349      0.531111                  1
2     -0.053941      0.562933                  0
3      0.134753      0.492901                  1
4     -0.024842      0.459818                  0
…           …             …                 …
24995  0.102083      0.542857                  1
24996  0.090813      0.462371                  1
24997  0.145256      0.484103                  1
24998  0.065625      0.504514                  1
24999  0.239295      0.735897                  1

[25000 rows x 6 columns]
```

[24]:
```
# Counting number of Positive versus Negative Reviews
positive_polarity = 0
negative_polarity = 0
# Loop to go through each polarity
for TextBlobSentiment in data['TextBlobSentiment']:
    if TextBlobSentiment > 0:
```

```
        positive_polarity += 1
    elif TextBlobSentiment <= 0:
        negative_polarity += 1
    else:
        pass
print("Postive Sentiment Count:", positive_polarity)
print("Negative Sentiment Count:", negative_polarity)
```

```
Postive Sentiment Count: 19017
Negative Sentiment Count: 5983
```

Check the accuracy of this model. Is this model better than random guessing?

[25]:
```
from sklearn.metrics import accuracy_score
```

[26]:
```
orginial_sentiment = data['sentiment']
text_blob_sentiment = data['TextBlobSentiment']
accuracy = accuracy_score(orginial_sentiment, text_blob_sentiment)
print("Accuracy:", accuracy*100,"%")
```

```
Accuracy: 68.524 %
```

For up to five points extra credit, use another prebuilt text sentiment analyzer, e.g., VADER, and repeat steps (3) and (4).

Extra Credit

[27]:
```
pip install vaderSentiment
```

```
Requirement already satisfied: vaderSentiment in
/Users/feliperodriguez/opt/anaconda3/lib/python3.9/site-packages (3.3.2)
Requirement already satisfied: requests in
/Users/feliperodriguez/opt/anaconda3/lib/python3.9/site-packages (from
vaderSentiment) (2.28.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/Users/feliperodriguez/opt/anaconda3/lib/python3.9/site-packages (from
requests->vaderSentiment) (1.26.11)
Requirement already satisfied: certifi>=2017.4.17 in
/Users/feliperodriguez/opt/anaconda3/lib/python3.9/site-packages (from
requests->vaderSentiment) (2022.9.24)
Requirement already satisfied: idna<4,>=2.5 in
/Users/feliperodriguez/opt/anaconda3/lib/python3.9/site-packages (from
requests->vaderSentiment) (3.3)
Requirement already satisfied: charset-normalizer<3,>=2 in
/Users/feliperodriguez/opt/anaconda3/lib/python3.9/site-packages (from
requests->vaderSentiment) (2.0.4)
Note: you may need to restart the kernel to use updated packages.
```

[28]:
```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

[29]:
```
# Creates Analyzer
analyzer = SentimentIntensityAnalyzer()
# Creates Neg column using polarity score from Vader
data['neg'] = [analyzer.polarity_scores(x)['neg'] for x in data['review']]
# Creates Neu column using polarity score from Vader
data['neu'] = [analyzer.polarity_scores(x)['neu'] for x in data['review']]
# Creates Pos column using polarity score from Vader
data['pos'] = [analyzer.polarity_scores(x)['pos'] for x in data['review']]
```

[30]:
```
# Shows data with negative, neutral, and positive score
data
```

[30]:
```
            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…
…          …          …                                                 …
24995   3453_3          0  It seems like more consideration has gone into…
24996   5064_1          0  I don't believe they made this film. Completel…
24997  10905_3          0  Guy is a loser. Can't get girls, needs to buil…
24998  10194_3          0  This 30 minute documentary Buñuel made in the …
24999   8478_8          1  I saw this movie as a child and it broke my he…

       Polarity  Subjectivity  TextBlobSentiment    neg    neu    pos
0      0.001277      0.606746                  1  0.128  0.751  0.121
1      0.256349      0.531111                  1  0.080  0.713  0.207
2     -0.053941      0.562933                  0  0.135  0.809  0.055
3      0.134753      0.492901                  1  0.062  0.884  0.054
4     -0.024842      0.459818                  0  0.122  0.743  0.135
…           …             …                  …      …      …      …
24995  0.102083      0.542857                  1  0.026  0.825  0.149
24996  0.090813      0.462371                  1  0.082  0.680  0.238
24997  0.145256      0.484103                  1  0.053  0.800  0.147
24998  0.065625      0.504514                  1  0.154  0.753  0.093
24999  0.239295      0.735897                  1  0.143  0.729  0.128

[25000 rows x 9 columns]
```

[31]:
```
# Creates 'compound' column that produces overall score
data['compound'] = [analyzer.polarity_scores(x)['compound'] for x in␣
 ↪data['review']]
```

[35]:
```
# Creates column that has each compound value categorized into Negative,␣
 ↪Positive, and Neutral
vader_sentiment = []
```

```
for sentiment in data['compound']:
    # Creates count of one for positive sentiment
    if sentiment >= 0.05 :
        vader_sentiment.append(1)
    # Does not add count for others
    elif sentiment <= - 0.05 :
        vader_sentiment.append(0)
    else :
        vader_sentiment.append(0)

# Adds sentiment scores to Data
data["VaderSentiment"] = vader_sentiment
```

[36]:
```
# Shows data with Sentiment Column
data
```

[36]:

	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…	
…	…	…	…	
24995	3453_3	0	It seems like more consideration has gone into…	
24996	5064_1	0	I don't believe they made this film. Completel…	
24997	10905_3	0	Guy is a loser. Can't get girls, needs to buil…	
24998	10194_3	0	This 30 minute documentary Buñuel made in the …	
24999	8478_8	1	I saw this movie as a child and it broke my he…	

	Polarity	Subjectivity	TextBlobSentiment	neg	neu	pos	\
0	0.001277	0.606746	1	0.128	0.751	0.121	
1	0.256349	0.531111	1	0.080	0.713	0.207	
2	-0.053941	0.562933	0	0.135	0.809	0.055	
3	0.134753	0.492901	1	0.062	0.884	0.054	
4	-0.024842	0.459818	0	0.122	0.743	0.135	
…	…	…	…	…	…	…	
24995	0.102083	0.542857	1	0.026	0.825	0.149	
24996	0.090813	0.462371	1	0.082	0.680	0.238	
24997	0.145256	0.484103	1	0.053	0.800	0.147	
24998	0.065625	0.504514	1	0.154	0.753	0.093	
24999	0.239295	0.735897	1	0.143	0.729	0.128	

	compound	VaderSentiment
0	-0.8879	0
1	0.9736	1
2	-0.9883	0
3	-0.1202	0

```
4         0.6115                1
…          …                …
24995     0.8750                1
24996     0.9861                1
24997     0.9252                1
24998    -0.9598                0
24999     0.2934                1

[25000 rows x 11 columns]
```

[38]:
```
# Counting number of Positive, Negative, and Neutral Reviews from Vader
positive_polarity_vader = 0
negative_polarity_vader = 0
neutral_polarity_vader = 0
# Loop to go through each vader sentiment
for polarity in data['VaderSentiment']:
    # positive polarity count
    if polarity == 1:
        positive_polarity_vader += 1
    # negative polarity count
    else:
        negative_polarity_vader += 1
print("Postive Sentiment Count:", positive_polarity_vader)
print("Negative Sentiment Count:", negative_polarity_vader)
```

```
Postive Sentiment Count: 16507
Negative Sentiment Count: 8493
```

Check the accuracy of this model. Is this model better than random guessing?

[39]:
```
# Gets Vadersentiment data from dataframe
vader_blob_sentiment = data['VaderSentiment']
# Calculates accuracy
accuracy = accuracy_score(orginial_sentiment, vader_blob_sentiment)
print("Accuracy:", accuracy*100,"%")
```

```
Accuracy: 69.556 %
```

Part 2: Prepping Text for a Custom Model

Convert all text to lowercase letters.

[40]:
```
# Imports necessary libraries
import unicodedata
import sys
```

[73]:
```
# Reads data in
data2=pd.read_csv('labeledTrainData.tsv',sep='\t')
# Makes review into string
```

```
data2['review'] = data2['review'].apply(str)
data2
```

[73]:

	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…
…	…	…	…
24995	3453_3	0	It seems like more consideration has gone into…
24996	5064_1	0	I don't believe they made this film. Completel…
24997	10905_3	0	Guy is a loser. Can't get girls, needs to buil…
24998	10194_3	0	This 30 minute documentary Buñuel made in the …
24999	8478_8	1	I saw this movie as a child and it broke my he…

[25000 rows x 3 columns]

[80]:
```
# Creates decapitilzer for strings
def decapitalizer(string: str) -> str:
    return string.lower()
```

[81]:
```
# Applies decapitilizer
data2['review'] = data2['review'].apply(decapitalizer)
```

[44]:
```
data2
```

[44]:

	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…
…	…	…	…
24995	3453_3	0	it seems like more consideration has gone into…
24996	5064_1	0	i don't believe they made this film. completel…
24997	10905_3	0	guy is a loser. can't get girls, needs to buil…
24998	10194_3	0	this 30 minute documentary buñuel made in the …
24999	8478_8	1	i saw this movie as a child and it broke my he…

[25000 rows x 3 columns]

Remove punctuation and special characters from the text.

[45]:
```
# Creates list of punctuations
punctuation = dict.fromkeys(i for i in range(sys.maxunicode)
                            if unicodedata.category(chr(i)).startswith('P'))
```

[82]:
```
# Removes punctuations from review
data2['review'] = [string.translate(punctuation) for string in data2.review]
data2
```

[82]:

	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 hines…
2	7759_3	0	the film starts with a manager nicholas bell g…
3	3630_4	0	it must be assumed that those who praised this…
4	9495_8	1	superbly trashy and wondrously unpretentious 8…
…	…	…	…
24995	3453_3	0	it seems like more consideration has gone into…
24996	5064_1	0	i dont believe they made this film completely …
24997	10905_3	0	guy is a loser cant get girls needs to build u…
24998	10194_3	0	this 30 minute documentary buñuel made in the …
24999	8478_8	1	i saw this movie as a child and it broke my he…

[25000 rows x 3 columns]

Remove stop words.

[47]:
```
# Import libraries to remove stopwords
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

[48]:
```
import nltk
```

[49]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/feliperodriguez/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

[49]: True

[50]:
```
# Creates list of stopword
stop_words = stopwords.words('english')
```

[51]:
```
# Splits review into tokens
data2['review'] = data2['review'].str.split()
data2
```

[51]:

	id	sentiment	review
0	5814_8	1	[with, all, this, stuff, going, down, at, the,…
1	2381_9	1	[the, classic, war, of, the, worlds, by, timot…
2	7759_3	0	[the, film, starts, with, a, manager, nicholas…
3	3630_4	0	[it, must, be, assumed, that, those, who, prai…
4	9495_8	1	[superbly, trashy, and, wondrously, unpretenti…

```
…         …         …                                                  …
24995   3453_3         0  [it, seems, like, more, consideration, has, go…
24996   5064_1         0  [i, dont, believe, they, made, this, film, com…
24997  10905_3         0  [guy, is, a, loser, cant, get, girls, needs, t…
24998  10194_3         0  [this, 30, minute, documentary, buñuel, made, …
24999   8478_8         1  [i, saw, this, movie, as, a, child, and, it, b…

[25000 rows x 3 columns]
```

[53]:
```
# Removed stop words from the column review
data2['review'] = data2['review'].apply(lambda x: [word for word in x if word␣
 ↪not in stop_words])
```

[54]:
```
data2
```

[54]:
```
             id  sentiment                                             review
0       5814_8          1  [stuff, going, moment, mj, ive, started, liste…
1       2381_9          1  [classic, war, worlds, timothy, hines, enterta…
2       7759_3          0  [film, starts, manager, nicholas, bell, giving…
3       3630_4          0  [must, assumed, praised, film, greatest, filme…
4       9495_8          1  [superbly, trashy, wondrously, unpretentious, …
…         …         …                                                  …
24995   3453_3          0  [seems, like, consideration, gone, imdb, revie…
24996   5064_1          0  [dont, believe, made, film, completely, unnece…
24997  10905_3          0  [guy, loser, cant, get, girls, needs, build, p…
24998  10194_3          0  [30, minute, documentary, buñuel, made, early,…
24999   8478_8          1  [saw, movie, child, broke, heart, story, unfin…

[25000 rows x 3 columns]
```

Apply NLTK's PorterStemmer.

[55]:
```
# Import libraries
from nltk.stem.porter import PorterStemmer
```

[56]:
```
# Creates porter
porter = PorterStemmer()
```

[57]:
```
# Applies stem to column review
data2['review'] = data2['review'].apply(lambda x: [porter.stem(word) for word␣
 ↪in x])
data2
```

[57]:
```
             id  sentiment                                             review
0       5814_8          1  [stuff, go, moment, mj, ive, start, listen, mu…
1       2381_9          1  [classic, war, world, timothi, hine, entertain…
2       7759_3          0  [film, start, manag, nichola, bell, give, welc…
```

```
3       3630_4          0  [must, assum, prais, film, greatest, film, ope…
4       9495_8          1  [superbl, trashi, wondrous, unpretenti, 80, ex…
…          …          …                                                 …
24995   3453_3          0  [seem, like, consider, gone, imdb, review, fil…
24996   5064_1          0  [dont, believ, made, film, complet, unnecessar…
24997  10905_3          0  [guy, loser, cant, get, girl, need, build, pic…
24998  10194_3          0  [30, minut, documentari, buñuel, made, earli, …
24999   8478_8          1  [saw, movi, child, broke, heart, stori, unfini…

[25000 rows x 3 columns]
```

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.

[58]:
```
# Contains only the reviews
reviews_only_final = data2['review']
```

[59]:
```
# import libraries
from sklearn.feature_extraction.text import CountVectorizer
```

[60]:
```
# Creates vectorizer
vectorizer = CountVectorizer(analyzer=lambda x: x)
# Creates bag of words
bag_of_words = vectorizer.fit_transform(reviews_only_final)
```

[61]:
```
# Creates array of bag of words
bag_of_words.toarray()
```

[61]:
```
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]])
```

[62]:
```
# Displays size of bag of words
print(bag_of_words.shape)
```

```
(25000, 97771)
```

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.

[63]:
```
# Import libraries
from sklearn.feature_extraction.text import TfidfVectorizer
```

[64]:
```
# Creates Tfid Vectorizer
tfidf = TfidfVectorizer(analyzer=lambda x: x)
# Creates feature matrix
feature_matrix = tfidf.fit_transform(reviews_only_final)
```

[65]:
```
# Creates array of feature matrix
feature_matrix.toarray()
```

[65]:
```
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.]])
```

[66]:
```
# Displays size of feature matrix
print(feature_matrix.shape)
```

```
(25000, 97771)
```