Assignment_4_2_Raghuwanshi_Prashant_DSC550

September 25, 2021

Assignment: 4.2 Exercise: Sentiment Analysis

Name: Prashant Raghuwanshi

Date: 9/25/2021

Course: DSC550-T301 Data Mining (2221-1)

[97]:
```
# Import Libraries
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np
import pandas as pd
from textblob import TextBlob
import matplotlib.pyplot as plt
```

Load the data file DailyComments.csv from the Week 4 Data Files into a data frame.

[42]:
```
# Import the sample data file to dataframe
dailycomment_df = pd.read_csv (r'C:
 ↪\Users\dell\Documents\Machine_learning_assigments\DailyComments.csv')
# Show the dataframe data values & copy df to df1
dailycomment_df1 = dailycomment_df
dailycomment_df1
```

[42]:

	Day of Week	comments
0	Monday	Hello, how are you?
1	Tuesday	Today is a good day!
2	Wednesday	It's my birthday so it's a really special day!
3	Thursday	Today is neither a good day or a bad day!
4	Friday	I'm having a bad day.
5	Saturday	There' s nothing special happening today.
6	Sunday	Today is a SUPER good day!

Identify a scheme to categorize each comment as positive or negative. You can devise your own scheme or find a commonly used scheme to perform this sentiment analysis. However you decide to do this, make sure to explain the scheme you decide to use.

[52]:
```
# comment schema has been identified for sentiment analysis
# splitting the dataframe with comment schema and storing it in new corpus df
corpus = dailycomment_df['comments']
```

Implement your sentiment analysis with code and display the results. Note: DailyComments.csv is a purposely small file, so you will be able to clearly see why the results are what they are.

Method 1 - using sklearn library

[58]:
```
# Convert a collection of text documents to a matrix of token counts.
# In the script above we use the CountVectorizer class from the sklearn.
 ↪feature_extraction.text module to create a document-term matrix.
# We specify to only include those words that appear in less than 80% of the␣
 ↪document and appear in at least 2 documents.
# We also remove all the stop words as they do not really contribute to topic␣
 ↪modeling
vectorizer1 = CountVectorizer(max_df=0.8, min_df=2, stop_words='english')
X1 = vectorizer1.fit_transform(corpus)
print( X1.toarray())
```

```
[[0 0 0 0 0]
 [0 1 1 0 1]
 [0 1 0 1 0]
 [1 2 1 0 1]
 [1 1 0 0 0]
 [0 0 0 1 1]
 [0 1 1 0 1]]
```

[56]:
```
#fetch words from our vocabulary
print(vectorizer.get_feature_names())
```

```
['bad', 'day', 'good', 'special', 'today']
```

[61]:
```
#check for positive words and negative words
dailycomment_df['positive1'] = dailycomment_df.comments.str.count('good')
dailycomment_df['positive2']= dailycomment_df.comments.str.count('special')
dailycomment_df['negative'] = dailycomment_df.comments.str.count('bad')
dailycomment_df['TotScore'] = dailycomment_df.positive1 + df.positive2 - df.
 ↪negative

print("")
print(dailycomment_df)

Z = sum(dailycomment_df['TotScore'])
print("")
print("Overall Score:  ",Z)
```

```
  Day of Week                                        comments  positive1  \
0      Monday                           Hello, how are you?          0
1     Tuesday                         Today is a good day!          1
2   Wednesday  It's my birthday so it's a really special day!          0
3    Thursday       Today is neither a good day or a bad day!          1
```

```
4      Friday                              I'm having a bad day.          0
5    Saturday       There' s nothing special happening today.          0
6      Sunday                          Today is a SUPER good day!          1

   positive2  negative  TotScore
0          0         0         0
1          0         0         1
2          1         0         1
3          0         1         0
4          0         1        -1
5          1         0         1
6          0         0         1

Overall Score:   3
```

Method 2 by using TextBlob librabry

[16]:
```
def getSubjectivity(text):
   return TextBlob(text).sentiment.subjectivity
def getPolarity(text):
   return TextBlob(text).sentiment.polarity
```

[34]:
```
dailycomment_df1['TextBlob_Polarity'] = corpus.apply(getPolarity)
```

[36]:
```
dailycomment_df1['TextBlob_Subjectivity'] = corpus.apply(getSubjectivity)
```

[37]:
```
dailycomment_df1
```

[37]:
```
  Day of Week                                        comments  \
0      Monday                              Hello, how are you?
1     Tuesday                             Today is a good day!
2   Wednesday  It's my birthday so it's a really special day!
3    Thursday       Today is neither a good day or a bad day!
4      Friday                              I'm having a bad day.
5    Saturday       There' s nothing special happening today.
6      Sunday                          Today is a SUPER good day!

   TextBlob_Polarity  TextBlob_Subjectivity
0           0.000000               0.000000
1           0.875000               0.600000
2           0.446429               0.571429
3          -0.087500               0.633333
4          -0.700000               0.666667
5           0.357143               0.571429
6           0.604167               0.633333
```

[38]:
```
def getAnalysis(score):
  if score < 0:
```

```
    return 'Negative'
  elif score == 0:
    return 'Neutral'
  else:
    return 'Positive'
```

[39]:
```
dailycomment_df1['TextBlob_Analysis'] = dailycomment_df1['TextBlob_Polarity'].
↪apply(getAnalysis)
```

[40]:
```
dailycomment_df1
```

[40]:
```
  Day of Week                                        comments  \
0      Monday                            Hello, how are you?
1     Tuesday                           Today is a good day!
2   Wednesday  It's my birthday so it's a really special day!
3    Thursday       Today is neither a good day or a bad day!
4      Friday                           I'm having a bad day.
5    Saturday       There' s nothing special happening today.
6      Sunday                        Today is a SUPER good day!

   TextBlob_Polarity  TextBlob_Subjectivity TextBlob_Analysis
0           0.000000               0.000000           Neutral
1           0.875000               0.600000          Positive
2           0.446429               0.571429          Positive
3          -0.087500               0.633333          Negative
4          -0.700000               0.666667          Negative
5           0.357143               0.571429          Positive
6           0.604167               0.633333          Positive
```

For up to 5% extra credit, find another set of comments, e.g., some tweets, and perform the same sentiment analysis.

[64]:
```
# this analyis is to find the differnce betwen tweeter provied sentiment vs␣
↪textblod sentiments
# Import the sample airline tweet data file to dataframe
airlinetweet_df = pd.read_csv (r'C:
↪\Users\dell\Documents\Machine_learning_assigments\airlinetweet.csv')
# Show the dataframe data values & copy df to df1
airlinetweet_df.head()
```

[64]:
```
     _unit_id  _golden _unit_state  _trusted_judgments _last_judgment_at  \
0  681448150    False   finalized                   3      2/25/15 5:24
1  681448153    False   finalized                   3      2/25/15 1:53
2  681448156    False   finalized                   3     2/25/15 10:01
3  681448158    False   finalized                   3      2/25/15 3:05
4  681448159    False   finalized                   3      2/25/15 5:50

  airline_sentiment  airline_sentiment:confidence negativereason  \
```

```
0           neutral                        1.0000            NaN
1          positive                        0.3486            NaN
2           neutral                        0.6837            NaN
3          negative                        1.0000     Bad Flight
4          negative                        1.0000     Can't Tell

   negativereason:confidence         airline airline_sentiment_gold  \
0                        NaN  Virgin America                    NaN
1                     0.0000  Virgin America                    NaN
2                        NaN  Virgin America                    NaN
3                     0.7033  Virgin America                    NaN
4                     1.0000  Virgin America                    NaN

         name negativereason_gold  retweet_count  \
0     cairdin                 NaN              0
1    jnardino                 NaN              0
2  yvonnalynn                 NaN              0
3    jnardino                 NaN              0
4    jnardino                 NaN              0

                                                text tweet_coord  \
0                @VirginAmerica What @dhepburn said.         NaN
1  @VirginAmerica plus you've added commercials t…         NaN
2  @VirginAmerica I didn't today… Must mean I n…         NaN
3  @VirginAmerica it's really aggressive to blast…         NaN
4  @VirginAmerica and it's a really big bad thing…         NaN

   tweet_created      tweet_id tweet_location               user_timezone
0  2/24/15 11:35  5.703060e+17            NaN  Eastern Time (US & Canada)
1  2/24/15 11:15  5.703010e+17            NaN  Pacific Time (US & Canada)
2  2/24/15 11:15  5.703010e+17      Lets Play  Central Time (US & Canada)
3  2/24/15 11:15  5.703010e+17            NaN  Pacific Time (US & Canada)
4  2/24/15 11:14  5.703010e+17            NaN  Pacific Time (US & Canada)
```

[66]:
```
# selectng schema and loading to new df
airlinetweet_df1 = airlinetweet_df['text']
airlinetweet_df1
```

[66]:
```
0                     @VirginAmerica What @dhepburn said.
1        @VirginAmerica plus you've added commercials t…
2        @VirginAmerica I didn't today… Must mean I n…
3        @VirginAmerica it's really aggressive to blast…
4        @VirginAmerica and it's a really big bad thing…
                              …
14635    @AmericanAir thank you we got on a different f…
14636    @AmericanAir leaving over 20 minutes Late Flig…
14637    @AmericanAir Please bring American Airlines to…
```

```
14638    @AmericanAir you have my money, you change my …
14639    @AmericanAir we have 8 ppl so we need 2 know h…
Name: text, Length: 14640, dtype: object
```

[74]:
```
# data cleansing
# creating the list of meaningless words
unmeaningful = ['i', 'you', 'me', 'to', 'the', 'a', 'my', 'is', 'in', 'and',␣
↪'for', 'on', 'of',
                'your', 'so', 'was', 'have', 'it', 'at', 'with', 'that',␣
↪'from', 'do', 'get',
                'but', 'this', 'can', 'just', 'they', 'we', 'are', 'an', 'be',␣
↪"i'm", 'will',
                'if', 'had', 'our', 'about', 'there', 'has', 'been', '-', 'by',␣
↪'like', 'or',
                'as', 'he', 'she', 'it', 'us', 'has' ,"i've", "it's", "don't",␣
↪'would', 'am',
               'flight', 'customer', 'any', 'very', "didn't", "you've",␣
↪'thing', 'take',
               'other', 'u', '', ' ', '.']
```

[75]:
```
def clean_text(str_in):
    """Remove special characters, @airline/username, empty string and
    """
    res = ""
    str_in = str_in.lower()
    str_arr = str_in.split(' ')
    for word in str_arr:
        # make all words into lower case
        word = word.lower()
        # remove not useful words from the original text
        if '@' in word or word == '' or word[:1] == '&':
            continue
        if word.lower() in unmeaningful:
            continue
        if word.isnumeric():
            continue
        res = res + " " + word
    return res
```

[76]:
```
airlinetweet_df1 = airlinetweet_df1.apply(clean_text)
print(airlinetweet_df1.head(5))
```

```
0                                          what said.
1          plus added commercials experience… tacky.
2               today… must mean need another trip!
3    really aggressive blast obnoxious "entertainm…
4                                    really big bad
```

```
Name: text, dtype: object
```

[85]:
```
compare_sentiments_df = pd.DataFrame(airlinetweet_df, columns = ['text',␣
↪'airline_sentiment'])
compare_sentiments_df['TextBlob_Polarity'] = airlinetweet_df1.apply(getPolarity)
compare_sentiments_df['TextBlob_Subjectivity'] = airlinetweet_df1.
↪apply(getSubjectivity)
compare_sentiments_df['TextBlob_Analysis'] =␣
↪compare_sentiments_df['TextBlob_Polarity'].apply(getAnalysis)
```

[86]:
```
compare_sentiments_df
```

[86]:
```
                                                  text airline_sentiment  \
0                      @VirginAmerica What @dhepburn said.           neutral
1        @VirginAmerica plus you've added commercials t…          positive
2        @VirginAmerica I didn't today… Must mean I n…           neutral
3        @VirginAmerica it's really aggressive to blast…          negative
4        @VirginAmerica and it's a really big bad thing…          negative
…                                                      …                 …
14635    @AmericanAir thank you we got on a different f…          positive
14636    @AmericanAir leaving over 20 minutes Late Flig…          negative
14637    @AmericanAir Please bring American Airlines to…           neutral
14638    @AmericanAir you have my money, you change my …          negative
14639    @AmericanAir we have 8 ppl so we need 2 know h…           neutral

       TextBlob_Polarity  TextBlob_Subjectivity TextBlob_Analysis
0               0.000000               0.000000          Neutral
1               0.000000               0.000000          Neutral
2              -0.390625               0.687500         Negative
3               0.006250               0.350000         Positive
4              -0.350000               0.383333         Negative
…                    …                      …                …
14635           0.000000               0.600000          Neutral
14636          -0.300000               0.600000         Negative
14637           0.000000               0.000000          Neutral
14638           0.000000               0.000000          Neutral
14639           0.166667               0.166667         Positive

[14640 rows x 5 columns]
```

[93]:
```
total_negative = (compare_sentiments_df['airline_sentiment']=='negative').sum()
total_positive = (compare_sentiments_df['airline_sentiment'] == 'positive').
↪sum()
total_neutral = (compare_sentiments_df['airline_sentiment'] == 'neutral').sum()
new_total_negative = (compare_sentiments_df['TextBlob_Analysis']=='Neutral').
↪sum()
```

```
new_total_positive = (compare_sentiments_df['TextBlob_Analysis']=='Positive').
 ↪sum()
new_total_neutral = (compare_sentiments_df['TextBlob_Analysis']=='Negative').
 ↪sum()
```

[100]:
```
sentiments = compare_sentiments_df['airline_sentiment'].unique()
#here we know there are 3 types only
values = [total_neutral, total_positive, total_negative] #
values1 = [new_total_negative, new_total_positive, new_total_neutral]
dictionary1 = {'Airline Sentiment':sentiments, 'Old Count':values, 'New Count':
 ↪values1}
dfSentimentCount = pd.DataFrame(dictionary1)
dfSentimentCount.head(3)
```

[100]:

	Airline Sentiment	Old Count	New Count
0	neutral	3099	5599
1	positive	2363	5446
2	negative	9178	3595

[ ]:
```

```