course_2_project

Due: 2018-11-25 01:34:00

Description: Final Project for Course 2 - Sentiment Classifier

Score: 0 of 4 = 0.0%

Questions

Not yet graded

We have provided some synthetic (fake, semi-randomly generated) twitter data in a csv file named project_twitter_data.csv which has the text of a tweet, the number of retweets of that tweet, and the number of replies to that tweet. We have also words that express positive sentiment and negative sentiment, in the files *positive_words.txt* and *negative_words.txt*.

Your task is to build a sentiment classifier, which will detect how positive or negative each tweet is. You will create a csv file, which contains columns for the Number of Retweets, Number of Replies, Positive Score (which is how many happy words are in the tweet), Negative Score (which is how many angry words are in the tweet), and the Net Score for each tweet. At the end, you upload the csv file to Excel or Google Sheets, and produce a graph of the Net Score vs Number of Retweets.

To start, define a function called `strip_punctuation` which takes one parameter, a string which represents a word, and removes characters considered punctuation from everywhere in the word. (Hint: remember the *.replace()* method for strings.)

Save & Run | 11/9/2020, 11:27:26 PM - 3 of 3 | Show in CodeLens

```
1 punctuation_chars = ["'", '"', ",", ".", "!", ":", ";", '#', '@']
2
3 def strip_punctuation(x):
4     for c in punctuation_chars:
5         x = x.replace(c, '')
6     return x
7
8
9
```

ActiveCode (assess_ac_18_1_1_1)

Result	Actual Value	Expected Value	Notes

Pass	'Amazing'	'Amazing'	Testing that the correct output is returned when #Amazing is provided as input.
Pass	'wow'	'wow'	Testing that the correct output is returned when wow! is provided as input.
Pass	'incredible'	'incredible'	Testing that the correct output is returned when #in.cred..ible! is provided as input.
Pass	'wonderful'	'wonderful'	Testing that the correct output is returned when wonderful is provided as input.

You passed: 100.0% of the tests

Not yet graded

Next, copy in your strip_punctuation function and define a function called `get_pos` which takes one parameter, a string which represents one or more sentences, and calculates how many words in the string are considered positive words. Use the list, `positive_words` to determine what words will count as positive. The function should return a positive integer - how many occurrences there are of positive words in the text. Note that all of the words in `positive_words` are lower cased, so you'll need to convert all the words in the input string to lower case as well.

Save & Run 11/9/2020, 11:31:36 PM - 4 of 4 Show in CodeLens

```
5  with open("positive_words.txt") as pos_f:
6      for lin in pos_f:
7          if lin[0] != ';' and lin[0] != '\n':
8              positive_words.append(lin.strip())
9
10 def strip_punctuation(x):
11     for c in punctuation_chars:
12         x = x.replace(c, '')
13     return x
14
15 def get_pos(x):
16     return len(filter(lambda x: (strip_punctuation(x) in positive_words), x.lower().spl
17
18
```

ActiveCode (assess_ac_18_1_1_2)

Result	Actual Value	Expected Value	Notes
Pass	2	2	Testing "what a truly wonderful day it is today! #incredible"
Pass	2	2	what a truly Wonderful day it is today! #incredible
Pass	1	1	what a truly wonderful day it is today!
Pass	0	0	the weather is what it is.
Pass	0	0	The weather truely is abnormal - it's october and already snowing!

You passed: 100.0% of the tests

Not yet graded

Next, copy in your strip_punctuation function and define a function called `get_neg` which takes one parameter, a string which represents one or more sentences, and calculates how many words in the string are considered negative words. Use the list, `negative_words` to determine what words will count as negative. The function should return a positive integer - how many occurrences there are of negative words in the text. Note that all of the words in `negative_words` are lower cased, so you'll need to convert all the words in the input string to lower case as well.

Save & Run 11/9/2020, 11:32:17 PM - 2 of 2 Show in CodeLens

```
4  negative_words = []
5  with open("negative_words.txt") as pos_f:
6      for lin in pos_f:
7          if lin[0] != ';' and lin[0] != '\n':
8              negative_words.append(lin.strip())
9
10 def strip_punctuation(x):
11     for c in punctuation_chars:
12         x = x.replace(c, '')
13     return x
14
15 def get_neg(x):
16     return len(filter(lambda x: (strip_punctuation(x) in negative_words), x.lower().spl
17
```

ActiveCode (assess_ac_18_1_1_3)

Result	Actual Value	Expected Value	Notes
Pass	0	0	what a truly wonderful day it is today! #incredible
Pass	1	1	The weather truely is abnormal - it's october and already snowing!
Pass	2	2	their departure was rather abrupt. However, it was amusing how aloof they had been.
Pass	0	0	the weather is what it is.

You passed: 100.0% of the tests

Not yet graded

Finally, copy in your previous functions and write code that opens the file `project_twitter_data.csv` which has the fake generated twitter data (the text of a tweet, the number of retweets of that tweet, and the number of replies to that tweet). Your task is to build a sentiment classifier, which will detect how positive or negative each tweet is. Copy the code from the code windows above, and put that in the top of this code window. Now, you will write code to create a csv file called `resulting_data.csv` , which contains the Number of Retweets, Number of Replies, Positive Score (which is how many happy words are in the tweet), Negative Score (which is how many angry words are in the tweet), and the Net Score (how positive or negative the text is overall) for each tweet. The file should have those headers in that order. Remember that there is another component to this project. You will upload the csv file to Excel or Google Sheets and produce a graph of the Net Score vs Number of Retweets. Check Coursera for that portion of the assignment, if you're accessing this textbook from Coursera.

Save & Run 11/9/2020, 11:48:38 PM - 4 of 4 Show in CodeLens

```
1  def strip_punctuation(x):
2      for c in punctuation_chars:
3          x = x.replace(c, '')
4      return x
5
6  def get_pos(x):
7      return len(filter(lambda x: (strip_punctuation(x) in positive_words), x.lower().spli
8
9  def get_neg(x):
10     return len(filter(lambda x: (strip_punctuation(x) in negative_words), x.lower().spli
11
12 punctuation_chars = ["'", '"', ",", ".", "!", ":", ";", '#', '@']
13 # lists of words to use
14 positive_words = []
15
```

ActiveCode (assess_ac_18_1_1_4)

Result	Actual Value	Expected Value	Notes
Pass	'Numbe...core\n'	'Numbe...core\n'	checking that the headers are set correctly.
Pass	'19'	'19'	checking that the value for a particular cell matches.
Pass	'-3'	'-3'	checking that the value of the net score is correct for a particular cell.
Pass	20	20	checking that the file has the correct number of rows.
Pass	5	5	checking that the file has the correct number of columns.

Expand Differences

You passed: 100.0% of the tests

Data file: `resulting_data.csv`

```
0, 0, 0, 1, -1
0, 0, 1, 0, 1
47, 0, 2, 0, 2
2, 1, 1, 0, 1
0, 2, 1, 0, 1
0, 0, 2, 1, 1
4, 6, 3, 0, 3
19, 0, 3, 1, 2
0, 0, 1, 1, 0
```

Score Me