lab4_positivity

February 1, 2024

1 Lab 4: Find the most positive subreddit

In this lab, you're going to combine what you've learned about loops, conditions, and sentiment analysis to find the most positive subreddit you can!

First, run the code to get the sentiment analyzer running:

[1]:
```
import nltk
nltk.download(["vader_lexicon"])
from nltk.sentiment import SentimentIntensityAnalyzer
sia = SentimentIntensityAnalyzer()
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /home/jovyan/nltk_data…
[nltk_data]   Package vader_lexicon is already up-to-date!
```

Remember, this is the code you can use to analyze the sentiment of a sentence:

[2]:
```
sentence = "I love INFO 103!!! It's the best class I've ever taken!!!"
sia.polarity_scores(sentence)["compound"]
```

[2]:
```
0.8902
```

Now, run the normal praw steps to access Reddit:

[3]:
```
import praw
```

[4]:
```
%run reddit_keys.py
```

[5]:
```
reddit = praw.Reddit(
    username=username, password=password,
    client_id=client_id, client_secret=client_secret,
    user_agent="a custom python script for user /" + str(username)
)
```

```
Version 7.7.0 of praw is outdated. Version 7.7.1 was released Tuesday July 11,
2023.
```

The code below will grab the 10 most "hot" posts from the "news" subreddit, and format those posts into a list called "submissions_list." Run it to store posts from your first subreddit.

[6]:
```
# Look up the subreddit "news", then find the "hot" list, getting up to 10 submission
submissions = reddit.subreddit("news").hot(limit=10)

submissions_list = list(submissions)
```

Now, modify the code below so that, instead of just printing the titles of the posts, you are calculating the average sentiment of each post title, and storing it in a variable. Hint: To do this, you'll need some variables to keep track of the number of posts and the total sentiment as you loop through each submission, so you can find the average at the end.

[7]:
```
#add some variables here to keep track of submissions and sentiment
num_submissions = 0
total_news_sentiment = 0

for submission in submissions_list:
    #modify this code to calculate sentiment (you can remove the print statements if you want)
    submission_sentiment = sia.polarity_scores(submission.title)["compound"]

    num_submissions += 1
    total_news_sentiment = total_news_sentiment +  submission_sentiment

    print("Sentiment: " + str(submission_sentiment))
    print("   Submission Title: " + submission.title)
    print()

#modify this code to calculate the average sentiment
average_news_sentiment = total_news_sentiment / num_submissions
display("The average sentiment was " + str(average_news_sentiment))
```

```
Sentiment: -0.5574
   Submission Title: 17-year-old alleged 'serial swatter' charged after police say he made threats throughout the country

Sentiment: -0.0772
   Submission Title: US military stockpiling supplies in Australia in readiness for any confrontation with China

Sentiment: -0.2263
   Submission Title: Judge dismisses Disney's lawsuit against Florida Gov. Ron DeSantis and his allies

Sentiment: 0.0
   Submission Title: Texas AG Paxton sues five cities, including Austin and San Marcos, over marijuana policies

Sentiment: 0.5538
```

```
   Submission Title: European Union agrees on a new 50 billion-euro aid package
for Ukraine despite Hungary's veto threat

Sentiment: -0.7717
   Submission Title: Oregon leaders declare 90-day state of emergency in
downtown Portland to address fentanyl crisis

Sentiment: 0.296
   Submission Title: US approves plan to strike Iranian targets in Syria and
Iraq, officials say

Sentiment: 0.4767
   Submission Title: Israeli intelligence report claims four UNRWA staff in Gaza
involved in Hamas kidnappings | World News

Sentiment: 0.2023
   Submission Title: Secret US spying program targeted top Venezuelan officials,
flouting international law

Sentiment: 0.2023
   Submission Title: Federal Reserve signals that interest rate cuts aren't
imminent


'The average sentiment was 0.009849999999999987'
```

Now, try to repeat the steps from above, but find the average sentiment of at least 3-4 different subreddits. Make sure to store them all in different variables.

[8]:
```
submissions = reddit.subreddit("cuteanimals").hot(limit=10)
submissions_list = list(submissions)

num_submissions = 0
total_cute_sentiment = 0

for submission in submissions_list:
    #modify this code to calculate sentiment (you can remove the print␣
 ↪statements if you want)
    submission_sentiment = sia.polarity_scores(submission.title)["compound"]

    num_submissions += 1
    total_cute_sentiment = total_cute_sentiment +  submission_sentiment

    print("Sentiment: " + str(submission_sentiment))
    print("   Submission Title: " + submission.title)
    print()

#modify this code to calculate the average sentiment
```

```
average_cute_sentiment = total_cute_sentiment / num_submissions
display("The average cute sentiment was " + str(average_cute_sentiment))
```

```
Sentiment: 0.0
   Submission Title: New updates in our community settings!

Sentiment: 0.5859
   Submission Title: Cutest dog ever

Sentiment: 0.4588
   Submission Title: This Cute Tiger

Sentiment: 0.0
   Submission Title: When asked if I have children I show them a picture of this
sleepy baby

Sentiment: 0.5574
   Submission Title: two different moods.. laugh out loud

Sentiment: 0.4588
   Submission Title: This Cute Squirrel

Sentiment: 0.4588
   Submission Title: This Cute Rabbit

Sentiment: 0.6486
   Submission Title: Cute bunny curious about the camera

Sentiment: 0.4588
   Submission Title: This Cute Black Swan

Sentiment: 0.4588
   Submission Title: This Cute Duck


'The average cute sentiment was 0.40859000000000006'
```

[10]:
```
submissions = reddit.subreddit("shoes").hot(limit=10)
submissions_list = list(submissions)

num_submissions = 0
total_shoes_sentiment = 0

for submission in submissions_list:
    #modify this code to calculate sentiment (you can remove the print␣
 ↪statements if you want)
    submission_sentiment = sia.polarity_scores(submission.title)["compound"]
```

```
    num_submissions += 1
    total_shoes_sentiment = total_shoes_sentiment +  submission_sentiment

    print("Sentiment: " + str(submission_sentiment))
    print("   Submission Title: " + submission.title)
    print()

#modify this code to calculate the average sentiment
average_shoes_sentiment = total_shoes_sentiment / num_submissions
display("The average sentiment was " + str(average_shoes_sentiment))
```

```
Sentiment: 0.4939
   Submission Title: Friendly reminder before you post

Sentiment: 0.4588
   Submission Title: One of my favorite pairs

Sentiment: 0.3612
   Submission Title: Are these too small, or is this just how heels are supposed
to fit?

Sentiment: 0.0
   Submission Title: New for musical shows. What do you think?

Sentiment: 0.5707
   Submission Title: Check out these beauties!

Sentiment: 0.0
   Submission Title: I'm looking for some shorts.

Sentiment: 0.2023
   Submission Title: converse con, mid top, red thread

Sentiment: 0.0
   Submission Title: Newest delivery

Sentiment: 0.4019
   Submission Title: Help me decide which ones I should get (rank them all)

Sentiment: 0.4404
   Submission Title: Which shoe looks better?


'The average sentiment was 0.29291999999999996'
```

[11]:
```
submissions = reddit.subreddit("books").hot(limit=10)
submissions_list = list(submissions)
```

```
num_submissions = 0
total_books_sentiment = 0

for submission in submissions_list:
    #modify this code to calculate sentiment (you can remove the print␣
 ↪statements if you want)
    submission_sentiment = sia.polarity_scores(submission.title)["compound"]

    num_submissions += 1
    total_books_sentiment = total_books_sentiment +  submission_sentiment

    print("Sentiment: " + str(submission_sentiment))
    print("   Submission Title: " + submission.title)
    print()

#modify this code to calculate the average sentiment
average_books_sentiment = total_books_sentiment / num_submissions
display("The average sentiment was " + str(average_books_sentiment))
```

```
Sentiment: 0.0
   Submission Title: What Books did You Start or Finish Reading this Week?:
January 29, 2024

Sentiment: 0.3182
   Submission Title: ‘I should not have written ‘A Clockwork Orange’’: How
Anthony Burgess came to disown his own novel

Sentiment: -0.4754
   Submission Title: What's a line that absolutely crushed your soul

Sentiment: -0.4824
   Submission Title: RANT: it ends with us

Sentiment: 0.2263
   Submission Title: Jurassic Park First Impressions

Sentiment: 0.5829
   Submission Title: Reading "The Body Keeps Score" has brought me so much peace
(TW: SA)

Sentiment: 0.0
   Submission Title: Can someone - maybe a British person - explain this line
from White Teeth by Zadie Smith?

Sentiment: 0.0
   Submission Title: Do you "dog ear" books?
```

```
Sentiment: 0.0
   Submission Title: Do you read YA novels as an "older" adult?

Sentiment: -0.4767
   Submission Title: Authors getting the protagonist's profession all wrong in
their novels


'The average sentiment was -0.030709999999999998'
```

[ ]:

Finally, create a list of all your average sentiment variables (e.g. all_averages = [news_sentiment, ...]). Use a for loop to iterate through that list, keeping track of which subreddit has the highest average sentiment, and display the highest score.

[12]:
```
mylist = [average_news_sentiment, average_cute_sentiment,␣
 ↪average_shoes_sentiment, average_books_sentiment]
display(mylist[1])
```

```
0.40859000000000006
```

[13]:
```
value = -1

for average in mylist:
    if average > value:
        value = average

display("The highest sentiment value was " + str(value))
```

```
'The highest sentiment value was 0.40859000000000006'
```

[ ]: