Brianna Tovar

**Lab 1: Option A**

For this lab, my problem to solve was adding a method to a program created to sort submission replies of negative, neutral, and positive replies attached to it. The method would then print these lists and would be used for any sort of submission given to the program.

My solution to this problem is to first plan out the missing method. This method needed to gather submission replies from a Reddit URL and sorts these replies into different lists ranging from positive to negative comments. After reviewing over the instructions, I got to work on first: traversing the replies to print. When first running the code the professor gave to us, I kept getting an error of, “No module nltk”. After looking up this problem, I was able to get a new error of, “No module nltk.metrics”. I emailed the professor back and forth about this problem, and I attended the IA’s office hours for help. Unfortunately, neither could figure out a solution, nor could I find one online. It took a matter of hours on the last day to be able to figure out a solution to this problem, and finally.. finally be able to work on my method.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Throughout my system interpreter problems, I kept testing methods in a different file to manage to traverse a list. Then, when I was able to get my code working with my interpreter, I moved over my practice and switched around certain variable names. Each time I changed my method, I kept using the test methods of “print(comments[0].replies[0].body”, so I can see if I can even replicate the print from the main method, then move on past errors I came across. Unfortunately, I’ve already completed what I had in the lab and closed the program, so my different runs and errors won’t show. On the other hand, I do have a screenshot of my final run and final output.

A screenshot of a computer

Description automatically generated

Overall, I learned how to traverse a list well, but not a tree-related list, such as the comments from Reddit. I’m still unsure about this lab, but will definitely continue asking my professors and TAs for help when needed. They continuously gave me help, even when they didn’t know much of the problem that arose from my code/laptop. Afterall, giving up is not an option.

**Academic Honesty:**

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.

**Source Code**

import nltk

from nltk.sentiment.vader import SentimentIntensityAnalyzer

import praw

reddit = praw.Reddit(client\_id='8n6lGHFZSRqmnQ',

client\_secret='TH1UO5SIpE24oDZliVmGmDgfR4U',

user\_agent='my user agent'

)

nltk.download('vader\_lexicon')

sid = SentimentIntensityAnalyzer()

def get\_text\_negative\_proba(text):

return sid.polarity\_scores(text)['neg']

def get\_text\_neutral\_proba(text):

return sid.polarity\_scores(text)['neu']

def get\_text\_positive\_proba(text):

return sid.polarity\_scores(text)['pos']

def get\_submission\_comments(url):

submission = reddit.submission(url=url)

submission.comments.replace\_more()

return submission.comments

#kept getting nltk.metrics error until fixed, delayed my coding

def main():

comments = get\_submission\_comments('https://www.reddit.com/r/learnprogramming/comments/5w50g5/eli5\_what\_is\_recursion/')

print(comments[0].body)

print(comments[0].replies[0].body)

neg = get\_text\_negative\_proba(comments[0].replies[0].body)

print(neg)

print("---------------------")#added to separate my code

main()

def process\_comments():#traversing replies

comments = get\_submission\_comments('https://www.reddit.com/r/learnprogramming/comments/5w50g5/eli5\_what\_is\_recursion/')

if len(comments)==0:#base case if no info

return ""

else:

count=0

print(comments[0].replies[count].body)

neg = get\_text\_negative\_proba(comments[0].replies[count].body)#sorting if which one

pos = get\_text\_positive\_proba(comments[0].replies[count].body)

neu = get\_text\_neutral\_proba(comments[0].replies[count].body)

count+=1

print("Negative",neg)

print("Positive",pos)

print("Neutral",neu)

return comments[0].replies[count].body#need to repeat for each reply

process\_comments()

#---------------------------------------------

#separated to test

"""

#traversing list of text from Reddit

def process\_comments(text):

if len(text)==0:

return ""

if get\_text\_negative\_proba(text[0].replies[0].body)<=1: #if reply is closer to neg

print(get\_text\_negative\_proba(text[0].replies[0].body)+"Negative") #prints when True

return process\_comments(text[0].replies[0+1].body) #returns next reply

if get\_text\_neutral\_proba(text[0].replies[0].body)<=1:

print(get\_text\_neutral\_proba(text[0].replies[0].body)+"Positive")

return process\_comments(text[0].replies[0+1].body)

if get\_text\_positive\_proba(text[0].replies[0].body)<=1:

print(get\_text\_positive\_proba(text[0].replies[0].body)+"Neutral")

return process\_comments(text[0].replies[0+1].body)

"""