Lab Report 1 – Option A

**Introduction**:

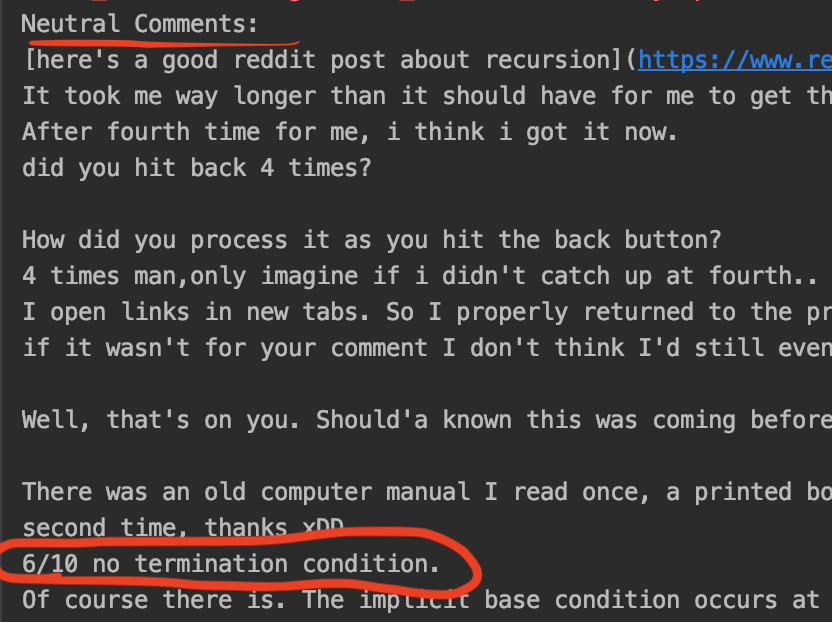
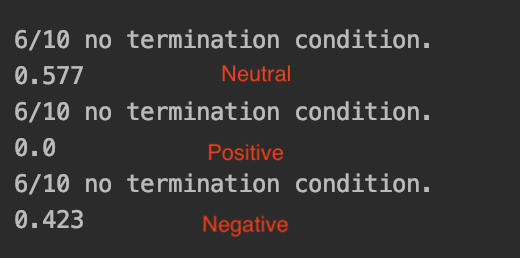
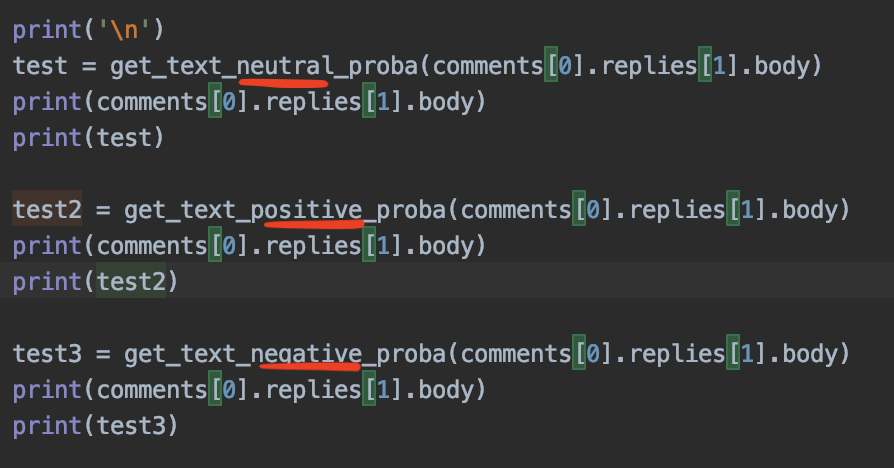
For lab one, I decided to do option A with was the Reddit lab. In this lab, the problem was that I had to create a recursive method in which it would separate the comments and comment replies of a Reddit post into lists depending on whether the comment was positive, negative, or neutral. In order for the problem to be solved, I had to use the methods already given in the code which determines the probability of the comment being positive, negative, and neutral to see in which list the comment belong to.

**Solution**:

In order solve the problem of separating the comments into their correct lists, I created a recursive method which in the parameter has the comments\_list which had every comment, and also it would take in the 3 lists that I created. In order to go through every single comment one by one; I created a for each loop which went through every comment in the comments\_list until there were no more. Afterwards I created an if statement that called the get\_text \_neutral\_proba method with the current comment. Since the probability method gave me a number of the probability of the comment being neutral, I used it in my if statement by comparing it to the number .5 because if the probability of the comment was greater than or equal to .5 it meant the comment was neutral. If the probability of the comment was greater than or equal to .5 then that comment was appended into the neutral list. If it turned out that the comment was not greater than or equal to .5 it meant that the comment was either negative or positive, to know in which list it belonged to I created 2 else if statements which called the get\_text\_positive\_proba and the get\_text\_negative\_proba and again compared the comments probability to .5. If the comment’s probability was again not greater than or equal to .5 it would go into the next if statement until it found its right list, and once it did, it was appended into the list. After the comment put into its correct list, the process would now repeat with that comment’s reply, by creating a recursive call with the comment’s comment. Once it was done with the comment’s reply, the for each loop would continue with the next top comment.

**Experimental Results:**

The way I tested my method to see if it worked and put all of the comments in the list they belong to, was that I picked a random comment and checked the probability of it being neutral, positive, and negative. By checking comparing the 3 probabilities I saw which one was the highest to know in which list it belonged to; after knowing in which list it belonged to, I checked my already printed lists with the comments and checked if the comment I picked was in the list it belonged to because of its probability. I did this test with various comments and the results were always correct which meant my method was implemented correctly. Here is an example of the code I made in order to get the probability of the comment, and also the output I got, and a screenshot of how the same comment was in the list it was meant to be on.



**Conclusion:**

By doing the lab I mostly learned a lot about Python, because before this lab I had never actually used Python. Not only did I learn how to use Python but also, I was able to learn a lot about how for each loops can be used in recursion. Although I thought I was pretty good at recursion, this lab turned out to be a challenge because of all of the other methods that were used inside that same recursive method I had to create. Overall it was a great experience doing the lab because it was not so much in my comfort zone and also not in my frustration zone.

**Appendix**:

Import nltk

From nltk.sentiment.vader import SentimentIntensityAnalyzer

Import praw

Reddit = praw. Reddit(client\_id= ‘copy and paste your client id here’,

Client\_secret= ‘copu and paste your secret here’,

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)[‘neg’]

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

Return sid.polarity\_scores(text)[‘neg’]

Def get\_submission\_comments(url):

Submission = reddit.submission(url=url)

Submission.comments.replace\_more()

Return submission.comments

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)

Main()

**Honesty Certification**

I Ana Arellano 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 Ana Arellano did not share my code or report or provided inappropriate assistance to any student in the class.