Kevin Rodriguez

LAB 1 Report

**Introduction:**

In Lab 1, we are given a URL of post from Reddit. The main purpose of this lab is to create a program that traverse throughout the Reddit’s post’s comments. By doing this, you encounter yourself with a reply to the post’s comments’ and a reply to that reply and so on. While traversing through the list of comments and replies, you have to distinguish each comment as positive, negative or neutral. This is done by using Machine Learning. After distinguishing what category the comment or reply is, a list must be created for each of the categories (positive, negative, neutral), and display each list. Lastly, extra credit is to implement 3 methods. One to display the oldest comment and after each call would display the next oldest comment. Second, would be the same thing but only with the positive comment list. Lastly, is to create same method but with the negative comment list.

**Design & Implementation:**

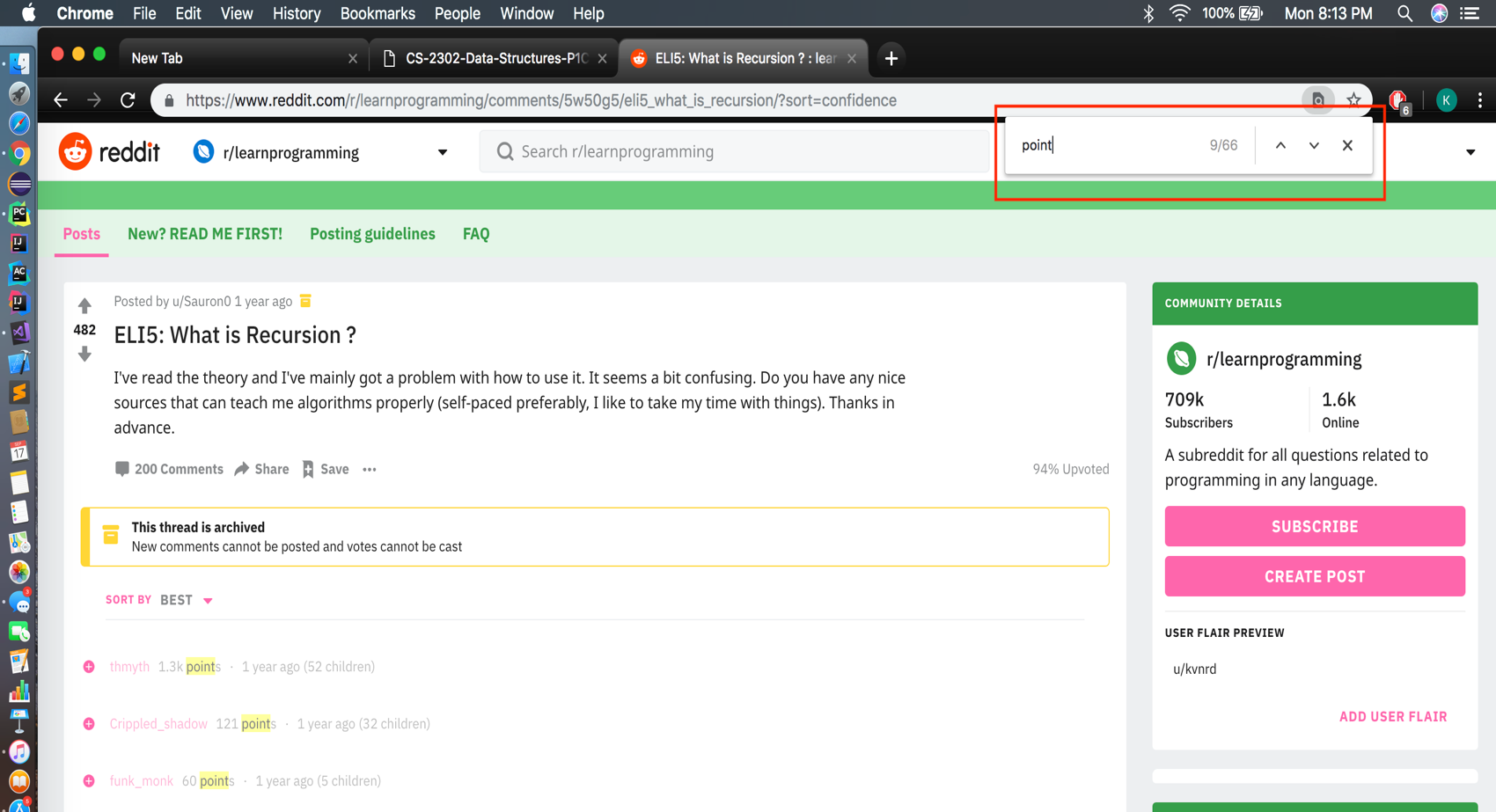
In order to traverse to the entire comments, I first created a method called traverse\_comments, that receives a list of comments. Since each comment can have its own reply and so on so forth, this method consists of a for loop that will end till it reaches the length of the entire comments list that the method receives. In order to go into the replies of the comments, I inserted a try and except, with the ‘try’ consisting on recursively calling the method again but with the replies of the comment (comment.replies). Inside the for loop, I created 3 variables (neutral, positive, negative) that would receive a value by calling the Machine Learning methods with the comments body. After receiving the values of each of the 3 variables, I compared each of them by using 3 IF statements and also by implementing the max() function in python. In order to separate these comments, I created 3 global variables (lists) which consist of the 3 categories for the comments. I used lists as my solution because it would help me for the extra credit that is included in the report. Inside each of the IF statements, I appended the comment to the corresponding list. Lastly, I print each of the three lists.

***Extra Credit***

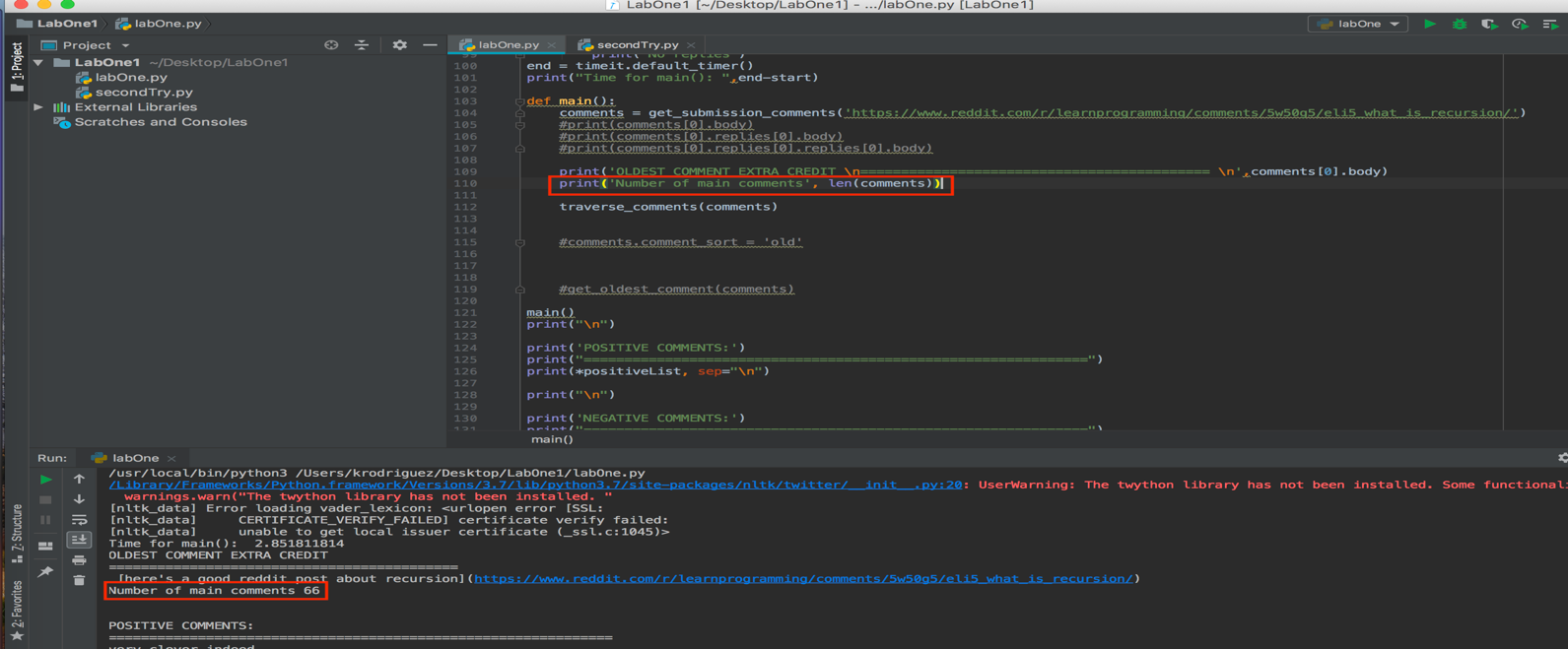
The first thing that I did was to edit the source code, specifically the method ‘get\_submession\_comments’. I added the function (submission.comment\_sort = 'old'). This function was used in order to receive the submitted comments sorted from oldest to most recent. In order to print the oldest comment out of all of them, all I had to do was to print the comments list at index 0 (print(comments[0].body). After doing this, I implemented the sort(reverse=True) function to the positive comments list and also to the negative comment list. The reasoning behind this, is to be able to just pop() the list and print the oldest comment and at the same time erasing it from the list, so whenever the method oldest\_positive\_comment or oldest\_negative\_comment is called, it will print the next oldest comment.

**Test Cases and Results:**

The first case that I made was a simple one. I printed out the length of the list of comments, meaning how many **main** comments are there. The following 2 pictures will show how the number of main comments match with those with the Reddit post.



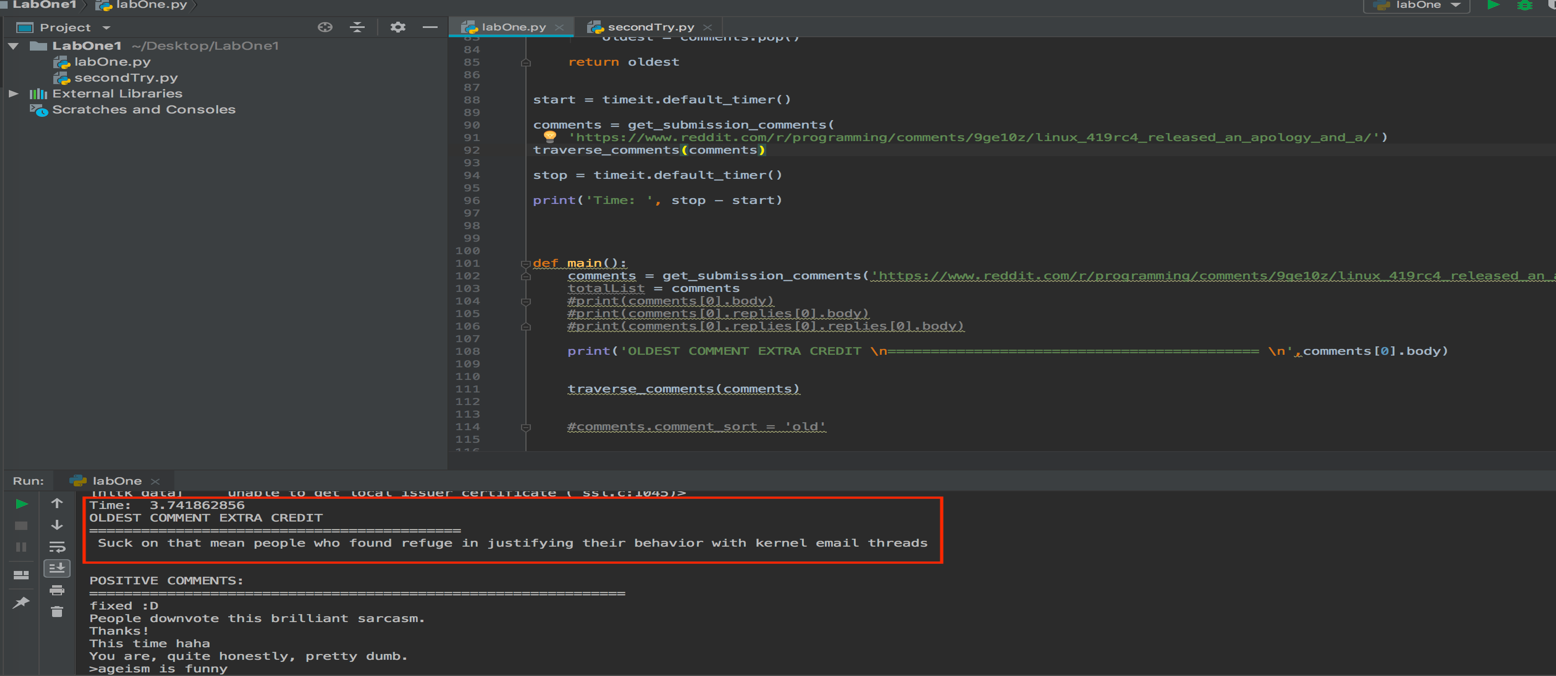
Since each comment has a point associated to it, I closed each main comment and searched for the word ‘point’ in order to see how many main comments there were.

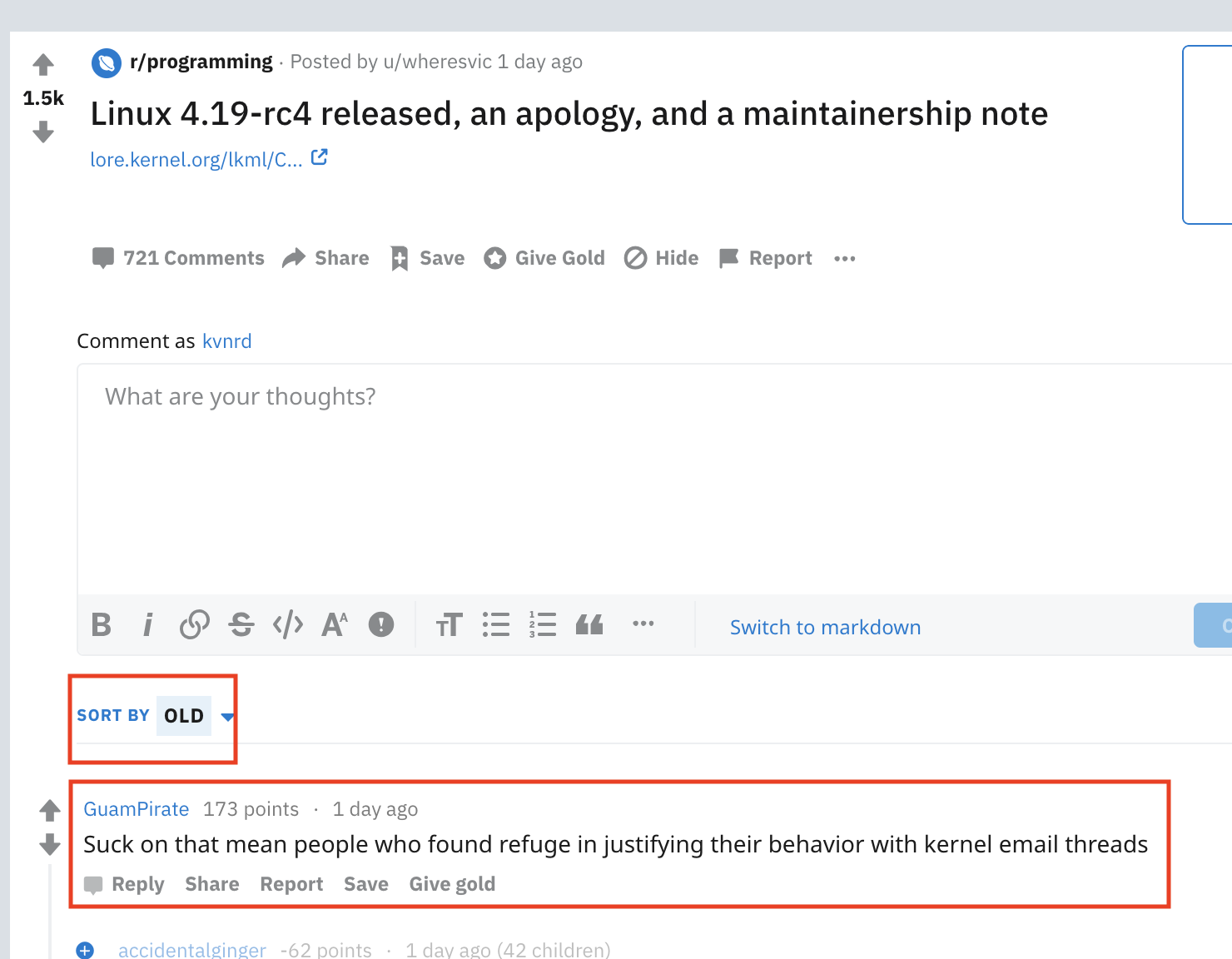


In Pycharm you can see in the red rectangles that I print the length of the comments and in my console I get 66.

**Second Case:**

In order to see if my code would work for any Reddit URL, I looked up some other Reddit posts and copied its corresponding URL into my code. The following is the Result.

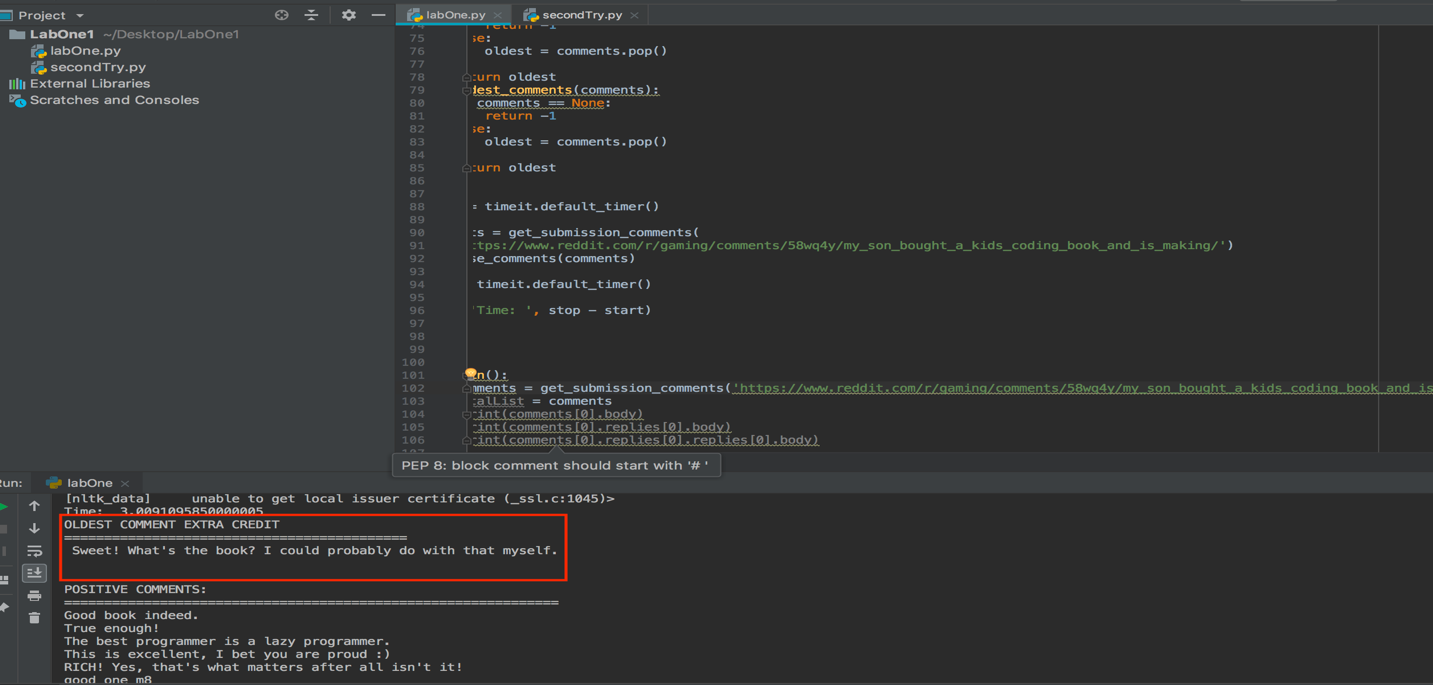




As you can see, not only was it able to read the comments, my program was able to print the oldest comment in the entire thread. I also tested how much time it takes for my program to run.

In the first picture, you could see how long the program took in order to gather the comments, traverse through the comments, rate them (negative, positive, or neutral), append to the appropriate list. This specific post with the number of 721 comments, took approximately 3.74 seconds.

**Third Case:**

****

****



The following table illustrates the three test that I conducted. It shows the amount of time each test takes depending on the number of comments and as you can see the larger the number of comments the longer the program takes.

**Time Complexity**

The time complexity of the program was pretty quick in regard to the size of the input. The Big O notation of this program is **.** Since it has to traverse to every single comment and reply, no matter what it is going to be the length of the entire list. When the posts have a lot of comments and replies, as seen in the method ‘traverse\_comments’ if in the for loop the comments[i] has replies, it recursively calls the same method going into the for loop again, therefore making it **.**

**Conclusion**

This lab was really interesting and a great way in order to refresh your brain about recursion. One of the main things that I learned in this lab is PYTHON. Since I was so used to JAVA, it forced me into researching and learning this new program which is a great tool have. Alongside this, the usage of PRAW was something new to my mind. I learned how to use new functions to your benefit when working with Reddit. I also learned that implementing a for loop with a recursion call inside could work in these types of scenarios. I am looking forward to keep working with PYTHON and becoming more proficient with the language as well as receiving challenging labs.

**Academic Dishonesty Certification**

“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 students in the class.”

Kevin Rodriguez

09/17/18