**Sarah Zazyczny**

**Part 3: Project Write-up and Reflection**

1. **Project Overview**
   1. I used IMDB Movie Reviews which first required loading the imdbpie package in the command prompt using “pip install imdbpie.” Once installed I was able to pull movie reviews for my selected film, *Clueless*. I chose to analyze the movie reviews using sentiment analysis with the NLTK - the Natural Language Toolkit. From my analysis I wanted to be able to identify the reviewers’ sentiments on the movie as well as discover what the main topic of the review was based on word frequency distribution.
2. **Implementation**
   1. For my text mining assignment I had to import different packages, work with different data structures, and create functions to complete sentiment analysis of movie reviews. By using the imbd database the movie reviews are imported as a dictionary. The reviews dictionary has the following keys: dict\_keys(['@type', 'base', 'paginationKey', 'reviews', 'totalReviews']). The reviews dictionary values contain additional dictionaries with review information. To get the review information that is required for sentiment analysis it was necessary to transform the review text into a string. This can be observed by the reviewA, reviewB, and reviewC examples, as well as *review = (reviews['reviews'][n]['reviewText'])* within the *review\_sentiment\_analysis(n)* function. This is an important step because the *nltk* package function for *SentimentIntensityAnalyzer().polarity\_scores()* could not be applied to the reviews in dictionary form.
   2. When it came to tokenizing the review strings into individual word strings I had to decide what process to use. I first used the *nltk.tokenize* package, but this did not provide the results I wanted. The output printed tokenized letters and did not strip the punctuation or special characters. I had to find another option to tokenize the reviews which led me to find the function *text\_process()* on stack overflow. This function provided the results in a list printing each word as its own string. This made it easier to analyze the text used within reviews.
3. **Results** [~2-3 paragraphs + figures/examples] Present what you accomplished:
   1. First I created example cases for reviewA, reviewB, and reviewC to test the functionality of the new packages for text analysis. I selected and computed the sentiment scores for these three reviews individually. In addition I used the *len()* and *count()* functions to further analyze these reviews. As shown in Exhibit 1, I was able to print the sentiment scores, the length of the review in characters, and count the number of times a certain word was used in the review. For the example I chose to search for “Cher” because she is the main character of the movie *Clueless*. To simplify my findings, I created the function *review\_sentiment\_score()* to print movie review sentiment scores for selected reviews based on index number (Exhibit 2). This function prints the same results as my examples, but can be used to find the score for any selected review. The results show that the three selected reviews have mainly neutral language and postive sentiment is slightly greater than negative.
   2. In addition to sentiment analysis I wanted to analyze how certain words were utilized in the text. For my examples I first had to tokenize each review using the *text\_process(text)* function. Then I was able to use *FreqDist()* on the tokens to print the frequency distribution. From here I used the *most\_common()* function to see what the top ten words used and their frequencies. Many of the words relate to the setting of the movie: high school and Beverly Hills and it is not surprising to see overlaps of most common words across the reviews (Exhibit 3). To simplify this process I created a function *freqdist\_tokens(n)* which tokenizes the reviews and creates a frequency distribution when run. The results are the same as Exhibit 3 and the same *most\_common()* analysis can also be completed on these reviews. It would be interesting to look at this information on a larger scale to see what the biggest topics of discussion are on the movie *Clueless.*

Exhibit 1

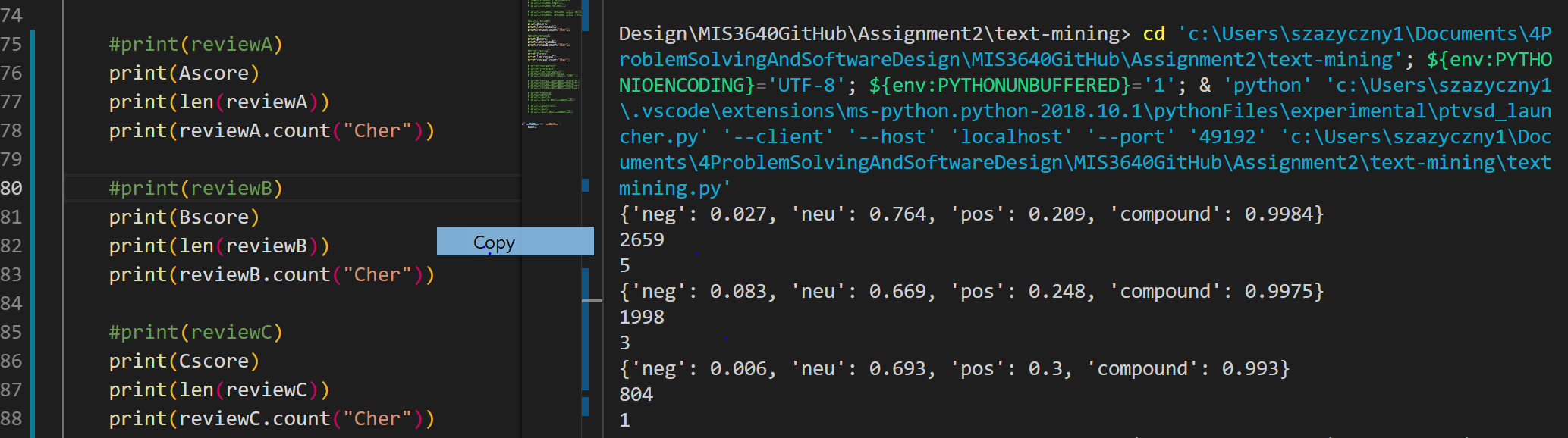


Exhibit 2

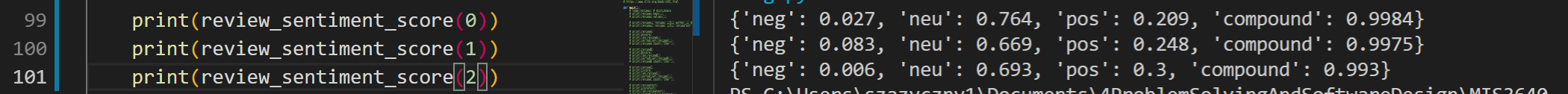


Exhibit 3

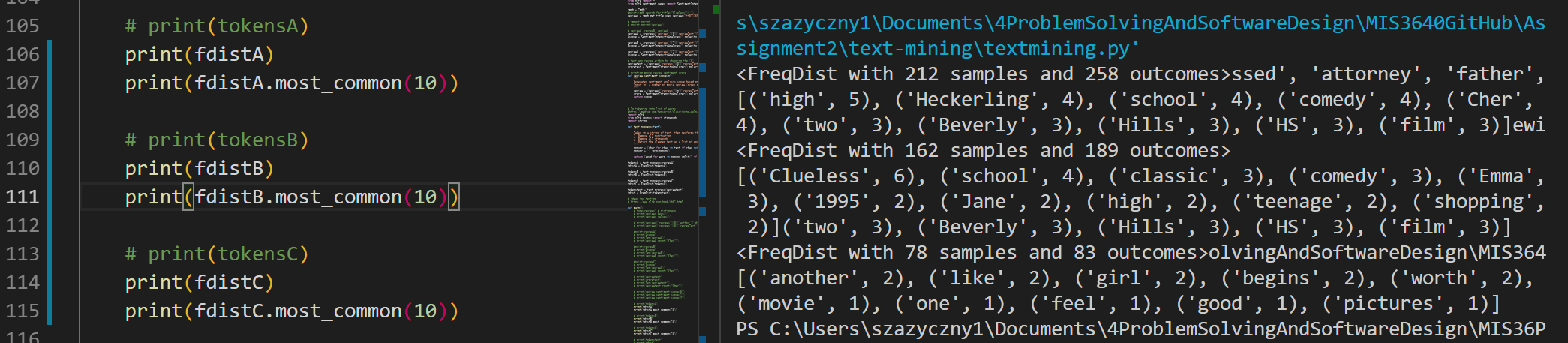
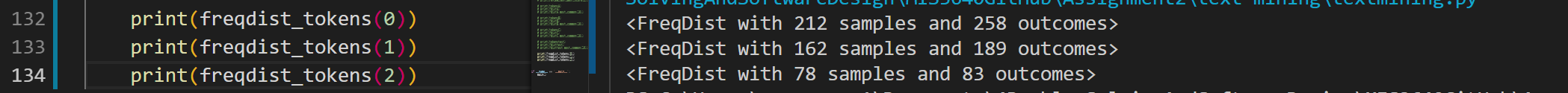


Exhibit 4



1. **Reflection**
   1. I think that things went well during this assignment. I was unsure of how to begin this assignment and attempted to make it much more complicated than it needed to be. But once I figured out what packages to use and how they were used the assignment worked out.
   2. I spent a lot of time testing different functions and learning how to use the packages and import data/use correct data structures which limited the time I had to create more functions. I worked alone on this project and believe the functionality could be greatly increased if I were to work with a partner. I used the main function for testing and created a lot of test code (much of which I have deleted). This process may be time consuming but it helps to see hands on what you are accomplishing step by step.
   3. Going forward I will ask for more help from peers when I get stuck and hopefully be able to develop some collaborative code.