Alison Ostlund
CSCI3832 HW3

I used three different libraries in order to get my program to work. First I used Counter, which is helpful when trying to get the frequencies of the words within the training set to obtain a unigram. Next I used the string library to help split up the text files when reading in the sentences. This was useful to strip punctuation from the text file to get more accurate probabilities on words. Then I also used the math library in order to use logs when handling underflow.

To approach this assignment, first I read in each text file separately to get two unigrams one for positive training and the other for negative training. When reading in the files, I striped the files from the ID to remove them from the training set, I also removed punctuation and converted the training text into lowercase. This was to help ensure accurate probabilities when using the training set with the test set.
Using the Naive Bayes equation, I was able to generate probabilities for each word in the positive and negative training sets. I used smoothing here by finding the fraction of the current word + 1 over the size of the particular unigram + the size of the vocabulary as a whole for the training sets. For cases that require underflow handling, I took the log of the result of the Bayes equation. In the classification function, this is where I took in the test sentences and ran it through the classification equations using the probabilities generated from the training sets and multiplying them together. I check if the pos and neg probabilities are 0, and if so then I run the same sentence through the log probabilities that handle underflow using addition of the probabilities. From here I check if the neg or pos probability is greater than the other, and then classify the sentence positive or negative given which had the higher probability.