John Mikos
Homework 2

1. Splitting of data into training, development, and test was done using Sci-kit learn's train_test_split() since it was permitted in class.
   Additionally, I believe it necessary to state that I used the movie review dataset that we used for homework one. I felt this dataset was split and cleaned really well to the point where I thought it was good for training.

2. I began classifying a Naïve Bayes Classifier splitting all the words of every line and just adding it to a bag of words, removing all stop words and/or punctuation. I treated positive and negative as two separate classes. I then used collections' Counter tool in order to count how often the word occurred in each class separately. I then implemented Multinomial Naïve Bayes and mapped all the word percentages for each class, positive and negative. I then tested my classifier on the development dataset and got consistently between 25 and 31 percent correct reviews. I thought this was not good enough, so I implemented a new way to only include the top *x* words as polar words. This is done because some words in the sentence might drive the percentage lower than it should be just because it rarely shows up. Words that would not show up would be penalized with a percentage of occurrence lower than any other word at 1/100,000. When I implemented only including words that exceed *x* instances, my percentages increased and later polarized (meaning one class was fairly accurate and the other was in the 30% range). The reason for this is because the higher the *x* instances, the less words to have an actual percentage which gave equal positive and negative percentages. This would result in whichever side the equivalent state would take, the higher that percentage. For this reason, I chose the equivalent condition to go to the positive class and tried to max the percentage of the negative class. With this I received my max at $x = 80$ (with each class's dictionary having only about 30 words), with a positive percentage of about 68 and negative percentage of about 50. I still felt that this percentage was artificial since when I looked deeper into the words in each class did not seem polar enough. I now implemented a min and max instances of words and tuned the numbers to my development set. I got my minimum instances value of 4 and maximum value of 200 (with a dictionary of about 2000 words each), which resulted in a positive percentage of about 60% and negative percentage of about 55%. Although this decreased my positive percentage, it increased my negative percentage which means that there is less guessing needed to be done on the equivalence condition.

3. Using my classifier on the test set for the *x* instances method I got 71 for positive percentage and 52 for negative percentage. I then tried it on my new classifier (of min and max instances values), I got 62 for positive percentage and 57 for negative percentage. Both of these performed slightly better and there is not a huge change from the development set which means the this is a good classifier.

4. I found that if the probability of the word being found in the positive class was equal to the probability of the word being found in the negative class, the classifier was almost always confident, meaning that one of the classes had a probability greater than the other by at least 10-fold. Of course, this is not necessarily confidence because one words probability can contribute to 100,000-fold difference, but overall the percentages were quite different when they were not equal. I then updated to view the 100,000-fold difference and saw the positive classifications were about 90% confident and the negative classifications were almost 100% confident.

5. The most useful features for each class were definitely the more popular positive/negative words. Removing stop words is also very important because those words would return similar probabilities and dominate most sentences. Penalizing missing words was also very important for both classes because it resulted in the classifier understanding that a missing word is not a good sign. It was interesting to see ‘good’ pop up in negative reviews and still provide a good percentage.