In first approaching this assignment to implement a Naive Bayes classifier from scratch, I started with movie review data that was labeled as negative and positive reviews. In order to initially process this text, I started by cleaning it and removing punctuation and newline characters. I also had to remove apostrophes from the beginning or end of words sometimes so I split each review into tokens and then joined them back together once this process was completed. Next, I split the newly cleaned data into three sections—training data (70%), development data (15%) and test data (15%)—while keeping them under their separate negative and positive labels.

Once I was ready to start training my classifier, I first calculated the prior probability that a review was either negative or positive. Then, I split each negative review into tokens and added each unique word to the negative vocabulary. I then repeated this process for the positive reviews and calculated the counts for each word in the vocabulary within the training data. After that, I wrote the actual classifier, which takes in a document and calculates the probabilities that it is either positive or negative. It calculates the probability that a given word in a document falls within a positive or negative review. Then, it goes through each word in the given document and multiplies each of those probabilities by the prior probability that the document is either positive or negative. Finally, it compares the positive and negative probabilities and assigns the document whichever class has the higher probability.

After initially training the classifier, I began to evaluate its performance on the development set after making slight changes. The initial accuracy on the development set without making any changes was about 74.8%. First, I tried removing words that only occurred once in the training set from the vocabulary, but this decreased the development set accuracy to 74.0%. I also tried keeping only the top 1000 most frequent words in the training set but it also decreased the accuracy to 73.7% so I decided that neither of these options would be the best change to make. I then added general lists of negative and positive words to the vocabulary and counted them as if they occurred one additional time. This improved the accuracy to 75.2%. Next, I tried removing stop words using spaCy and this decreased the accuracy slightly to 74.6%. But I also tried removing words that were only one letter because the vocabulary included some "words" like "s" or "b" and this improved the accuracy again to 75.2%. The accuracy of removing words that were only one letter without removing stop words was only 71.7% and logically these words would probably not have a significant impact on whether a review is positive or negative. Therefore, I decided it would be best to remove stop words and one letter words from the vocabulary. Once I had made these improvements, I then trained the classifier again, this time on the combination of the training and development set and the accuracy increased to 93.2% (although this is expected since it was the training data). Finally, I tested this classifier on the test set and the accuracy was 76.6%.

Based on the classifications made, it appears this classifier is much less confident on reviews that contain more words, while it is much more prone to error on reviews that have very few words. For example, the review "for dance completists only" was classified wrongly as positive with a relatively large difference in the negative and positive probabilities (2.3252137800549268e-21). However, the review "the story of trouble every day is so sketchy it amounts to little more than preliminary notes for a science-fiction horror film and the movie's fragmentary narrative style makes piecing the story together frustrating difficult" was classified correctly as negative with a very small difference in the negative and positive probabilities (-3.3590832119180413e-167) despite having four words from the negative list. This demonstrates that although these lists of words are important features in classifying a review, the performance of the classifier depends heavily on the length of the review it is classifying—and will likely do best on reviews that are not too long or too short. Although the length of the review plays a large role, it does appear that these lists of words have a pretty big impact on the classifications. To illustrate, the review “a fairly harmless but ultimately lifeless feature-length afterschool special” contains the word “lifeless” and was classified correctly with a difference of -8.110539309430629e-41. Additionally, the review “wise and deadpan humorous” contains the words “wise” and “humorous” and was classified correctly with a difference of 5.858120283219098e-19.