To begin our testing, we decided to find the constant C value just using the sample words already provided us to create the SVM. Trying values ranging from 0.1 to 5 for the constant, we found the model to be most accurate with values below 1, and eventually settled on a C equal to 0.5. To further improve the model, we began examining the emails themselves for common words. Using misclassified emails from our cross-validation set, we tried to pinpoint words that would create valuable features; however, this quickly became tiresome as it was neither efficient nor very effective, with only small gains in accuracy coming from these efforts.

From here, we abandoned these efforts, and instead begin looking for the most common words in both ham and spam emails. We created dictionaries of each word in both ham and spam emails, sorted them by their occurrences using the OrderedDict function the “collections” library, and created two lists of the 50 most common words in each type of email\(^1\). This led to a large jump in accuracy, precision and recall. Going further, we eliminated words with lengths less than 3 to avoid using words such as “a”, “or”, and “the” as features. We began increasing the size of these lists, as taking 100 words increased our accuracy to over 90%. After a few more increases, we settled on taking the 600 most common words from each list which gave the model an accuracy of nearly 97%. Using any fewer or greater number of words led to a decrease in accuracy. Lastly, through the same process, we took the 600 least common words from each set of emails and added those words as features as well.

\(^1\) Collections documentation can be found here: https://docs.python.org/3/library/collections.html