After several grueling hours of debugging, we created a kick-ass SVM for classifying spam emails. After separating the training data into a training set of 6000 emails and a cross-validation set of 2000 emails, we were getting a standard 81% accuracy, so we knew we had to do better. The most obvious change we made was changing the words (features) that we train the SVM on. We created a method that runs through the emails and copies all unique words into a dictionary. If a word is repeated, its value in the dictionary is incremented by 1 to represent an increase in frequency. We then found a frequency that worked best (which was 70) and filtered words of frequency greater than 70 when checking them against the words in the email. This increased our accuracy on the training set and cross-validation set greatly (98.55%, 97.95%).

With a high accuracy, we felt that it wasn’t necessary to manipulate the word list anymore and went straight to finding the best c to use. We did this by using the subprocess module and using an array of different c’s in a huge range, .01 to 100 by huge increments. We found that .1 gave the best result, so we lowered the range to .1 to 1 with increments of .1 and found that .1 still gave the best answer, so this is the c we stuck with. After doing this, we got a 99.9% training accuracy and a 98.3% cross-validation accuracy. We were happy with the increase and happy that the difference between the two is only 1.3% meaning we aren’t overfitting.

Changing the word list to find the most frequent words and finding the optimal c helped bring our accuracy up greatly, however we wanted to try and do a little bit better. We took the ham emails and the spam emails and took the unique words in spam and the unique words that
Ammar Almahdy and Matthew Ottomano

appear in ham emails. We then went through all spam-unique words, giving a value of 1 if it appears only in spam emails but not in ham emails. Words that appeared only in ham emails were then used for the features vector. We thought that this may help our accuracy reach 100%, however our accuracy for the cross-validation went down to 97.67% so we decided to stick with frequent words instead.

We were happy with our accuracies and ended there.