In order to complete the assignment and create our model, we worked together on several occasions. At first, we wrote the code to create the svm files correctly. Once we had that, we began to work through different features of spam emails.

At first, we looped through the spam emails to create a dictionary of word frequencies in spam emails. After we finished that, we chose different amounts of the most frequent words to choose. We started with 1000 as the homework assignment suggested. We also tried increasing this to 2000 of the most frequent words, as well as just using all the words that were used in spam emails. Interestingly, as we added other features, we found that using 2000 of the most frequent instead of all the words in spam emails worked much better.

As an additional test, we looked at using the autocorrect package that python has to check the number of misspellings in emails. We looped through the content of the email when creating the svm files and counted the number of misspellings as a feature. However, when adding this feature, the precision actually decreased so we took out this feature.

On the same note as the most frequent words in spam emails, we also tried just looking at the words in all of a subsection of emails. Another group said they found a lot of success using the words of the first 20 emails as their features; however, our group did not find this same success. It is possible that other random factors or parameters they are using allowed that to work, but it did not work for us.

Lastly, we played around with the parameters of the svm. Without specifying $C$, svmlight was using a $C$ value of 0.0145. Every time we added a new feature, we manipulated the value of $C$ using a loop to find the best fit. With our final model, we found a value of $C$ that gives the best accuracy, precision and recall is 0.075.

In addition to $C$, we played with how many examples to use as training versus cross validation. At first we used about 80 percent as the training data, and 20 percent as the cross validation data. However, increasing it to be closer to 90
percent of the training data (~7000 elements) and 10 percent cross validation gave the highest statistics.

Overall, it seemed like using less features was the better way to go with the spam classifier in our case. Possible sources of our error include explanatory features that we may not have discovered or manipulated to the best of our ability.

Highest stats achieved:

Accuracy on test set: 97.50% (974 correct, 25 incorrect, 999 total)
Precision/recall on test set: 97.05%/97.27%