Christopher Grant
CS 1571 Homework 4 Programming Report

Scores:
Fold 1 False Positive Rate: 0.031 False Negative Rate: 0.074 Error Rate: 0.105
Fold 2 False Positive Rate: 0.030 False Negative Rate: 0.074 Error Rate: 0.104
Fold 3 False Positive Rate: 0.030 False Negative Rate: 0.054 Error Rate: 0.085
Fold 4 False Positive Rate: 0.033 False Negative Rate: 0.051 Error Rate: 0.084
Fold 5 False Positive Rate: 0.037 False Negative Rate: 0.064 Error Rate: 0.101
Avg Avg False Positive Rate: 0.032 Avg False Negative Rate: 0.063 Average Error Rate: 0.096

Looking at the results, we can see that the best results come when testing on folds 3 and 4, meaning that the best results occur when Folds {1,2,5} are in the training set. There is no specific indication to explain why this is, as according to the data splitting outputs, there is no real difference between these sets in terms of number of negative or positive values. This thus indicates that the data contained in Folds {1,2,5 } is better for learning and the data contained Folds {3,4} is better for testing that learning.

If we were to just chose the majority class, that would effectively have the same results as having no spam monitoring at all. Looking at the testing set samples, the negative is always in the majority, thus meaning that for every email, we would mark it as 0, and not spam. This obviously makes no sense because if we wanted everything marked as not spam, we wouldn't have to do any sort of analysis or learning. However, if we wanted to minimize the number of false positives, this would be a way to do it, as because every entry is marked as 0, there is no possibility of having a false positive (This could be useful in the case of someone who wanted to make sure absolutely nothing important got marked spam) Because our naive bayes implementation got some correct spam identifications, its score is better than just choosing the majority class and therefore would be the better option to use.