Tyler Bateman
Assignment 3 Report

I used the Vocabulary library from nltk.lm to create my vocabulary. My first approach was to get a vocabulary size close to 2500 by setting the unknown token cutoff to 1, and incrementing it until the length of the vocab was less than 2600. This worked well for the full document set, but when I trained the classifier on different sizes of training sets, the vocabulary size was inconsistent. My solution to this was to set the cutoff to the value such that if it were to be incremented by one more the vocabulary size would drop below 2500. I then removed words with frequencies just above the cutoff from the vocabulary until the size of the vocabulary was 2500. This kept the size consistent for all training sets.

For the multinomial version of Naïve Bayes, I scaled the weight of each word in the document by the number of occurrences in that document.

To facilitate using separate scripts for training and testing, I used the pickle library to store the training data.

Results
My spam classifier produced the following results:

Results for classifier trained on full 700 document training set:

	Predicted spam	Predicted non-spam	Totals
Actual spam	70	60	130
Actual non-spam	1	129	130
Totals	71	189	260

Precision: 0.9859154929577465
Recall: 0.5384615384615384
F score: 0.6965174129353234

Results for classifier trained on 50 documents:

	Predicted spam	Predicted non-spam	Totals
Actual spam	68	62	130
Actual non-spam	1	129	130
Totals	69	191	260

Precision: 0.9855072463768116
Recall: 0.5230769230769231
F score: 0.6834170854271358

Results for model trained on 100 documents:

	Predicted spam	Predicted non-spam	Totals
Actual spam	70	60	130
Actual non-spam	1	129	130
Totals	71	189	260

Precision: 1.0
Recall: 0.5615384615384615
F score: 0.7192118226600985

Results for model trained on 400 documents:

	Predicted spam	Predicted non-spam	Totals
Actual spam	63	67	130
Actual non-spam	0	130	130
Totals	63	197	260

Precision: 1.0
Recall: 0.4846153846153846
F score: 0.6528497409326425

Discussion

No matter the size of the training data, the classifier tended to have an extremely high precision while having a somewhat mediochre recall. Of course the ideal classifier would do well in both precision and recall, but if I had to choose either good precision or good recall for a spam filter, I would want one with good precision. I would rather have some spam slip through the filter than miss something important that was miscategorized as spam. By this metric, the classifier did farely well; False positives were extremely rare, and it still managed to correctly categorize about half of the the spam emails.