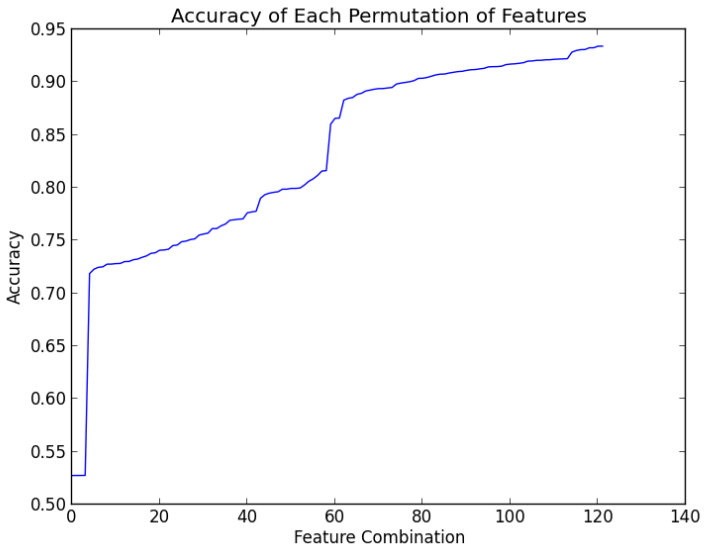
Robert Crimi
Feature Engineering Analysis
CSCI 5622 – Machine Learning
9/18/15

For this assignment, I took an exhaustive search approach. Below I have listed the steps I took to achieve higher accuracy:

- My first method of improving accuracy was to take every word in a sentence and make all letters uppercase. When only using the base feature (word count) this improved my training accuracy by 5.42%. This improvement in accuracy was due to the fact that there would be no unnecessary repeated words due to character case. For example, "Killed" would be classified the same as "killed."
- My next step was to exhaustively search for performance gains over all permutations of features. To do this, I used features from the sample script, such as "word", "one_before", "one_after", "dictionary," etc. I incorporated these features along with the Analyzer class to my classify script. I then created a wrapper script that would permute all features available and then run classify.py for each. I then took these results, sorted them, and plotted the results. Below is a graph of performance gains for each permutation.



The top scoring permutation was "word, all_before, all_after, one_before, characters" which got 93.41% training accuracy. While this scored highest on the training set, using a tfidf representation scored highest on Kaggle. I believe this scored highest because this representation gave lower weights to common words such as "the", "and", "it", etc., which are not important for classification.