(**MY USERNAME ON KAGGLE IS jale7220, BUT I SHOW UP AS Jay LeCavalier**)
Machine Learning
Due 9/18/15

Analysis Portion (Feature Engineering)

At the beginning of the assignment, the features were just single words of sentences that were either from spoiler posts or not from spoiler posts. I decided to give these single words some context by increasing the number of words allowable within a single feature. This way, pairs and even triples of words were being considered in addition to single words as features. This strategy increased my accuracy by a lot because words were being considered in addition to the words surrounding them. For example, “the” might not be a very revealing word about whether or not a sentence is a spoiler, but “the dentist” may be. To implement this change, I assigned the attribute “ngram_range” to the CountVectorizer using two arguments passed in by the user. The user specifies a min and a max. the min is the minimum number of words that should be considered together as a single feature (I usually got the best results when I just used 1 for the min), and the max is the maximum numbers that should be considered a single feature (I got the best results around 3-6).

Another modification I made was to create a threshold for how frequent a word had to appear in the document before it could be considered a feature. For example, if the word “bagel” only appears once in the entire training document, I don't want it to sway me very much in deciding whether a sentence is a spoiler or not. I implemented this threshold using the “min_df” attribute of the CountVectorizer, which specifies the minimum number of times a word must appear in a document in order to be considered a feature. This can also be specified by the user as an argument. I got the best results when I set this to 2.

Perhaps the least significant modification I made was to remove 'stop words' from the list of features. There were a few errors in the document, such as the word 'sentence' or 'trope' appearing in the document. I simply told the featurizer not to count these as features using the “stop_words” attribute of the CountVectorizer.

Finally, the most important modification I made was to actually manipulate the strings that were passed in as training data. I found that the source (or 'trope') of a sentence could be very telling about whether or not the sentence is a spoiler. Some sources for information were more likely to produce spoilers than others. For example, the source of information called 'the reveal' usually produced spoilers, while 'NBC' did not. Therefore, I appended the trope to the end of each sentence in the document. This made the featurizer consider the source of a sentence when training (and, thus, while testing as well). This increased my score more than any other modification I made because it added additional, relevant information into the list of features. Instead of just checking the words in a sentence, the algorithm now also checks the source of a sentence. To implement this, I simply appended the tropes as strings to the end of each sentence string before doing the training.

Overall, I was able to increase my score from about 62% to roughly 69%. The feature modifications I made that had the biggest impact were the “ngram_range” and the inclusion of tropes as features.