**Homework 4: Analysis**

**Submitted by Aashish Panta**

We were given both train set and test set for this programming assignment and we were required to determine a classifier that determines the best accuracy on the test set. I achieved the accuracy of 90.128% by using various techniques for preprocessing and building a suitable classifier for the project. These techniques and alternative options are analyzed below.

Since the data given were just raw, I removed all unnecessary symbols like “**. ; : ! \ ' ? , \ " ( ) \ [ \ ] ]”**

and changed all the words to lower cases. I defined a custom stopwords set and used it in ‘ngram vectorizer’ to remove the most commom neutral sentiments. This will prepare our data for tokenizing. Following that, I used ‘Countvectorizer’ from python library to tokenize the text data and make it ready for Support Vector Machined with linear kernel. I then, set the range of n-gram to 2 words so that I can get reasonable sentiments to train and test. Setting n-gram means taking two words as a sentiment. At last, I used the SVM library to classify them (setting value of to 0.007) to get the accuracy of 90.128%.

There were many ways to complete this project, but I found this the most efficient one for me. This could be done without using the stopwords, but that would just take extra time without being productive. The n-grams range could be set to 1 or 3, but then that would not give precise analysis. I used logistic regression for analyzing the dataset, but that was not satisfactory as the accuracy was found to be around 88%. For being 2% more accurate, I changed the whole classifier to SVM, changed the n-grams range to 2 words and added some neutral sentiment words to the list of ‘stop words.’

This analysis beats the accuracy of Aaron from the website.