INFO284
Spring 2018
First Obligatory Group Assignment
Naive Bayes for Sentiment Analysis

Candidates: 1, 5, 98, 130
Character count: 2175 (with space), 1810 (without space)

Error rates for training set:
Positive:
- Classified as positive: 11220 (89,8%)
- Classified as negative: 1280 (10,2%)

Negative:
- Classified as negative: 11627 (93%)
- Classified as positive: 873 (7%)

Error rates for test set:
Positive:
- Classified as positive: 9909 (79,2%)
- Classified as negative: 2591 (20,8%)

Negative:
- Classified as negative: 10944 (87,6%)
- Classified as positive: 1556 (12,4%)

Precision = 11220 / (11220 + 873) = 0.928 = **92,8%**

Recall = 11220 / (11220 + 1280) = 0.898 = **89,8%**

F-measure = 2 * 0.928 * 0.898 / (0.928 + 0.898) = 0.913 = **91,3%**

Explanation of the performance and how we can improve it:

Our program has an error rate of 10,2% on the training set for positive and 7% for negative. After removing stop-words the error rate for positive got two percentage points worse, but it's still the same on the negative. When running the code on the test set after removing the stop-words, both positive and negative error rates got improved by approximately two percentage points. Since we made the models on the training sets it makes sense that the error rate is higher on the test set.

We measured precision to 92,8% and recall to 89,8%. This made our f-measure turn out to be 91,3%. We used the F1- measure where precision and recall are equally balanced and $\beta=1$.

Term Frequency-Inverse Document Frequency: We could decrease the score of a word if the word has equally high frequency in the negative and positive models. We can also give words that have higher frequency in one model, higher score on that mode. By doing this we expect to give more precise classifications on the reviews in the test set.

To further improve, we could use some adverbs to join for example all the "not"s with the trailing word, so that if a negative review contains "..not good..", the algorithm will store it as "notgood". We think that doing this will impact the negative reviews, giving negative reviews a more negative score by making words that are classified as positive, negative.

We could also pre-process the models by joining inflected words. This will give those words a higher frequency and will therefore have a higher impact on the classification of the review.