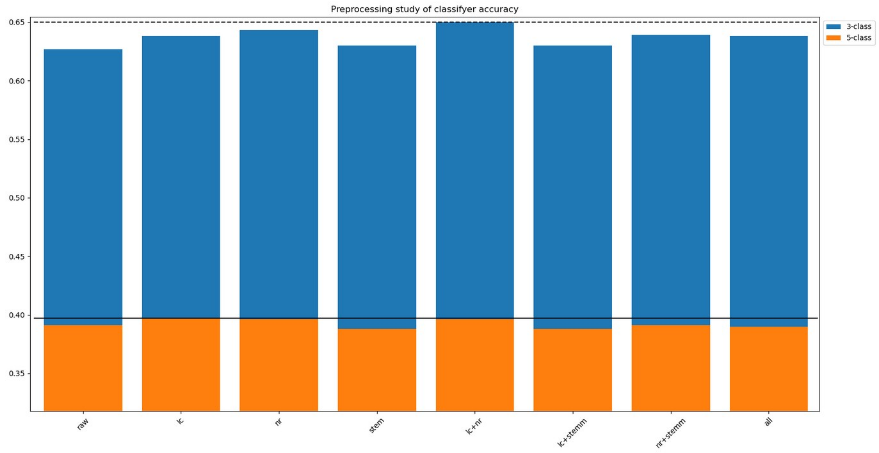
Luca Del Basso COM3110 Assignment 2
Sentiment Analysis of Movie Reviews Report

Description

In this assignment I fully implemented a non-binary Naive Bayes Classifier for sentiment analysis of film reviews. I undertook preprocessing steps which included lowercasing of words and noise removal. I experimented with stemming as an additional preprocessing step however this reduced the accuracy score of the development sets in my first model. My second model removed any word found in the reviews, after preprocessing, that were not a positive or negative word, provided by sets I found online. Again using these features seemed to produce a lower score than using all the words as features, even upon experimenting with a variety of word groups such as adjectives. In the second model, stemming as a preprocessing step, in addition to stemming the positive and negative word sets, produced higher results, I think this was because there were variations of positive and negative words in the text files however stemming them all meant less chance of variation when extracting features.

Preprocessing study



Whilst preprocessing steps had minimal affect on the accuracy performance for all words as features, noise reduction in addition to lowercasing provided the highest accuracy of 65%.

Noise Reduction, Lowercasing and all words as features

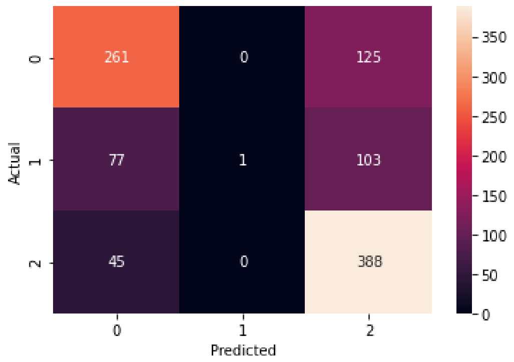


Figure 2: accuracy: 65%

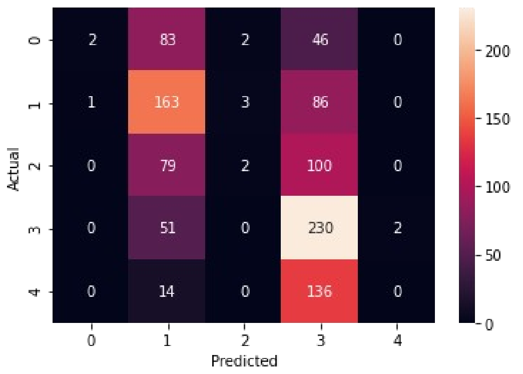


Figure 1: accuracy: 39.7%

When using all words as features, it is clear that the classifier struggled to determine all but 1 review from the development set as neutral in the 3 class system. This could be because class 1 “neutral” had a representation in the training set of just ~19.5% compared to that of ~38% and

~42% for negative and positive respectively. By having representative training data and data that has a more even distribution of each class then biases based on prior probabilities alone will be reduced. You can see that a large amount of misclassification has come from the neutral class, with negative and positive being successfully classified 68% and 90% respectively. For 5 classes the problems seen in the 3 class set are increased in the 5 class, with class 0, 2 and 4 being poorly classified throughout. Having a larger amount of classes requires larger training data as there needs to be a sufficient amount of training data for each class. Again in the case of the 5 class we see the classes with the highest priors (1 and 3) being classified the most compared to the others. Highlighting that the training data is not evenly distributed.

Noise Reduction, Lowercasing, Positive and Negative words

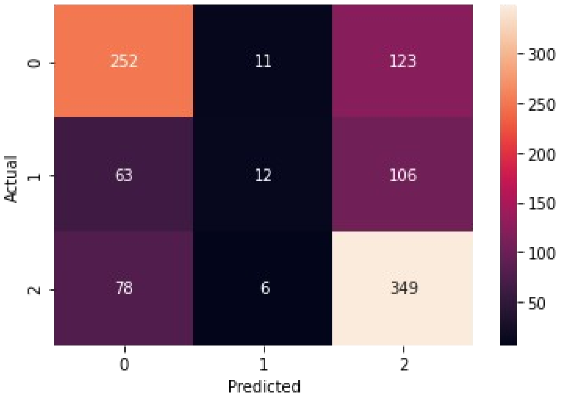


Figure 3: accuracy: 61.3%

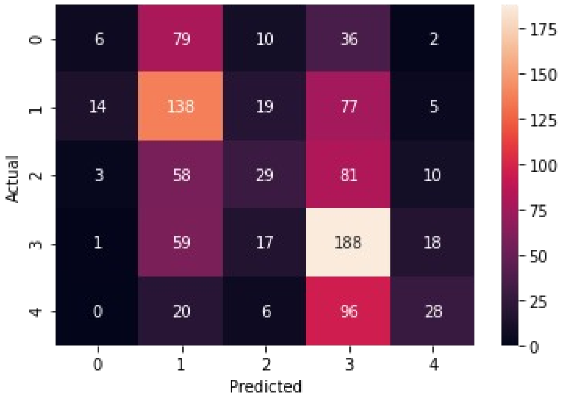


Figure 4: accuracy: 38.9%

By using positive and negative words you can see higher positive classifications for the neutral class but only slightly, with higher miss-classifications between classes 0 and 2 -in the 3 class model. in both the 3 class and 5 class cases, there were slight improvements on the classification of the neutral class. And the 5 class model performed approximately just as well as the 5 class model using all words as features. In some cases it may be better to use this second model as it would have reduced features (this model had a vocabulary ~3.5 times smaller than using all words) but a somewhat similar performance and also successfully classified a higher range of classes, for example the 4th class in this model positively classified 28 reviews compared to the zero positive classifications in the first, so it achieved a similar performance with a wider distribution of positive classifications.

Discussion

The curse of dimensionality and representative training data are the most important aspects to consider when training a naive Bayes Classifier. Using word groups alone seems to not be enough in regards to improving classification performance, however whilst producing slightly reduced results, significantly reduces the feature size. As a result, in my case it may come down to space complexity vs accuracy of the classifier. The second model performing slightly worse on the development set does not mean it would necessarily perform worse on the test sets.