Nicholas McKillip
CS-489 Huang

Programming Assignment 1

1. To run the code please note that I have used python**3**
   Have the data directory nearby (not included for email size constraints)

cd python
python**3** NaiveBayes.py ../data/imdb1
OR
Python**3** NaiveBayes.py -f ../data/imdb1
OR
python**3** NaiveBayes.py -b ../data/imdb1

2. Results

**Multinomial Naïve Bayes**

```
Nicks-MacBook-Pro-2:python nickmckillip$ python3 NaiveBayes.py ../data/imdb1
[INFO]  Fold 0 Accuracy: 0.765000
[INFO]  Fold 1 Accuracy: 0.850000
[INFO]  Fold 2 Accuracy: 0.835000
[INFO]  Fold 3 Accuracy: 0.825000
[INFO]  Fold 4 Accuracy: 0.815000
[INFO]  Fold 5 Accuracy: 0.820000
[INFO]  Fold 6 Accuracy: 0.835000
[INFO]  Fold 7 Accuracy: 0.820000
[INFO]  Fold 8 Accuracy: 0.755000
[INFO]  Fold 9 Accuracy: 0.840000
[INFO]  Accuracy: 0.816000
```

**Multinomial Naïve Bayes Stop Words Removed**

```
Nicks-MacBook-Pro-2:python nickmckillip$ python3 NaiveBayes.py -f ../data/imdb
[INFO]  Fold 0 Accuracy: 0.765000
[INFO]  Fold 1 Accuracy: 0.825000
[INFO]  Fold 2 Accuracy: 0.815000
[INFO]  Fold 3 Accuracy: 0.830000
[INFO]  Fold 4 Accuracy: 0.795000
[INFO]  Fold 5 Accuracy: 0.830000
[INFO]  Fold 6 Accuracy: 0.835000
[INFO]  Fold 7 Accuracy: 0.835000
[INFO]  Fold 8 Accuracy: 0.760000
[INFO]  Fold 9 Accuracy: 0.820000
[INFO]  Accuracy: 0.811000
```

**Binarized Multinomial Naïve Bayes**

```
Nicks-MacBook-Pro-2:python nickmckillip$ python3 NaiveBayes.py -b ../data/imdb1
[INFO]  Fold 0 Accuracy: 0.805000
[INFO]  Fold 1 Accuracy: 0.835000
[INFO]  Fold 2 Accuracy: 0.835000
[INFO]  Fold 3 Accuracy: 0.820000
[INFO]  Fold 4 Accuracy: 0.835000
[INFO]  Fold 5 Accuracy: 0.825000
[INFO]  Fold 6 Accuracy: 0.845000
[INFO]  Fold 7 Accuracy: 0.835000
[INFO]  Fold 8 Accuracy: 0.790000
[INFO]  Fold 9 Accuracy: 0.855000
[INFO]  Accuracy: 0.828000
```

**What other features could be relevant for sentiment analysis?**

Bigram and Trigram could be useful for sentiment analysis. For example: the bigram "not fun" is much more telling than just seeing "fun" and "not" separately and not understanding the relationship. The occurrence of bigrams or trigrams is not independent from Unigrams, but they are more informative and could even be used in conjunction.

Another feature to consider is the part of speech weighting. For example, you could give adjectives more weight than nouns because the presence of "fantastic" or "awful" is likely to be more telling than "him".

You could also consider the inverse document frequency and give more weighting to uncommon words. The effect of this is to reduce the weights of what could be considered stop words, but to increase the weights unique and likely more informative words.

**Analysis**

Naïve Bayes performed well with 81.6% accuracy.

When I removed the stop words the accuracy dropped slightly to 81.1%. I expect that this is because we are throwing away information. Although we would expect these words they are apparently informative. It seems that it is a good practice not to throw away information unless we have a strong reason to do so.

Binarized Naïve Bayes performed noticeably better achieving 82.8% accuracy. My intuitive explanation for this would that the occurrence of a positive word tells you a lot about the document, but the fact that it occurred a certain number of times does not tell you more and is in fact noise that naïve Bayes can overfit to.

**To my knowledge there are no bugs in my program.**