# COM 3110: Sentiment Analysis of Movie Reviews

December 2020

## 1 Introduction

There are 8 models implemented for the naive Bayes classification of the sentiment of movie reviews.The first model is the 3 class model with no pre-processing, the 2nd is the 5 class model and the last 6 models are the first and 2nd models with pre-processing applied.

For pre-processing, a stop-list from the NLTK module is used and then words ared added to the stop-list from the top 50 most common words in the train data which i think are too general with regards to their occurrence in movie reviews, and their presence will negatively impact the likelihood probability of words that would help classification. These words tend to be genre names like 'thriller' or 'action' , names of people like 'john' and 'David', and numbers like 'two' or '3'. 'movies' and 'films' were also removed instead of 'movie' and 'film' because the former are more general, and there are greater polarity to the preceding words of the latter. Someone could express "i like this film" which would be conveyed as a positive sentiment whereas films would mostly be used in a context like " out of all films, i dislike/like this film the most", which would not really help to classify a review by its occurrence.

Words like 'make', 'makes','made' were also removed because they could equally be used to convey sentiments in all classes, same as 'plot' which would generally have negative, positive or neutral sentiments prepended to it instead of only a single sentiment which would aid in classification.
By removing the above-mentioned words, more weight will be given to words in the train data that are more effective in differentiating between classes. Most punctuation were also removed except for '!'. Additionally the top 20 least common features from the train data were removed because i find that the least common features tend to be names that are only specific to a particular movie and would not help in classification, and removing them makes not discernible difference to my results. Thus, for 3 class classification and 5 class classification, each specification consists of three models one with a stop-list,one with 20 least common features removed and the last one with the stop-list and 20 least common features removed.

## 2 Implementation

A model class was created to process the tsv files and to create dictionaries.
The dictionaries created depending on parameters fed into the model, are {id:sentiment},{id:{word:count}} for the dev data, and {id:sentiment}, {id:{word:count}} and a set of all unique words of the train data. A {id:{word:count}} dictionary of the test data was also created.

The Bayesian class is used to calculate the priors for each class and the likelihood of a word given a class, and then the classify class is used to calculate the posterior probabilities of a class given a review by first finding the joint likelihood of words in a review and then multiplying it by the prior of a class derived from the train data.

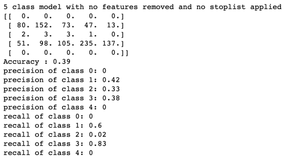
This method is used to find the posterior probabilities of a class given a review for the dev and test data after finding the likelihood of a word given a class and the prior probability of a class in the train data, and storing the posterior probabilities for each class in an array.
Array is stacked and argmax is used to find the class with the greatest posterior probability. The predicted class label will then be stored in an array in the results attribute of the classify class and then compared with the gold label results of the dev data in a confusion matrix. A function was created to calculate the

precision, accuracy and recall of the confusion matrices.

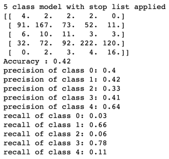
The code is stored in a notebook, and to run the code just run every input box from the beginning. Each mode will have its own input box. Near the end of the notebook there are 8 models and the model that utilizes the stop list and the removal of 20 least common features is used to classify the test data.

# 3 Results

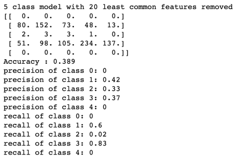
## 5 class models



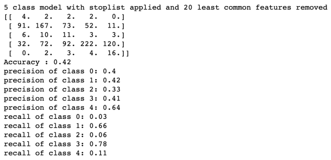
```
5 class model with no features removed and no stoplist applied
[[  0.   0.   0.   0.   0.]
 [ 80. 152.  73.  47.  13.]
 [  2.   3.   3.   1.   0.]
 [ 51.  98. 105. 235. 137.]
 [  0.   0.   0.   0.   0.]]
Accuracy : 0.39
precision of class 0: 0
precision of class 1: 0.42
precision of class 2: 0.33
precision of class 3: 0.38
precision of class 4: 0
recall of class 0: 0
recall of class 1: 0.6
recall of class 2: 0.02
recall of class 3: 0.83
recall of class 4: 0
```



```
5 class model with stop list applied
[[  4.   2.   2.   2.   0.]
 [ 91. 167.  73.  52.  11.]
 [  6.  10.  11.   3.   3.]
 [ 32.  72.  92. 222. 120.]
 [  0.   2.   3.   4.  16.]]
Accuracy : 0.42
precision of class 0: 0.4
precision of class 1: 0.42
precision of class 2: 0.33
precision of class 3: 0.41
precision of class 4: 0.64
recall of class 0: 0.03
recall of class 1: 0.66
recall of class 2: 0.06
recall of class 3: 0.78
recall of class 4: 0.11
```



```
5 class model with 20 least common features removed
[[  0.   0.   0.   0.   0.]
 [ 80. 152.  73.  48.  13.]
 [  2.   3.   3.   1.   0.]
 [ 51.  98. 105. 234. 137.]
 [  0.   0.   0.   0.   0.]]
Accuracy : 0.389
precision of class 0: 0
precision of class 1: 0.42
precision of class 2: 0.33
precision of class 3: 0.37
precision of class 4: 0
recall of class 0: 0
recall of class 1: 0.6
recall of class 2: 0.02
recall of class 3: 0.83
recall of class 4: 0
```

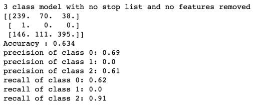


```
5 class model with stoplist applied and 20 least common features removed
[[  4.   2.   2.   2.   0.]
 [ 91. 167.  73.  52.  11.]
 [  6.  10.  11.   3.   3.]
 [ 32.  72.  92. 222. 120.]
 [  0.   2.   3.   4.  16.]]
Accuracy : 0.42
precision of class 0: 0.4
precision of class 1: 0.42
precision of class 2: 0.33
precision of class 3: 0.41
precision of class 4: 0.64
recall of class 0: 0.03
recall of class 1: 0.66
recall of class 2: 0.06
recall of class 3: 0.78
recall of class 4: 0.11
```

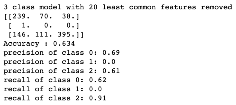
As depicted by the confusion matrices of the 5 class models, The 5 class model with no stop list or feature selection applied scored zero on recall and precision for classes 0 and 4, and and 2% recall for class 2, which seems to imply that there are less opinions that fall on the extreme ends of the sentiment scale and in the middle.

The removal of 20 least common features seem to have have no discernible effect on the classification of the dev data. Accuracy increases to 42% when a stop-list is applied, with recall, and especially precision increasing quite substantially for classes 0, 4 and 2. With a stop list and the least common 20 words removed, the accuracy of classification remains at 42% which shows that the stop-list is more effective at increasing classification accuracy.

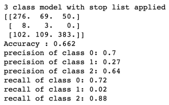
## 3 class models



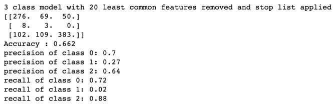
```
3 class model with no stop list and no features removed
[[239.  70.  38.]
 [  1.   0.   0.]
 [146. 111. 395.]]
Accuracy : 0.634
precision of class 0: 0.69
precision of class 1: 0.0
precision of class 2: 0.61
recall of class 0: 0.62
recall of class 1: 0.0
recall of class 2: 0.91
```



```
3 class model with 20 least common features removed
[[239.  70.  38.]
 [  1.   0.   0.]
 [146. 111. 395.]]
Accuracy : 0.634
precision of class 0: 0.69
precision of class 1: 0.0
precision of class 2: 0.61
recall of class 0: 0.62
recall of class 1: 0.0
recall of class 2: 0.91
```



```
3 class model with stop list applied
[[276.  69.  50.]
 [  8.   3.   0.]
 [102. 109. 383.]]
Accuracy : 0.662
precision of class 0: 0.7
precision of class 1: 0.27
precision of class 2: 0.64
recall of class 0: 0.72
recall of class 1: 0.02
recall of class 2: 0.88
```



```
3 class model with 20 least common features removed and stop list applied
[[276.  69.  50.]
 [  8.   3.   0.]
 [102. 109. 383.]]
Accuracy : 0.662
precision of class 0: 0.7
precision of class 1: 0.27
precision of class 2: 0.64
recall of class 0: 0.72
recall of class 1: 0.02
recall of class 2: 0.88
```

As depicted by the confusion matrices, the accuracy of the classification seems to be more impacted by the implementation of the stop list and less by the removal of 20 features. Precision and recall of class zero is zero with no stop list and/or feature selection applied.

Removing 20 least common features have no effect on class 0's precision and recall but applying a stop-list have a more noticeable effect on class 0, with precision increasing to 27% and recall increasing from 0 to 2%. Like in the 5 class model, the class in middle of the sentiment spectrum seem to attain the worst recall and precision results, and this seems to suggest that most casual movie goers tend to be less ambivalent in their reviews and either like or dislike a movie but not in an extreme fashion and thus there are less neutral or extreme features available in the training data to train the classifier with regards to neutral and extreme sentiments.

The naives bayes classifier seems to work best when the likelihood of a feature occurring in a class is noticeably greater compared to the likelihood of the same feature occurring in another class. By removing features that have contradictory sentiments or features that tend to not have a sentiment value especially in the context of movie reviews, it will increase the likelihood value of a helpful feature and increase classification accuracy.