# Report for Part A:

Dataset: Algerian\_forest\_fires\_dataset, <https://archive.ics.uci.edu/ml/datasets/Algerian+Forest+Fires+Dataset++>

Average accuracy: 0.7878688524590163

Average probability for correct: 0.9825308002204249

Average probability for incorrect: 0.9497672714945602

The average accuracy score for this Naïve Bayes algorithm was worse than the one for Decision Trees, however it was better than the one for kNN. This could be because the Naïve Bayes algorithm assumes all items follow a normal distribution whereas the Decision Trees make no such assumption. The reasoning for it doing better than the kNN algorithm could be because I messed up the kNN algorithm.

# Report for Part B:

## Data Description

The Reuters-21578 Corpus was used for the classification tasks. Those 5 tasks were Topics earn vs not earn, Topics coffee vs not coffee, Topics crude vs not crude, Places usa vs not usa, Places egypt vs not egypt. In 21578 documents there are, 3987 where Topics include earn, 145 where Topics include coffee, 634 where Topics include crude, 12541 where Places include usa, and 49 where Places include egypt.

## Description of Text Representation

The 3 things changed for each version were using stems vs words, removing stop words or not, and using MultinomialNB or ComplementNB, with all combinations of these to make up the 8 versions. Thus the 8 versions were: no stemming or word removal with ComplementNB, stemming with no word removal using ComplementNB, no stemming with word removal using ComplementNB, no stemming or word removal with MultinomialNB, stemming and word removal with ComplementNB, stemming with no word removal using MultinomialNB, no stemming with word removal using MultinomialNB, and stemming with word removal using MultinomialNB.

## Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Confusion Matrix | Accuracy | Precision | Recall |
| no stemming  no word removal  ComplementNB | [[20865 1689]  [ 1285 3136]] | 0.88975 | 0.94199 | 0.92511 |
| stemming  no word removal  ComplementNB | [[16257 6297]  [ 233 4188]] | 0.75792 | 0.98587 | 0.72080 |
| no stemming  word removal  ComplementNB | [[20840 1714]  [ 1284 3137]] | 0.88886 | 0.94196 | 0.92400 |
| no stemming  no word removal  MultinomialNB | [[20870 1684]  [ 1284 3137]] | 0.88997 | 0.94204 | 0.92533 |
| stemming  word removal  ComplementNB | [[16259 6295]  [ 233 4188]] | 0.75800 | 0.98587 | 0.72089 |
| stemming  no word removal  MultinomialNB | [[20494 2060]  [ 1267 3154]] | 0.87666 | 0.94178 | 0.90866 |
| no stemming  word removal  MultinomialNB | [[20848 1706]  [ 1283 3138]] | 0.88919 | 0.94203 | 0.92436 |
| stemming  word removal  MultinomialNB | [[20498 2056]  [ 1267 3154]] | 0.87681 | 0.94179 | 0.90884 |

## Discussion

Overall, the version with the best performance was 4, no stemming or word removal with MultinomialNB. However, the versions with the best precision were 2, stemming with no word removal using ComplementNB, and 5, stemming with no word removal using MultinomialNB. Judging from the results, stemming decreased the accuracy and recall but slightly improved precision, while stop word removal slightly decreased all 3, and overall MultinomialNB did better than ComplementNB. Therefore, I would recommend version 4, no stemming and no word removal using MultinomialNB.

## Future Work

More text representations I would do would be removing frequent words or term frequency-inverse document frequency or even using bags of n-grams instead of bag of words/stems. These would take up a lot more time and use more RAM than those I did which is why I did not implement them now.