Cera Oh
ML
Winter 2021
2/10/2022

# Programming 2 Report

## Introduction

For this assignment, I implemented and used the Gaussian Naïve Bayes on the Spambase data from the UCI ML repository to classify whether an email was a spam or not. The data set is made up of 4601 rows of data, each row of data containing 57 features representing word frequencies of certain common words found in emails and a class label. The data set contained 1813 spam email data (39.4%) and 2788 non-spam email data (60.6%). The data set was shuffled and split into training data and test data using train_test_split function of sklearn library in Python. Then, the prior probability of spam and non-spam classes are calculated using the training data. For each feature, the mean and the standard deviation for each of spam and non-spam classes of the training data were calculated and saved as arrays. Any zero standard deviation value was replaced with 0.0001 to avoid division by zero. The Gaussian Naïve Bayes algorithm was implemented and used on the test data to determine whether each sample was a spam or not. The equation for the probability density function was given in the assignment sheet and was used to calculate each of the likelihood of feature given a class. Log division by zero was avoided by assigning a very small epsilon value, $10^{-100}$, for $e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}$ if it was rounded to 0. I then used the logarithms to avoid underflows in the classification method. A sample was predicted to be a spam if log of prior probability of spam added to the sum of log of all likelihood of features given the spam class were greater than log of prior probability of non-spam added to the sum of log of all likelihood of features given the non-spam class. Prediction was non-spam otherwise. Accuracy, precision, recall, and confusion matrix were calculated from the results.

## Results

Test Accuracy: 0.8169565217391305

Test Precision: 0.6887796887796888

Test Recall: 0.953514739229025

Test Confusion Matrix

| | **Spam** | **Non-Spam** |
|---|---|---|
| **Spam** | 841 | 41 |
| **Non-Spam** | 380 | 1038 |

**Analysis**

The test accuracy was about 81.7%, which is not too bad. Making the epsilon value smaller improved accuracy. Test precision was about 68.9% and test recall was about 95.4%. Precision is the ratio of correct positive examples to the number of actual positive examples. This shows that most of spam mails in the test data set were correctly labeled as spam. Recall is the ratio of the number of correct positive examples out of those that were classified as positive. Recall was low, indicating that many of the non-spam data were labeled as spam. The confusion matrix also shows that the Gaussian Naïve Bayes algorithm was able to pick out most of the spam mails from the sample, correctly labeling 841 mails as spam and only failing to mark 41 of the spam mails as spam. This shows that 841/(841+41) = ~95.4% of the spam emails were correctly labelled. However, the algorithm was only able to correctly label 1038 samples as non-spam, labelling correctly only 1038/(1038+380) = ~73.2% of the non-spam mails correctly. Overall, Gaussian Naïve Bayes performed decently but not as good as I expected it to do.

**Do you think the attributes here are independent, as assumed by Naïve Bayes?** I don't think the attributes here, which are the frequencies of certain words, are as independent as assumed by Naïve Bayes. Here, we are assuming word frequencies are independent of one another and what matters is how often certain words show up in a spam or a non-spam email. Combinations of words matter in conveying the meaning of the message. Same or similar word frequencies may be used in both a spam message as well as non-spam message because what matters is the arrangement of the words not the words by themselves. Therefore, I don't think the attributes are as independent as assumed.

**Does Naïve Bayes do well on this problem in spite of the independence assumption?**
Since Gaussian Naïve Bayes was able to reach 81.7% accuracy, I think Gaussian Naïve Bayes do well on this problem in spite of the independence assumption. However, I think it would

probably do better if it did not assume independence and maybe if we gave meanings to frequencies of group of words in an email instead of individual words. It may also do better if we included more attributes besides 57 attributes we had. It also may do better with bigger training set.

**Speculate on other reasons Naïve Bayes might do well or poorly on this problem.**

I believe Naïve Bayes will have done better with bigger training size, more attributes to consider, and using frequencies of group of words as attributes instead of considering frequencies of individual words. I believe 57 attributes is too small and having larger attributes will help the algorithm tell the difference between spam and non-spam better. Also, bigger training data size will allow for better mean and standard deviation calculations, which will allow better classification. Considering frequencies of group of words may improve performance over considering only frequencies of individual words, since I don't think the attributes are independent of each other.

**Conclusion**

In conclusion, the Gaussian Naïve Bayes performed decently, getting 81.7% accuracy. Its recall was high at 95.4%, so it classified most spams correctly. However, it's precision was lower than ideal, at 68.9%. This shows that many non-spam emails were labelled incorrectly. Including more attributes or considering frequencies of group of words instead of individual words may increase the performance of the Gaussian Naïve Bayes algorithm.