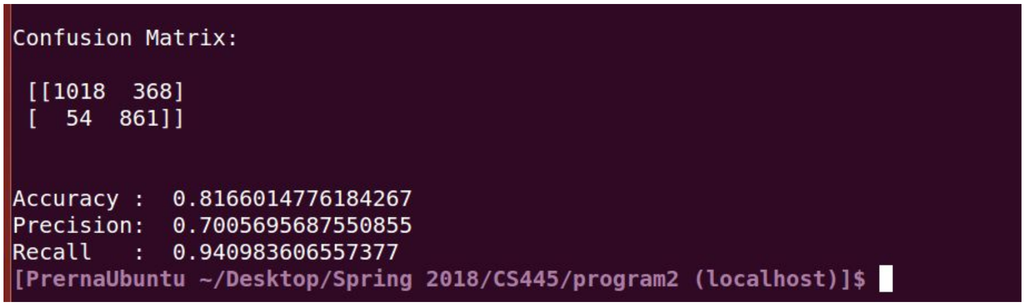
Prerna Agarwal
CS 445

Programming Assignment 2 Report

Description:
In this assignment, I was supposed to implement the Gaussian Naïve Bayes algorithm to classify the spam and non spam emails.

I first got the data, thoroughly shuffled the instances and then split it into two halves, a training set and a test set. Each of these have about 2,300 instances, and each have about 40% spam, 60% not spam. Further, each feature is independent of all others, I did not standardize the features.
Then for each of the 57 features, I compute the mean and standard deviation in the training set of the values given for each class. After that, if any of the features has zero standard deviation, I assign it a “minimal” standard deviation (e.g., 0.0001) to avoid a divide-by-zero error in Gaussian Naïve Baye. Lastly, I calculated the confusion matrix, accuracy, precision and recall values.



```
Confusion Matrix:

 [[1018  368]
 [  54  861]]


Accuracy :  0.8166014776184267
Precision:  0.7005695687550855
Recall   :  0.940983606557377
[PrernaUbuntu ~/Desktop/Spring 2018/CS445/program2 (localhost)]$
```

I got about 81.6% accuracy, 70.05% precision and 94.09% recall values. The confusion matrix shows that for most part it is able to classify spam and non spam emails. However, the accuracy is still not great, it is about 81.6%.

Do you think the attributes here are independent, as assumed by Naïve Bayes?

Our assumption is that all the attributes are mutually independent in Naive Bayes so that we can multiply the class conditional probabilities in order to compute the outcome probability. However, there can be few dependencies and some of the attributes can be correlated, it is easy to ignore one of the correlated attributes. However, the classifies does not know which attributes are dependent on one another.

Does Naïve Bayes do well on this problem in spite of the independence assumption?
We can see from the result, the Naive Bayes does alright. It would have been a great model if it gave us around 90% accuracy however, it gives 80% accuracy. We can improve this model by removing redundant data i.e., correlated attributes and use probabilities for feature selection i.e., selection of those data attributes that best characterize a predicted variable. We can use a

search algorithm and find the combination of the probabilities of different attributes and evaluate their performance for predicting the output variable.