COMP 6721
Applied Artificial Intelligence
Winter 2020

Project 2 Report

Professor Rene Witte

Team Members:
Ankit Jana - 40059380
Sandeep - 40082052
Van Duong - 40075017

Analysis

The purpose of our program is to classify file content as spam or ham using Naive Bayes model. There are several steps in our implementation. Firstly, a vocabulary set of words is built based on the given training files. Secondly, a dictionary of spam words and a dictionary of ham words, which keeps track of the words and their occurrence, are also built based on the same training files. Thirdly, the probability of spam and ham files (prior) are computed. Additionally, the probability of each word in its corresponding class is calculated. A smoothing constant of a =0.5 is applied to each word probability to avoid zero probability, as followed:

$$P(w_i \mid c_j) = \frac{(frequency\ of\ w_i\ in\ c_j + a)}{(total\ number\ of\ words\ in\ c_j + a * vocabularySize}$$

After all the training files have been processed and "learned", testing files are processed to test the program's performance. For each test file, the sum of log base 10 of probability of all the words in each class is calculated. The file is then classified to the class which has greater probability.

The evaluation metrics to measure the program performance include accuracy, recall, precision, and F-1 measure. They are calculated as followed:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
$$precision = \frac{TP}{TP + FP}$$
$$recall = \frac{TP}{TP + FN}$$
$$F - 1 = \frac{2(precision * recall)}{(precision + recall)}$$

The results shown in the table below were obtained from running our program.

		Model's prediction	
		Spam	Ham
Expected Results	Spam	357	11
	Ham	43	389

Accuracy = 0.9325; Precision = 0.9701; Recall = 0.8925; F-1 = 0.9297

Based on the results, our model's performance is quite accurate and precise, which is partly due to the large number of training files (997 spam files, 1000 spam files). Although Naive Bayes is a very simple classification model, its performance is surprisingly good.

References

1. Karmali, T. (2017, August 3). Spam Classifier in Python from scratch. Retrieved from https://towardsdatascience.com/spam-classifier-in-python-from-scratch-27a98ddd8e73
2. Professor Rene Witte's lecture notes on Naive Bayes Classification