Spam Classifier

Introduction

Our project is meant to imitate or even improve upon current spam email classifiers based off of our older homeworks such as the TF-IDF for query and documents. We used feature vectors and with a lot of help from Python's Sklearn library to perform the Naive Bayes multinomial algorithm and the Support Vector Machine algorithm to help find out which algorithm is best used to classify spam. We used a compilation from Ling-spam (http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz) corpus to generate a training folder with half spam and half ham to train our system. Once trained, we ran a test batch to see how our program ran.

Run & Evaluate

To run our program run:
Python3 spamfilter.py

Output will be in the form of:

Algorithm	Ham	Spam
Ham	x	x
Spam	x	x

Techniques

We implemented feature vectors to calculate the features of a 2d matrix, using rows as emails and columns as common words, moreover we used the stop words from the query and documents homework so that we do not use common words in our dictionary to give us false keywords. Algorithms we currently have implemented are the Naive Bayes Multinomial and Support Vector Machine Algorithm.

Short description of future plans

We want to move over from technical to analytical to further understand what people classify as spam such as emails that are scams and emails that are undesirable based on the user. We want to be able to implement a spam filter which understands the type of “person” that is using it as to further more accurately filter desirable and undesirable emails. Furthermore, we want to the errors between the two algorithms we have so far such as the risk of flagging ham emails from spam and how to mitigate these errors by using the analysis we mentioned above.