Institute of Technology Carlow
Kilkenny Road,
Carlow
Bsc (Hons) Cybercrime and IT Security

The Spam Catcher
Final Report

Author: Aya Aloraimi
Student Number: C00212086
Supervisor: James Egan

Abstract

The purpose of this project is to develop a web application where users can upload email messages that they received in mailbox into the web application to check if they are spam or ham by using Naïve Bayes classifier. The web application should get result back to the user and review old scans in the same page.

Table of Contents

Abstract ***2***

Introduction ***5***

Description of the final project ***6***

Development challenges ***8***

Python **8**

Machine learning **8**

Conformance to Specification and Design ***9***

Learning outcomes ***10***

-Technical outcomes **10**

-Personal outcome **10**

Project Review ***11***

- **Project success and achievements** **11**
- **Project Failures** **11**
- **Future Development** **11**

Acknowledgements ***12***

Table of Figures

FIGURE 1- CONTENT AND HEADER ANALYSIS 6
FIGURE 2- SCANNING HISTORY PAGE 7

Introduction

This document represents the final project report of the spam catcher web application. It will outline various topics such as description of the final project, technologies used to develop this project along with difficulties. Also it will highlight how the implementation compared to the original specification and design for the spam catcher web application. This report will provide technical and personal achievements and also review of project, what went right, what went wrong, what is left to do and what would be done differently.

Description of the final project

The spam catcher web application is the final year project of Aya Aloraimi under the supervision of James Egan. The project was built using Python language, Scikit-Learn and Flask framework. The user interface was built using a mark-up language like HTML and CSS to display contents in a web application. Also SQLite was used to provide local data storage for this web application.

The aim of this project was to develop a web application that identifies spam emails using machine learning algorithms like Naïve Bayes classifier. This web application contains different features to identify spam emails such as uploading .txt and .zip files, copy and paste the content of email message and also using email header features (To, From, Subject and Message ID) to classify email messages. In this web application, the user should register to get access to the system and then they can upload an email message and scan it to see the result.

Since spams are originally designed to trick people so spams are already had some typosquatting behaviours on the sender's email as well as some linguistical pattern in the content. The system captures those linguistic characteristics and calculates the probability of being a spam email using a multivariate naive Bayes classifier on both headers itself as well as the content of the email.

The process of filtering spam emails using Naïve Bayes classifier is not really complicated, In general, I got a large number of spam emails from Enron dataset to be read into pandas dataframe and then I tokenized them to words like “your income” and build a huge dictionary of the words which spam emails contain. After that I went through the email header and content and count the word frequency of that words if those have appeared on that given email, using the feature vector I calculate the probability of being spam or ham using the Naïve Bayes classifier.

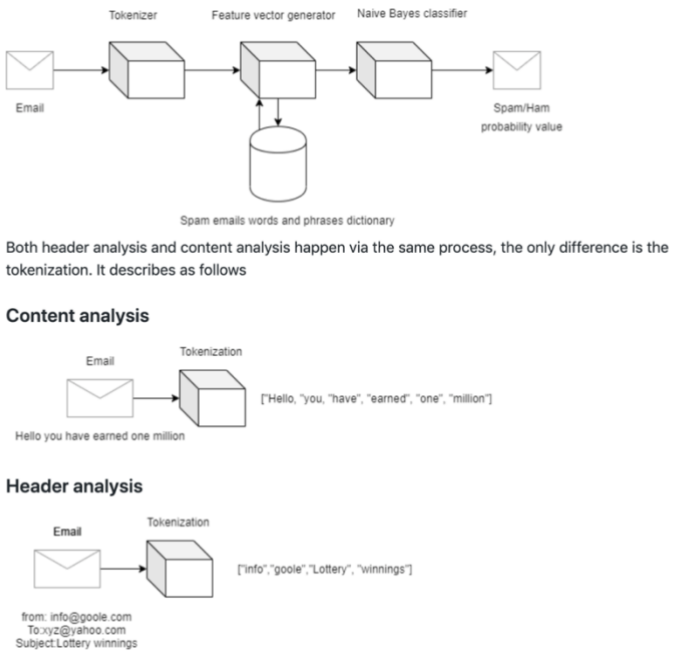

Figure 1- content and header analysis

For example, in the above Figure, assume our spam phases dictionary has five words like ["free," "guarantee," "opportunity," "earned," "million,"] And given email contains a message "Hello you have earned one million". Then tokenize the email content/header into tokens "Hello you have earned one million" -> ["Hello, "you, "have", "earned", "one", "million"] Now we compare with spam phases dictionary and make the feature vector so in this case, it will be like this [0,0,0,1,1] Then we plug that value to the trained NB classifier and take the probability value out.

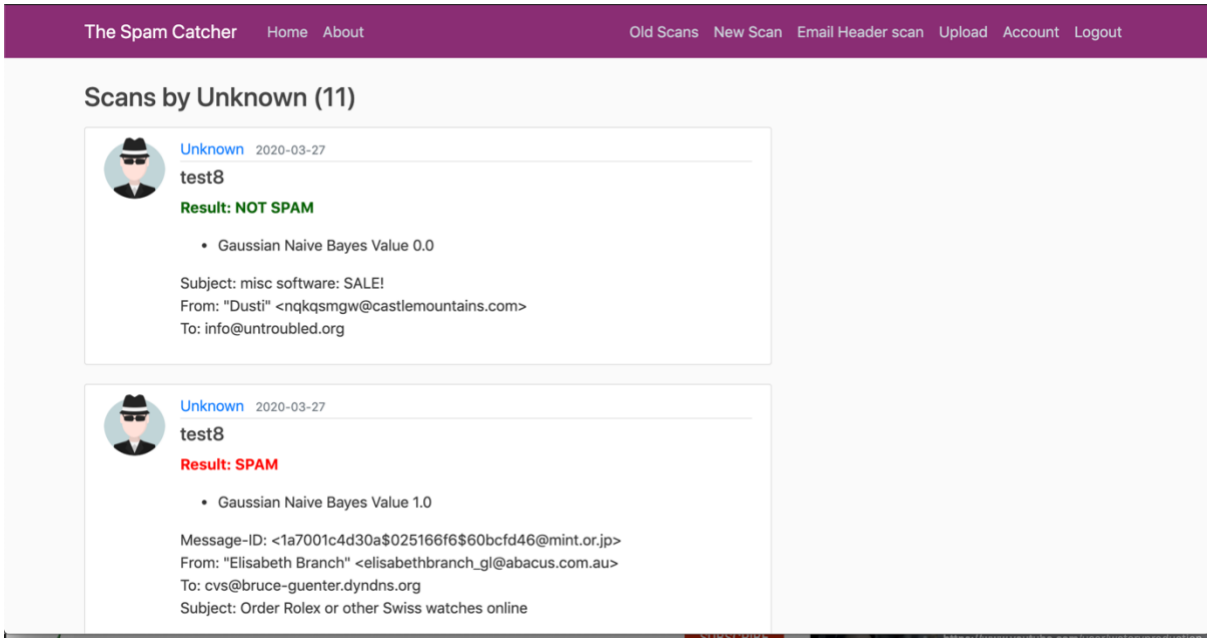

Figure 2- scanning history page

In this web application the user has to register first and then click on ‘New Scan” to scan their email message. The user can use either email header scan or content scan to check email messages that they receive in mailbox. If the email message is grater than 0.5 it will be spam and if it is less than 0.5 this mean that the email message is not spam.

Development challenges

Python

At the beginning I was planning to develop the web application using Java language because I have used it many times during my course. However, after doing my research manual I found it better to develop the web application with Python language as I am using machine learning algorithms in this project. Developing a web application with a new language that I have never used it before is a big challenge for me. Learning a new programming language has a value and it needs lots of time and by using lots of resources such as Stack Overflow, and online tutorials help me get started and speed along the process. Furthermore, looking for other projects related to my topic is very useful when implementing my project.

Machine learning

Since this project is based on using machine learning algorithms, it was another challenge for me as I have to implement Naïve Bayes classifier in the web application to classify email messages into spam and ham. I used online resources to understand the concept of Naïve Bayes classifier and how it can be implemented in Python.

Conformance to Specification and Design

The spam catcher web application meets its design specifications successfully. There were a number of features to be added into the web application. They were as follows:

- User interface: by displaying user interface, a user can interact with the system directly.
- Register and login: the user should be able to register or login to the system in order to perform different tasks.
- Apply Naïve Bayes classifier into the system to classify email messages
- An option to scan an email message by using the header features of an email such as subject, to, from and message ID to classify an email message as spam or ham.
- Upload a file: the user should be able to upload a file (email messages) whether it is .txt or .zip
- Enron dataset: in this project we are going to use Enron dataset which contains huge amount of real email messages for testing purposes
- Scan history : the system should save user's results after each scan.
- Generic message: regarding to user login page if a new user tries to login and he/she is not a member, a generic message should be displayed to order to inform the user for registration
- Web front end : the system should runs as a web application which should be developed using, HTML and CSS

There was a small change made to specification during implementation such as adding another option which is scan email header. This option was added to scan email header features like subject, message ID, from and to. The idea of designing the graphical user interface (GUI) remained true and nothing new was added to it.

Learning outcomes

-Technical outcomes

Over the course of the project, many new technologies and programming language had to be learned during implementation. As mentioned in development challenges, creating a web application using Python was not difficult but the programming language itself was new for me as I haven't had any work with Python in previous years. The majority of the project's development time was taken up attempting to learn and develop in Python.

Understanding how machine learning algorithms work was the next significant learning experience as I had no knowledge of machine learning . After doing my own research to identify the best algorithms that are used in classifying email messages, I chose one type of supervised machine learning algorithm which is Naïve Bayes classifier that uses Bayes' Theorem. The theorem relies in the naïve assumption that input variables are independent of each other. This classifier has many types and each type has different concept. I chose Multinomial Naïve Bayes as it can be used in identifying email messages into spam and ham. Through my research I identified Scikit-learn To implement this algorithm which is a free machine learning library for Python.

-Personal outcome

Over the course of this project, the first lesson I have learned is to make plans when creating any project to be more organized and time will be managed as well. A final year project contains many challenges as I faced many issues during my development such as implementing machine learning algorithms codes which was hard for me but I overcome this issue by searching and obtaining additional online resources.

Another lesson that I have learned is time management, controlling time spent on developing my final year project and doing other college studies was a challenging experience. It is necessary to an individual that they learn how well they can handle pressure. I have learned to balance my time by using a daily schedule template or set a daily goal to plan my day which help motivate me to do more.

A great lesson that I have learned is that everyone has a different perspective so it is important to listen to advice and feedback from others especially from the project supervisors which was helpful to correct mistakes and improve this project.

Project Review

- Project success and achievements

The spam catcher was a success as all features of the project in the functional specification were achieved. The user can register and login to use the features of this web application to scan their email messages such as scanning the content and header of an email message. In addition, scanning an email message in a text file was implemented successfully as well in the spam catcher web application.

- Project Failures

The main thing is that I had very little knowledge of Python programming language and machine learning algorithms. Therefore, I wen t wrong when start coding this web application and I was really confused but after watching some tutorials and reading various researches I had the ability to found errors and correct them to make this web application runs perfectly. At the beginning I attempted to get the accuracy score for many classifiers and display them in the user interface of the spam catcher web application but it did not work.

- Future Development

There was additional work remains to be done with this web application such as making the front end looks significantly more modern, adding another classifier which is the logistic regression classifier to get a percentage between 0 and 1 and then compare the two different results from each classifier. Also, there is another feature that should be added in the spam catcher which is false positive spam detection rates. This is something that could be addressed in future development.

Acknowledgements

I would like to thank my supervisor James Egan, whose feedback and advices was helpful throughout the course of this project. Also I would like to thank all lecturers over the four years I spend in the college for their time and knowledge they shared with us.