CLASSIFYING EMAILS USING MACHINE LEARNING FOR ANTI-SCAM PURPOSES

Candidate Number: 184521

Contents

Acknowledgements

I would like to extend my gratitude to my advisor Julie Weeds for helping me throughout the process of this report, as well as to Paula for supporting me emotionally whilst developing the solution to this project.

I also wish to thank my family and friends for helping me conduct the necessary research and providing support when asked for, as well as the 64 participants that took part in the Questionnaire and Testing.

Summary

These days people have to deal with fraudulent emails on a day-to-day basis, so a solution to separate these from users' actual emails is required. Therefore, this report sets out the design and implementation of research of a bot which helps users classify emails by using Natural Language Processing techniques along with Machine Learning. Although these techniques are already widely used, this report compares them against each other to show their strengths and weaknesses in an effort to ensure that more vulnerable people are less successfully targeted by using an easy-to-use solution which aims to perform just as well if not better than other papers or solutions. This is done by researching various detection methods as well as conducting questionnaires on how people classify emails to better replicate them and give this technical knowhow to others by having them use the program. This is done with a variety of classifiers such as Naïve Bayes, SVM's, Random Forests and C5.0 algorithm classifiers tried in different configurations and combinations with some performing a lot better than others. This report therefore covers how they perform, how people classify these emails themselves and an in-depth view of how the final solution was implemented.

Introduction

Scam emails are defined as fraudulent emails which are created by criminals with the intent of preying on the vulnerable for their data or money. Unfortunately, this type of email have become a daily occurrence in our everyday lives; spam emails accounted for 45% of all emails sent in 2018 [1], with this number only increasing since, leading to 76% of businesses falling victim to a scam within 12 months [2]. These larger businesses have IT support and security personnel employed to stop breaches such as this from occurring and yet they still get caught out. If this is possible, then imagine how often less informed people are successfully targeted by the criminals that run the scams.

Before beginning this project, solutions to this problem ranged from anti-virus software requiring an annual subscription to handcrafted solutions found on places such as GitHub which are complicated to set up. This therefore leaves people having to either spend a lot of money or require a complex understanding of programming languages in order to configure parameters for a custom anti-spam bot. The former option is chosen most often, with costs adding up quickly; the average price of a subscription being $40+ annually [3]. Whilst there are free anti-virus software options available, from analysing user ratings and reviews they do not perform as well as the previously mentioned options.

Therefore, in this project an Artificial Intelligence program has been researched, designed and created that is able to tell if a given email is Scam, Spam or neither, also known as Ham. The program functions by accessing an inbox to which the user has forwarded their emails and retrieving the required data from it. This data is then processed using different Natural Language Processing (NLP) techniques to allow classifiers to conclude what category the email falls into. The recipient is then informed of the results of the program. This therefore presents an elegant and easy-to-use solution for multiple audiences, even for those people who are less technologically inclined, as all that is required by the user to operate the program is the ability to send an email to the correct address and await the response.

This project has been able to test the ease of use and performance of the program by seeing if the created program performs better than included anti-spam bots found in other email services as well as invoking Initial User Tests. The former was done simply by testing the Accuracy and other measures of the created software and comparing if they were higher than that found in established email services. This is in addition to comparing the Accuracy of the users in detecting scam and spam emails by themselves using a dataset created from Questionnaire responses. Therefore, the aim of this project was completed to the best degree possible whilst keeping functions and features in scope. The original objective was to try to create a detection and classification algorithm that can work better than that of any other readily available classifiers used in most inboxes. The reason for this was to help vulnerable people to be less successfully targeted along with any other users of the service.

The following report details the research and design of the solution, as well as how the final solution was implemented in the Related Work and Design & Implementation sections. Following this, extensive research was done as to how different classifiers perform, how participants of a questionnaire shaped the design and how these answers can be used to improve the performance of the solution in the Analysis and Results by trying to get the performance of the solution to a 96% Accuracy. Initial user tests were also conducted in this section to make sure the solution operated correctly before a final conclusion was reached.

Professional and Ethical Considerations

This project involved adult participants who agreed to participate in a questionnaire by giving feedback on how they interact with Spam/Scam emails. Therefore, there are potentially serious professional and ethical issues to be considered, since the questioning may affect the participants and/or project members as well as the project itself.

BCS Code of Conduct

The contents of this project fulfil all requirements of the BCS Code of Conduct, derived from the Chartered Institute for IT:

Section 1; Public Interest

"You shall:

a. have due regard for public health, privacy, security and wellbeing of others and the environment.

b. have due regard for the legitimate rights of Third Parties.*

c. conduct your professional activities without discrimination on the grounds of sex, sexual orientation, marital status, nationality, colour, race, ethnic origin, religion, age or disability, or of any other condition or requirement

d. promote equal access to the benefits of IT and seek to promote the inclusion of all sectors in society wherever opportunities arise [27].*"*

In respect to the above, the project has regards for the project participants' wellbeing (1a), as participants were required to sign a consent form and confirm that they were over the age of 18. All data retrieved from the project was done so anonymously ensuring privacy, security and wellbeing of anyone who chose to participate (1b). In order to prevent discrimination, the questionnaire was shared openly using Google Forms so anyone who wishes to participate may do so after consenting to the previous(1c). To align with the Code of Conduct (1d), all data gained was used to create a tool that is free to use and promote the benefits of IT to all.

Part 2; Professional Competence & Integrity

"You shall:

a. only undertake to do work or provide a service that is within your professional competence.

b. NOT claim any level of competence that you do not possess.

c. develop your professional knowledge, skills and competence on a continuing basis, maintaining awareness of technological developments, procedures, and standards that are relevant to your field.

d. ensure that you have the knowledge and understanding of Legislation and that you comply with such Legislation, in carrying out your professional responsibilities.*

e. respect and value alternative viewpoints and, seek, accept and offer honest criticisms of work.

f. avoid injuring others, their property, reputation, or employment by false or malicious or negligent action or inaction.

g. reject and will not make any offer of bribery or unethical inducement [27]."

The project was discussed multiple times with my supervisor and was believed to be within my professional competence and I will not break any legislations, e.g. GDPR, within the project's lifetime (2a, 2b, 2c, 2d). Time was also blocked out for meetings on a weekly basis to share progress of the project and allow for feedback and criticism (2e). This was to ensure that the completed project was a functioning tool that met all aims and objectives that were set by myself or the supervisor (2f). I was also not tempted by bribery or any unethical inducement and I did not seek to encourage this behaviour within the project (2g).

Part 3; Relevant Authority

"You shall:

a. carry out your professional responsibilities with due care and diligence in accordance with the Relevant Authority's requirements whilst exercising your professional judgement at all times.

b. seek to avoid any situation that may give rise to a conflict of interest between you and your Relevant Authority.

c. accept professional responsibility for your work and for the work of colleagues who are defined in a given context as working under your supervision.

d. NOT disclose or authorise to be disclosed or use for personal gain or to benefit a third party, confidential information except with the permission of your Relevant Authority, or as required by Legislation.

e. NOT misrepresent or withhold information on the performance of products, systems or services (unless lawfully bound by a duty of confidentiality not to disclose such information) or take advantage of the lack of relevant knowledge or inexperience of others [27]."

The relevant authority for the overview of this project was the School of Informatics located at the University of Sussex. Therefore, I abided by the University's requirements by producing all specified criteria by their deadlines (3a). I also ensured that I avoided any conflicts of interest between myself and the University of Sussex throughout the duration of the project by completing the aims and objectives of both the project and the University (3b). Therefore, I accept all responsibility for the project and tools' successes as it was created through my professional competence (3c). No confidential information was disclosed me refusing to comment when asked and not discussing any confidential information other than with my supervisor. This includes any data that was obtained during the questioning phase of the project which was made anonymous to give security and privacy to the participants (3d). Any information about the project was not withheld or misrepresented in anyway by freely giving up the progress of the project (3e).

Part 4; Duty to the Profession

"You shall:

a. accept your personal duty to uphold the reputation of the profession and not take any action which could bring the profession into disrepute.

b. seek to improve professional standards through participation in their development, use and enforcement.

c. uphold the reputation and good standing of BCS, the Chartered Institute for IT.

d. act with integrity and respect in your professional relationships with all members of BCS and with members of other professions with whom you work in a professional capacity. [27]"

I agree, to the best of my ability, that this project was completed in a professional manner and made sure it adhered to the BCS Code of Conduct (4a, 4b, 4c). This professional manner was extended to the members of the BCS to ensure integrity and respect (4d).

Ethical Issues

This report did not require ethical approval as it met with all 12 points of the "Ethical Compliance Form for UG and PGT Projects" and therefore only required myself and my Supervisor to sign the Ethical Compliance Form which has been included in the Appendix. Consequently, I was able to confirm that the project was to be completed with complete regard for health and safety as well as professional and ethical regulations, by making sure that all participants were over the age of 18 and that they have explicitly consented to being a part of the project. In addition, they must have confirmed to not having a disability that could have limited their understanding, communication, or ability to be able to consent to taking part. In order not to corrupt the project and data collected, no incentives were given to the participants and neither myself nor my technical supervisor were placed in a position of authority over the participants.

Prior to taking part in the project the participants were also fully informed about all aspects of the project without intentionally leaving out any information on the goals or conclusion of the project and how their information shall be used. Therefore, they were able to withdraw at any time as well as being given contact details for both myself and my supervisor so that questions can be asked at any time. All data was stored anonymously using password-protected Google Forms and the participants were informed if how their data was to be stored.

Related work

This project's goal was to create a tool to discern whether a received email is Spam, Scam or Neither (Ham) and update the user in an appropriate manner. In creating the aims and goals for this project, an idea of how it may function already existed which can be seen in the original proposal attached in the Appendix. However, to further develop this Methodology, research was conducted to understand how similar projects had been completed in the past. These ideas were then adapted and improved for use in this project or as evidence that a certain methodology would not be appropriate for the given task. Fortunately, creating this program was not a completely new idea and there were plenty of papers available for research purposes.

Detection Methods

Throughout the twenty papers that were read for research purposes, there were many proposed methods for detecting fraudulent emails which can easily be divided up into separate categories: such as detection methods for Scam and others for Spam emails. Scam is also referred to as phishing emails, which are defined as emails that actively try to steal information or money from the user, whereas Spam emails are simply unsolicited junk mail [28]. Each category requires different methods for detection, meaning it is only possible to determine if an email is Spam or Scam through differing techniques when processing the email.

For example, as concluded by multiple papers, phishing emails are usually quite complex in their nature, employing certain phrases and JavaScript to trick the end user out of their information and/or money. Andronicus et al.'s study 'Classification of phishing emails using the Random Forest machine learning technique' [4], raised multiple ways to classify an email. However, they managed to condense these down to 15 main features of phishing emails ranging from including different HTML links to IP addresses in HTML links or specific key words and phrases. These are all features that were to be explored when structuring the tool to classify a given email's type with the best Accuracy possible. The aforementioned inclusion of JavaScript inside of phishing emails can be attributed to Fette et al.'s study into learning to detect phishing emails [7] in which it is discussed that it is impossible to operate a scam without some JavaScript or HTML links. These ideas were replicated in both Zhang and Yuan's 'Phishing detection using neural network' paper [8] and Basnet et al.'s 'Detection of phishing attacks: a machine learning approach' [10]. These papers discuss that having multiple web domains in an email increased the chance of it being a phishing email, this makes sense as non-fraudulent emails would be for a specific company and therefore would only include a singular domain. However, there may be cases where this is not the case, such as email chains or research, but this would be taken as one of many features. In Basnet et al.'s paper[10], they also discuss that there are six main word groups that words and phrases fall into that are used by phishing emails. Therefore, by measuring the number of these, such as large digits, signalling money, or hyperlink text in website links or even an address you could classify an email. Although this proposed method could have functioned, it was less advanced than previously mentioned methods such as Andronicus et al's study [4]. Therefore, the best method for detecting was deemed to be by collecting a set of features such as different words, phrases and inclusions and seeing how many appear in an email. A classifier can then be trained to recognise how a signature of these features relates to Scam, Spam or Ham emails and therefore by looking at the frequency of these features in an email, the program decides whether to classify the email as Scam, Spam or Ham.

Classifiers

Throughout all the papers read whilst researching, multiple classifiers were tried and tested with the purpose of determining how an email should be classified, with all displaying varying levels of success. For example, all that were able to function correctly can be described as highly accurate as the lowest Accuracy of these was only 95%. This can, however, be dependent on the balance of classes, Scam, Spam or Ham, in the test and training samples. The lowest Accuracy of these papers was recorded in Pattewar & Rathod's "A Comparative Performance Evaluation of Content Based Spam and Malicious URL Detection" [25] achieving a 95% Accuracy. This used the same classifier as in Rathod and Pattewar's paper 'Content based spam detection in email using Bayesian classifier' [23] which achieved a 96% Accuracy. In this study, they applied a Naïve Bayes classifier to the problem to attempt to find Spam emails; this differs to looking for Scam emails but is similar in function and shows that using only a Bayesian classifier may not be suitable for the project. It should, however, be noted that Microsoft Outlook uses a Naïve Bayes classifier called SpamBayes for its Spam email detection [15]. This therefore resulted in a target goal to have the resulting Accuracy of the program's classifier to be above 96%, in order to be better at classifying scam emails than Outlook which has a large user base with over 400 million users in 2018[32]. Another paper that achieved 95% Accuracy with their classifier was Ozarkar and Patwardhan's paper 'Efficient Spam classification by Appropriate Feature Selection' [26] in which a Partial Decision Tree was used. This demonstrates that there are other methods that can be used to get a high Accuracy.

There were also multiple examples of papers that have achieved over 96% Accuracy with their classifiers, most of which used a combination of different classifiers to achieve this. For example, Toolan and Carthy's paper 'Phishing detection using classifier ensembles' [21], gave a 97% Accuracy using a mixture of C5.0, Naive Bayes, SVM, Linear Regression and K-Nearest Neighbours. It should be noted that any mix of classifiers in a paper which included a C5.0 algorithm Decision Tree gave the best results and therefore a C5.0 classifier is a prime candidate to be added to a mixed classifier. However, this 97% Accuracy was also achieved when applying a SVM (Support Vector Machine) with nine layers and looking at nine specific features. However, results of the SVM classifier could have been biased as it operated with a small dataset of only 1000 emails which could explain the high Accuracy result. This was shown in Form et al.'s paper 'Phishing email detection technique by using Hybrid Features', [21] in which they split the dataset perfectly to have an even number of normal and scam emails. This therefore raises a point on how the final data set is to be constructed as it must be well balanced, but large enough in size to train and test the program effectively. As for the chosen classifier, the highest scoring in Accuracy was the Random Forest classifier scoring accuracies as high as 99.7% on multiple occasions. One of these being in the paper, 'An intelligent classification model for phishing email detection' by Yasin and Abuhasan [6], where they used TF-IDF values and other methods to detect features that can then be used to train the Random Forest classifier. This is a similar method to the paper 'Detection of fraudulent emails by employing advanced feature abundance' by Nizamani et al. [4] although they used a SVM as their classifier thereby giving them lower Accuracy. Therefore, it is evident that Random Forest, SVM, C5.0 and Naïve Bayes should be tried in different configurations to get the best classifier possible.

Design and Implementation

Design and Implementation details the requirements of each module of the program as well as how each module will be implemented into the program. A diagram also shows how the overall program operates.

Requirements Specification

The following is a detailed breakdown on how each of the elements in the program operates to reach a point of minimum functionality along with extra tasks that were undertaken to advance the programs complexity further.

Email Module

Module responsible for the retrieval and sending of the email for the program

- **Ability to check an inbox**
  The program can access and retrieve emails from an assigned inbox
- **Ability to send an email to recipient**
  The program can take a recipient's email and reply to them
- **Contain a useful tutorial for how to mark emails as spam**
  The reply email sent to the user when an email is confirmed to be spam or scam, also contains instructions as to how to set up an inbox's spam filter for an address

Classification Module

Module in control of classifying whether an email is Spam, Scam or Ham

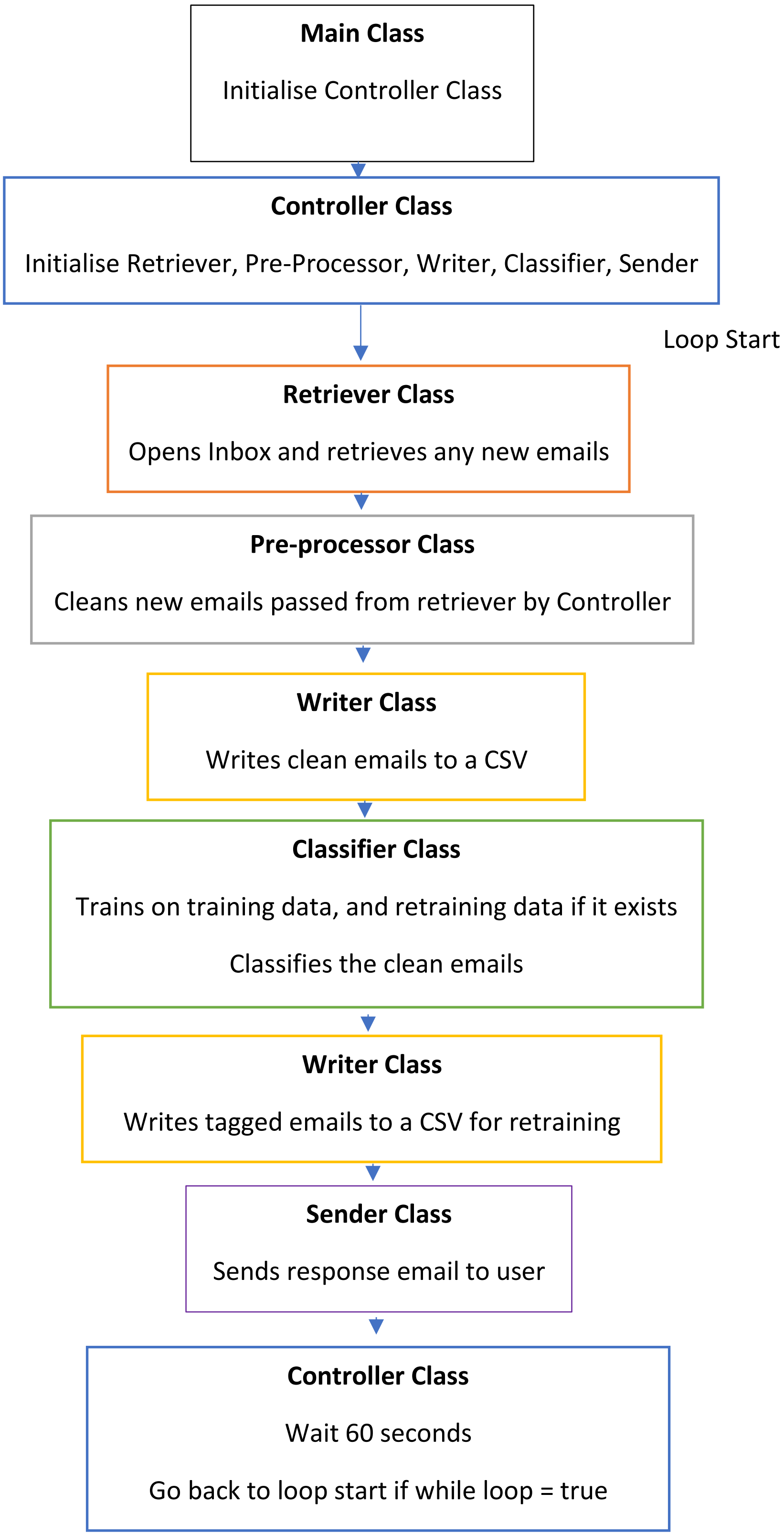
- **Ability to calculate keywords**
  Works out keywords prior to training
- **Ability to train a model based on training data**
  Allow the algorithm to train a model from training data of a data set
- **Ability to classify an email as Scam, Spam or Ham**
  Know what features relate to features of the training data and classify accordingly
- **Ability to gather features**
  Look through given email and gather features via pre-processing

Extensions

The following tasks were to be researched once the above became functional

- **(Researched) Ability to write emails to scammers.**
  Program can reply to the original scammer in a convincing way to waste their time
- **(Implemented) Ability to add emails to the data set**
  Take emails that have been identified as scam, spam or neither and add them to the data set
- **(Implemented)Ability to retrain itself off data set**
  Take a refreshed dataset and use it to retrain the classifier model
- **(Removed) Ability to modify dataset**
  Alter or combine known emails to create new data to train and test itself

Program Overview Diagram


Main Class
Initialise Controller Class
Controller Class
Initialise Retriever, Pre-Processor, Writer, Classifier, Sender
Loop Start
Retriever Class
Opens Inbox and retrieves any new emails
Pre-processor Class
Cleans new emails passed from retriever by Controller
Writer Class
Writes clean emails to a CSV
Classifier Class
Trains on training data, and retraining data if it exists
Classifies the clean emails
Writer Class
Writes tagged emails to a CSV for retraining
Sender Class
Sends response email to user
Controller Class
Wait 60 seconds
Go back to loop start if while loop = true

Tasks

Primary Tasks

Pre-Production

Before beginning construction of the program, a questionnaire was required to gather information as to how people interact with both Spam and Scam emails. The goal of this was to use the information gathered to determine what type of features should be used to train the classifiers and the best methods to classify emails as Spam, Scam or Ham; such as sums of money or exotic destinations. Therefore, the questionnaire included enquiries as to what people looked for in emails to determine their class, what key words they would associate with these and some demographic information. This demographic information allowed the project to be better tailored to all age ranges, even those with a lower technical expertise, thereby allowing the project to be able to adhere to its main aim of helping vulnerable people to be less successfully targeted. For example, this gave the ability to give less technical adept users similar recognition capabilities of someone within the industry or someone with more experience. However, from research I discovered more prevalent features for classification than those given and was therefore able to use these as well. Whether these found features proved to be more successful or not was a hypothesis to be tested in the Analysis section by having the program consider questionnaire gathered features as well as researched ones.

Retriever Class

The way the program initially operates is with a user sending an email to the isThisScamOrSpam@gmail.com email address to query if the email is Scam, Spam or Ham. This is the inbox where the Retriever class of the program retrieves all new emails and adds them to a CSV to be processed by the rest of the program. These emails can then be read using Natural Language Processing (NLP) techniques to determine the email's feature set, with these features mainly being the frequency of words, HTML links and the email address of the sender. However, there is one issue with using Gmail as the chosen inbox for the program as Gmail already has a Spam filter enabled so some of the emails get moved to a Junk section of the inbox. Although in its current iteration this can be mostly disabled, a future update to the program could have it check the junk mailbox as well as the main inbox.

Pre-processing

Before passing features to the Classifier class, the email must firstly be pre-processed which occurs in the Pre-Processor class of the program. Pre-processing begins by firstly collecting and then removing links from the email body and storing them separately before tokenising the email. Tokenising occurs by splitting strings of words and phrases into smaller strings, in this algorithm this would be splitting sentences into words [29]. Tokenisation occurs with both the subject and body of the email to allow it to be further processed by the pre-processing algorithm with the next process being stop word removal; stop words in NLP being words such as 'the' or 'and' as they are deemed irrelevant for the processing that is taking place. In this program, counting the number of times 'the' occurs is irrelevant as both Scam and Spam emails are likely to use 'the'. Therefore, nothing will be added to the classification capabilities of the classifier by including 'the', consequently 'the' and other irrelevant words are removed [30]. Following this, tokenisation occurs again as stop word removal can leave erroneous characters that need to be separated from other words.

Subsequently, the body and subject can be passed to the remove symbols function which removes erroneous characters and symbols that are irrelevant to the classification of an email such as characters such as ‘n’ or ‘/’ which were found to be quite frequent in the testing data. Next, the cleaned text is passed to the lemmatize function where words are grouped together by converting them to a variant of the same word. An example of this would be lots of words that mean ‘good’ such as ‘better’ being converted to the common word of ’good’. Following this pre-processing, the frequencies of the remaining key words can be calculated for both the body and the subject as well as any numerical value being converted to the “Number” string to be better be counted. These are then stored in a dictionary format with the word followed by their frequency in the email. This dictionary is then passed to the ‘get nouns’ function which sorts the dictionary by removing any words that are not nouns, adjectives or verbs as these are seen to be the most important features. These important features are then sent to the GetWordVector function where they are converted into a word vector so they can be read by the Random Forest and C5.0 classifiers by using a word index made from pre-processing the training datasets. These collected features are then stored in an array to be passed to the classifier with the word vectors.

Classifier Class

Once the features have been gathered by the pre-processor and retriever classes, they are passed to the previously trained classifier algorithm which calculates whether the email is Scam, Spam or Neither. The classifiers were trained by running training datasets through the Data Set Cleaner class which is the same pre-processor class used by the emails, but with some minor changes to allow it to read the dataset’s formats correctly. Once these data sets are processed and of the same format, they are fed to the training function of the Classifier class when the program initially runs at the same time as when the Inbox is opened by the retriever class. Here each classifier is fed the appropriate training data in the format required with Naive Bayes and SVM using dictionaries and Random Forest and C5.0 using the Word Vectors.

Once appropriately trained, emails from users that have been pre-processed are fed to the classify function of the classifier class where the Subject and Body are independently classified. Here each classifier is fed emails and the probability of being each tag is determined for each and added together to get the overall probability. This now leaves two variables, one with probabilities for Spam and Ham and the other for Scam and Ham, which are then passed to the Calc_Probabs function which calculates the overall probabilities for each email. These are then used to determine the final tag given to be given to each email.

Sender Class

Subsequently, the emails and their assigned tags and probabilities are sent to the Sender class which sends an email back to the user with a response stating the classification of the original email which is done by simply iterating through a list of the emails. This reply email can also include a short tutorial on how to mark emails as spam and block the address if the email is deemed to be Scam or Spam. The reason behind this is to increase how user-friendly the program is, as all the user must know is how to write an email and they will receive easy to understand information in return. Therefore, if this is all they know this tutorial will help them further allowing all generations to be able to use this program and spam filters without any advanced computer or coding knowledge. This differs from other solutions which required more knowledge of computers than just sending an email, either downloading a program or coding parameters of the filter yourself.

Secondary Extension Tasks

Machine Learning Reply Function

A proposed extension was to research into the creation of a function to generate and reply to these emails with believable emails to waste scammers' time. This required research into how Scam and Spam emails operated as well as what features carried the largest significance. The latter was already completed by the Classifier class when looking at the processed datasets and can be determined by outputting important features, leaving the meaning and context of each word being left to be discerned. However, in the creation of the classifier this was soon seen to be out of the scope of this program although a plan for how such a function would operate was devised. This would be by using data sets of replies to scam emails to be fed to a machine learning function where appropriate responses would be learnt when compared to the features of the scam emails. The features would be easy to collect as the Dataset Cleaner can be reused on these emails to get their features. However, adding this functionality would require a refactor of how the current program operates and the building of a new machine learning function which is out of the scope for this project.

Retraining Function

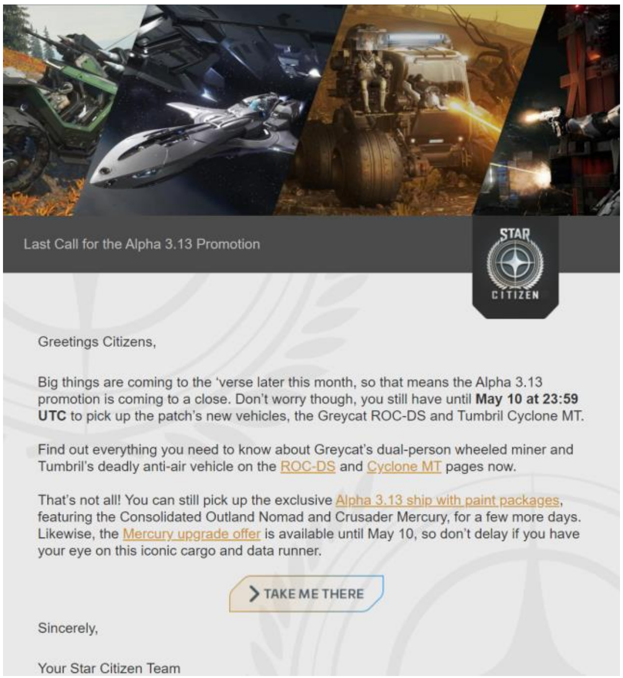
Two extension tasks that were pursued and are functional were allowing the program to collect tagged emails into a CSV file and then retrain itself from this file; where a CSV is a file format used for the data sets, leading to the creation of a Retraining feature. What had been previously planned was for a second inbox for people to send emails that they knew were Spam, Scam or Ham and tagged them as such in the subject, these would then be used to help in retraining the classifiers. The retraining function now functions by having emails tagged by the classifier added to the tagged data CSV by the Controller class for retraining. Then, on the next training pass for the Classifier class this data is added to the training data and the program retrains itself using this to improve Accuracy of the classifier. However, an issue with this is that over time there may develop a bias in the classifier as most emails sent to the program will likely be Scam or Spam with very little Ham. Therefore, some Ham emails may begin to incorrectly become Scam or Spam and may require some manual maintenance to keep the dataset balanced. However, a function was theorised which would work similarly to the Create More Data function by randomly shuffling the data set and splitting it evenly into Ham, Spam and Scam emails for training, but this was never added. For this reason, currently the Retraining function has to be activated by passing 1 for retraining_data to the Controller class in Main to add retraining data to the retraining CSV and run training again. Without long-term testing it is unknown how this may affect the Accuracy of the classifier as it may cause bias.

Create More Data Function

The final extension task was the ability to modify the emails in the dataset to create more data from the data already available. The goal of this would be to not simply duplicate but also to splice two Ham, Scam, or Spam emails together to create one new one. However, this was found to lead to repetition in the classifier and bias towards certain features of Scam or Spam emails and if retraining data is being added there was an argument that it may become redundant. For this reason, it was therefore removed from the Classifier Class where it was called by the Train function.

Implementation

The following are a selection of images showing the final implementation of the program. Image 1 shows a normal non spam email being sent to the program to be classified. This email is ham as it is part of a mailing list the user signed up to.



Last Call for the Alpha 3.13 Promotion

Greetings Citizens,

Big things are coming to the 'verse later this month, so that means the Alpha 3.13 promotion is coming to a close. Don't worry though, you still have until **May 10 at 23:59 UTC** to pick up the patch's new vehicles, the Greycat ROC-DS and Tumbril Cyclone MT.

Find out everything you need to know about Greycat's dual-person wheeled miner and Tumbril's deadly anti-air vehicle on the ROC-DS and Cyclone MT pages now.

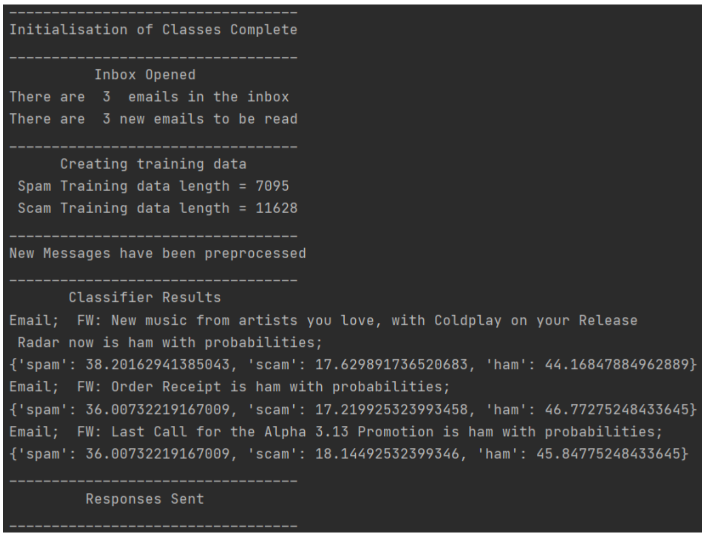
That's not all! You can still pick up the exclusive Alpha 3.13 ship with paint packages, featuring the Consolidated Outland Nomad and Crusader Mercury, for a few more days. Likewise, the Mercury upgrade offer is available until May 10, so don't delay if you have your eye on this iconic cargo and data runner.

Sincerely,

Your Star Citizen Team

Image 1: Email to be Sent to Program

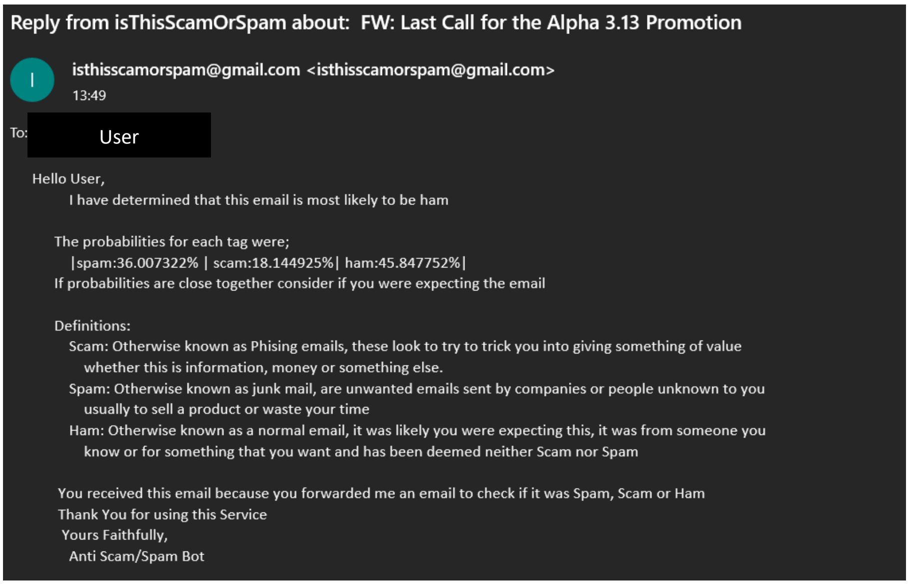
Image 2 shows the process undertaken by the program when initialising and classifying emails. The first thing that can be seen to be done is the initialisation of all classes, so they are ready to be utilised when needed. Following this the inbox is checked and values of the number of emails in the inbox and how many of these needed to be read are returned. Then the data sets are used to created training data with their lengths returned, if retraining data is added this shall also be shown here in more loops. After this the new emails for the inbox are pre-processed before being fed to the classifier, the results of which are then printed out in the classifier results. Following this the message of responses sent is displayed showing users have been sent a reply.



```
----------------------------------
Initialisation of Classes Complete

----------------------------------
          Inbox Opened
There are  3  emails in the inbox
There are  3 new emails to be read

----------------------------------
      Creating training data
 Spam Training data length = 7095
 Scam Training data length = 11628

----------------------------------
New Messages have been preprocessed

----------------------------------
       Classifier Results
Email;  FW: New music from artists you love, with Coldplay on your Release
 Radar now is ham with probabilities;
{'spam': 38.20162941385043, 'scam': 17.629891736520683, 'ham': 44.16847884962889}
Email;  FW: Order Receipt is ham with probabilities;
{'spam': 36.00732219167009, 'scam': 17.219925323993458, 'ham': 46.77275248433645}
Email;  FW: Last Call for the Alpha 3.13 Promotion is ham with probabilities;
{'spam': 36.00732219167009, 'scam': 18.14492532399346, 'ham': 45.84775248433645}

----------------------------------
         Responses Sent

----------------------------------
```

Image 2: Python Console

Images 3 and 4 display different outcomes for replies sent to the user with image 3 detailing the email sent if an email is found to be safe, ham, and image 4 if the email is found to be unsafe, either Scam or Spam. The difference between the two emails is the addition of a tutorial added to that found in the Spam and Scam email telling the user how set up a filter for that email. Both emails however contain the probabilities as well as a description of what each classification means.



Reply from isThisScamOrSpam about: FW: Last Call for the Alpha 3.13 Promotion

isthisscamorspam@gmail.com <isthisscamorspam@gmail.com>
13:49

To: User

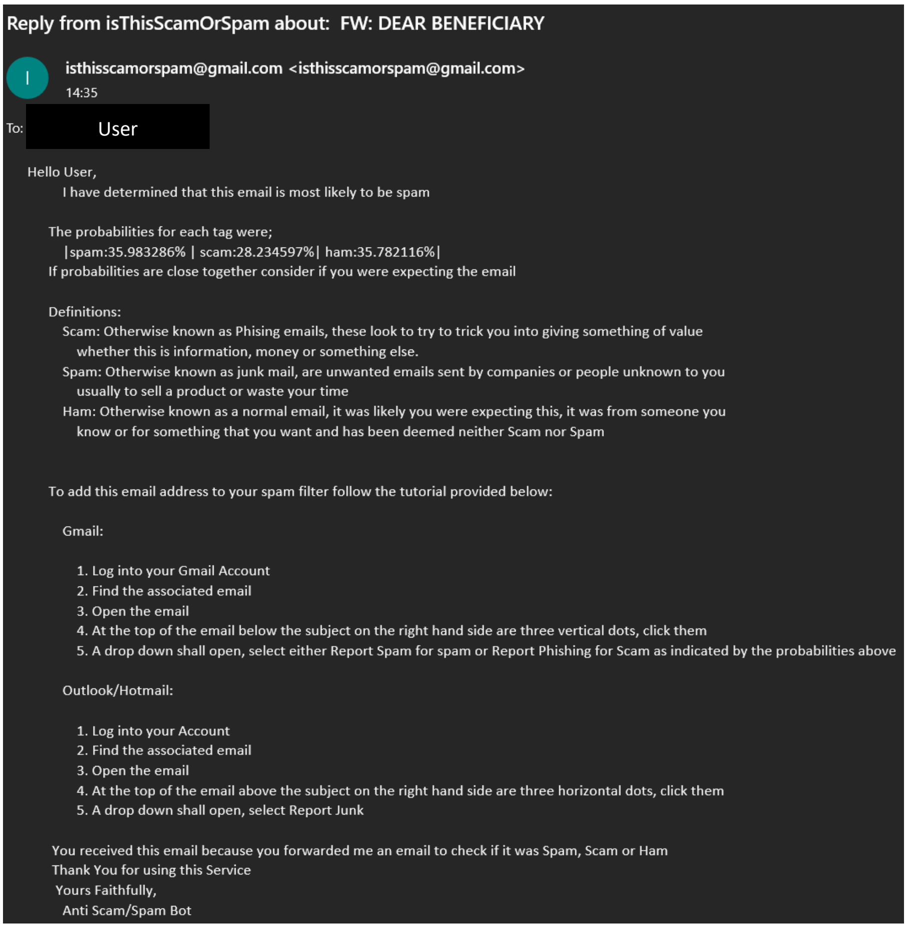
Hello User,
I have determined that this email is most likely to be ham

The probabilities for each tag were;
|spam:36.007322% | scam:18.144925%| ham:45.847752%|
If probabilities are close together consider if you were expecting the email

Definitions:
Scam: Otherwise known as Phising emails, these look to try to trick you into giving something of value whether this is information, money or something else.
Spam: Otherwise known as junk mail, are unwanted emails sent by companies or people unknown to you usually to sell a product or waste your time
Ham: Otherwise known as a normal email, it was likely you were expecting this, it was from someone you know or for something that you want and has been deemed neither Scam nor Spam

You received this email because you forwarded me an email to check if it was Spam, Scam or Ham
Thank You for using this Service
Yours Faithfully,
Anti Scam/Spam Bot

Image 3: Response email sent to user when email is ham



Reply from isThisScamOrSpam about: FW: DEAR BENEFICIARY

isthisscamorspam@gmail.com <isthisscamorspam@gmail.com>
14:35

To: User

Hello User,
I have determined that this email is most likely to be spam

The probabilities for each tag were;
|spam:35.983286% | scam:28.234597%| ham:35.782116%|
If probabilities are close together consider if you were expecting the email

Definitions:
Scam: Otherwise known as Phising emails, these look to try to trick you into giving something of value whether this is information, money or something else.
Spam: Otherwise known as junk mail, are unwanted emails sent by companies or people unknown to you usually to sell a product or waste your time
Ham: Otherwise known as a normal email, it was likely you were expecting this, it was from someone you know or for something that you want and has been deemed neither Scam nor Spam

To add this email address to your spam filter follow the tutorial provided below:

Gmail:

1. Log into your Gmail Account
2. Find the associated email
3. Open the email
4. At the top of the email below the subject on the right hand side are three vertical dots, click them
5. A drop down shall open, select either Report Spam for spam or Report Phishing for Scam as indicated by the probabilities above

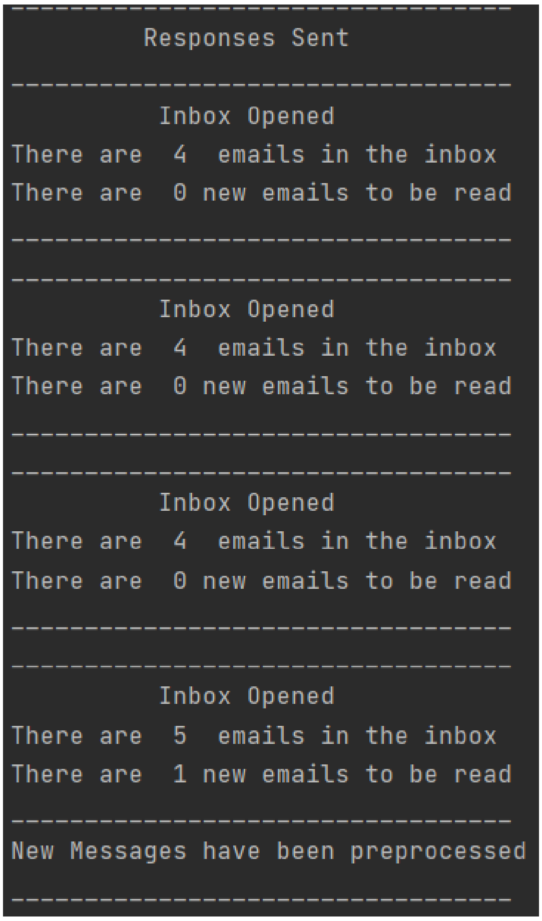
Outlook/Hotmail:

1. Log into your Account
2. Find the associated email
3. Open the email
4. At the top of the email above the subject on the right hand side are three horizontal dots, click them
5. A drop down shall open, select Report Junk

You received this email because you forwarded me an email to check if it was Spam, Scam or Ham
Thank You for using this Service
Yours Faithfully,
Anti Scam/Spam Bot

Image 4: Response email sent to user when email is spam or scam

Image 5 displays the python console checking the inbox every 60 seconds and when a new email is found performing the latter half of the processes found in image 2.



```
----------------------------------
         Responses Sent
----------------------------------
          Inbox Opened
There are  4  emails in the inbox
There are  0 new emails to be read
----------------------------------
----------------------------------
          Inbox Opened
There are  4  emails in the inbox
There are  0 new emails to be read
----------------------------------
----------------------------------
          Inbox Opened
There are  4  emails in the inbox
There are  0 new emails to be read
----------------------------------
----------------------------------
          Inbox Opened
There are  5  emails in the inbox
There are  1 new emails to be read
----------------------------------
New Messages have been preprocessed
----------------------------------
```

Image 5: Program checking inbox for new emails

Analysis and Results

This section displays all results of experiments conducted in the course of the project as well as the analysis used to gain useful insight into different areas of the project. These areas include the questionnaire that was conducted prior to the start of design, the testing and evaluation of the different classifiers and their combinations as well as an initial user to test to allow participants to trial the program.

Questionnaire

For the questionnaire the main goal was to be able to see how different demographics classify between Scam and Spam emails. Therefore, a questionnaire of nine questions was drafted ranging from questions on the participants age and gender to how they view an email. The Questionnaire was open to the public and hosted for two weeks on Google Forms where a total of 64 participants ranging from 18 to 80 participated. From these 64 participants, 576 different data points were gathered which have been analysed and used to help design how the program approaches classification.

Firstly, demographics of the participants were focused on with participants being classified as young when under the age of 41, this gave an almost 50:50 split between young and old participants. This is with 30 participants under the age of 41 and 34 over 41 resulting in a 47:53 split among young and old participants. This gave an interesting and non-biased comparison between how the young and old differentiate between Scam and Spam emails, as can be viewed in Figure 1. However, it should be noted the two largest categories for the survey were 18-21 and 51-60 resulting in a bias when discussing older and younger demographics as they are mostly in these age ranges. Similarly, there was only one participant for the 71-80 and 31-40 categories meaning that the demographics are not as large as previously thought. This means that the two groups for this survey should be viewed more as 18-30 and 41-70.

Age Range	Participants
18-21	20
22-30	9
31-40	1
41-50	9
51-60	16
61-70	8
71-80	1
81+	0

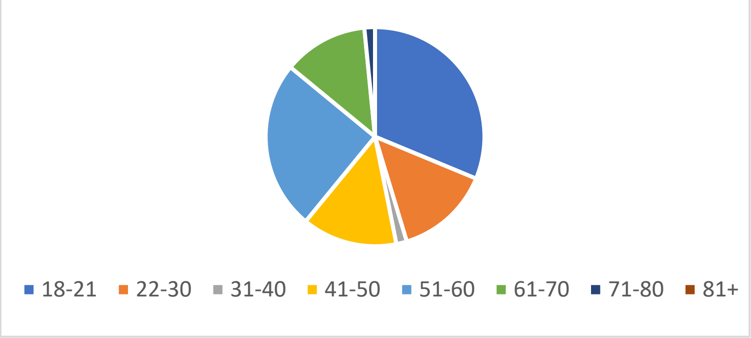


Figure 1: Age Range of Participants in Questionnaire

Equally, there was an approximate 50:50 split in Female and Male participants responding to the questionnaire. This is there were 32 female participants and 31 male participants, with one participant preferring not to say what their gender was (Figure 2).

Gender	Participants
Male	31
Female	32
Non-Binary	0
Prefer not to say	1

Figure 2: Gender of Participants

Gender was never going to impact the direction of the program, but combining age and gender allowed for more useful demographic information for the questionnaire. In the younger portion of the participants, there were slightly more males then females with a split of 56:44 which is the reverse of older participants where there were more female participants. In the older participants 61% of participants were female and only 39% of them were male (Figure 3). The most plausible reason for this was the survey was shared around Facebook which has a larger male population in the age ranges of 13-44, as found in a study from January 2021[31]. After this the female population is larger from 45-65+ which also fits with the questionnaire.

Age	Female	Male	Prefer Not To Say	Split	Split By Group
18-21	7	13	1	35:60:5	56:44
22-30	6	3	0	66:33:0	
31-40	0	1	0	0:100:0	
41-50	8	1	0	91:9:0	61:39
51-60	9	7	0	56:44:0	
61-70	4	4	0	50:50:0	
71-80	0	1	0	0:100:0	
81+	0	0	0	0:0:0	

Figure 3: Gender Compared with Age Range

With the demographic makeup of the participants known, it could be applied to future questions in order to calculate how key demographics classify emails. The next questions asked what participants pay attention to when self-classifying Spam or Scam emails, with choices of Subject, Address, Body or other being given. The aim of this was to understand what sections Spam and Scam classifiers should look at when classifying different types of emails. From this I was able to gather that the majority of participants, 64.1%, read the Subject Line when classifying Spam emails and 70.3% look at the address to classify emails as Scam. Therefore, the starting point for designing classifiers was looking at the Subject Line for Spam Classification and the address for Scam Classification. However, participants were able to pick multiple answers leading to the majority also saying they looked at email address for Spam, at 53%, and the main body for Scam, at 51%. Therefore, for each classifier type there was a possibility of combining these answers to make the best classifier possible.

When dividing these answers into age groups (figures 4 & 5), patterns on how emails are classified by different age groups emerge. For example, younger participants reported using the Subject line more for Spam identification, but as age increases this decreased whilst address and body use increases. This is true in all age categories except for 71-80 and 51-60 with the former only reporting Subject use, though with only a single participant in this group this was considered an outlier. The latter reported similar Subject use, but with the same address use as 61-70 and 41-50 whilst not using body. Therefore, for Spam email identification older generations rely more on the Address and Body whilst younger generations rely more on Subject. Therefore, by having the Spam classifier check both the Subject and the Address, the Spam classifier should work at its best. However, when researching datasets there were none found containing email addresses that could be digested by the algorithm and most did not include email addresses at all. Therefore, the Spam classifiers only works on the subject of a given email.

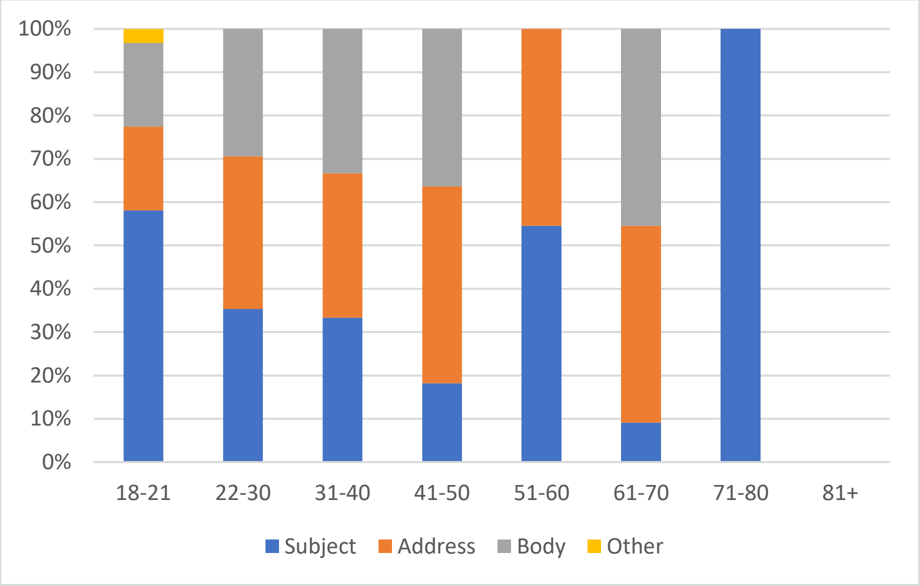


Figure 4: Spam Email Identifiers

The trend for Scam email identification is the reverse of that found in Spam email identification where younger participants reported using the Address more to identify which emails were Scam. As age increases Subject decreased slightly whilst Subject use increased except for in 51-60 and 71-80 where Address use became the majority again. Throughout the survey Body use is seen to fluctuate between low at 28% and 20% in the age ranges with the highest amounts of participants and rising to highs of 38% in groups with the least; meaning that in the majority of the ranges it was the second highest category. Consequently looking at the Address should have been the primary method for identifying Scam emails in the Scam classifier with the addition of body being secondary, however, this did not proceed as intended for the same reason as why Spam also does not use the Address.

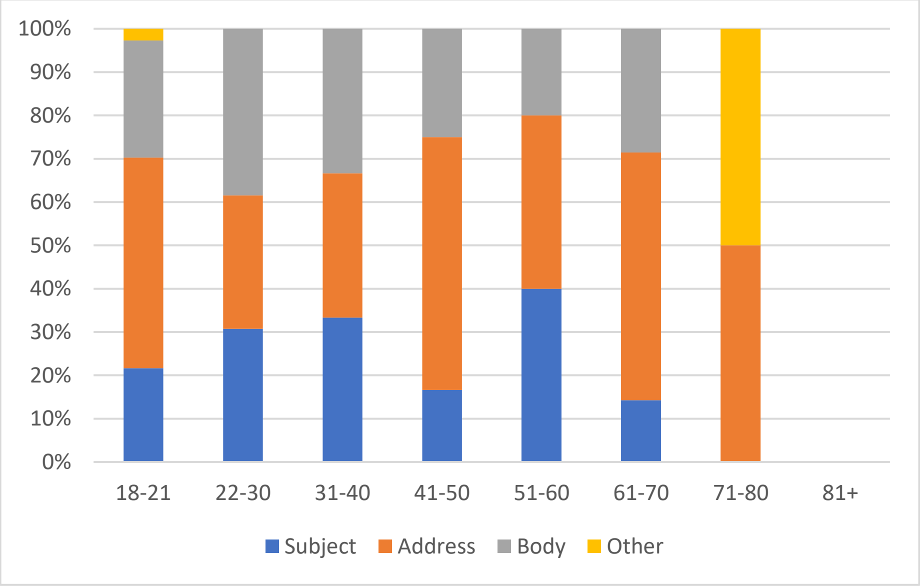


Figure 5: Scam Email Identifiers

It should however be noted that some participants filled out the other fields in both the lowest and highest age range. For Spam, non-alpha numeric characters were suggested as a potential identifier, however it proved difficult to find datasets including them as they are removed in pre-processing.

Another suggestion was the rule of "If something is too good to be true it is probably not" and, whilst true, classifiers cannot be built to recognise such a rule. One potential way this could have worked is having the program scan websites or databases to see if deals being offered were out of the possible ranges of what should be offered for a product. For the Scam classifier, participants expressed the use of the same rule as well as multiple participants recognising grammar and spelling errors as a sign of a scam email. The use of a spelling error checker is a feature that would have been easy to add to the program, however there would have been a difficulty in knowing the number of mistakes needed to classify something as Scam. This may have also increased false positive rates as Ham emails from friends and family may have also included spelling or grammar errors as they are more casual mail. To prevent this, it would have had to work in conjunction with other parts of the classifier and only be used if the chances of it being Scam or Spam were close already as a tie-breaker function.

The next two questions addressed the use of links in emails, as in research it appeared as a possible feature for Scam and Spam emails; with Scam hosting a single link to a fraudulent site and Spam emails containing many. Therefore, it was important to understand, when classifying emails, if the participants agreed with the question "Do links in unexpected emails make you cautious/suspicious?" being asked. In response, 97% of the participants answered 'yes', confirming the research(figure 6). The remaining 3% was divided between 'no' and "only if the sender's unknown", which again highlights the importance of the Address in identifying email types. For the one participant that answered 'no' they gave the reasoning of links themselves not being suspicious, however if they are in an already suspicious email then they would be suspicious of them. It should be noted that the lowest two age ranges were the only ones who gave responses other than 'yes'.

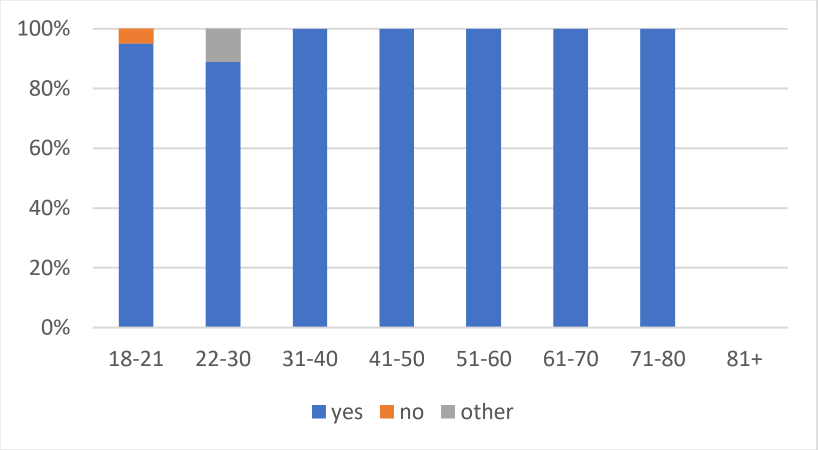


Figure 6: Are links suspicious?

The following question regarded the minimum number of links needed to be suspicious, in which 68% agreed that only one link is needed to be suspicious, whilst 12.7% agreeing on both 2 or 3 being suspicious. Finally, only 6.3% gave 4 links as a response for the minimum number needed to be suspicious. From these responses, it is possible to extrapolate that 68.3% would take one link as suspicious, 71% would take two, 83.7% would take three and 100% would take four or more as suspicious. When put against age ranges (figure 7), it is shown that the larger groups are the groups that have the participants requiring more links to be suspicious. The lower the size of the group the less links are needed to be seen as suspicious, because of this the number of links in an email are stored in the array given from the Pre-Processor they were just never utilised.

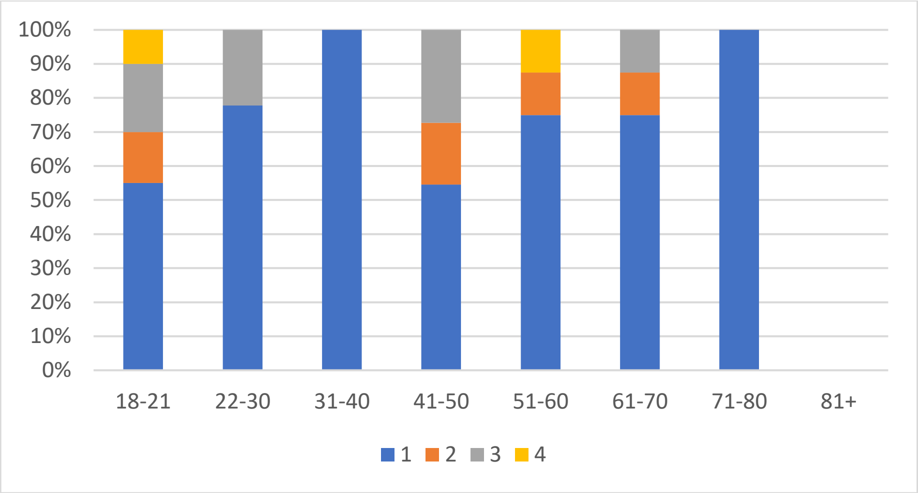


Figure 7: How Many Links in an Email Are Suspicious?

The final three questions of the questionnaire revolved around which words are suspicious if found in an email. The goal of this was to create a dataset that could be used by the classifiers to have the same performance as a human when classifying emails. This could be developed further by splitting the dataset into the different age ranges to see which age ranges have the best performance, although this was not done in testing.

In order to breakdown which words were given, they were run through an online tool that matches words to the most used synonym, similar to the lemmatize function of the pre-processor class. This therefore gives categories of words that were given which were then split by age group, as shown in Figure 8; it should be noted however there was some overlap e.g. money and payment. One category that was present throughout all age ranges was money and is seen to generally increase as you move through the age ranges meaning older generations worry more about money than younger generations. On the contrary, the younger generations have a large number of people using words found in the account category which is not seen in older age ranges.

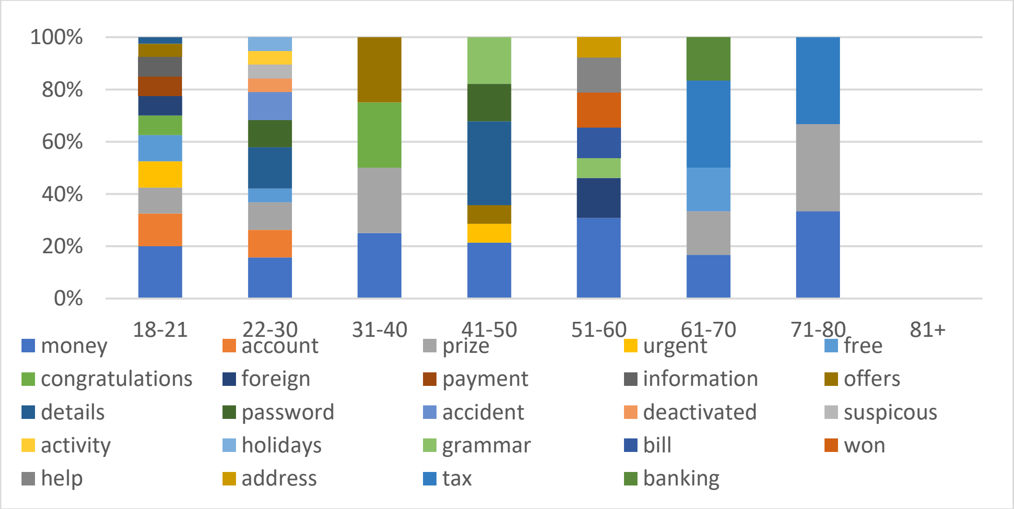


Figure 8: Key indicators for suspicious email

The remaining questions asked people to select the most and least suspicious words from the ones that they gave. From this the most suspicious words were 'money' and the phrase "you've won" which were chosen by 14% and 11% of participants. Following this, the remainder of the choices were equally split between 'bitcoin', 'free', 'HMRC', 'PayPal', 'update' and 'urgent' with either 3.5% or 4.6% of participants choosing them. Conversely, 18% of participants left no answer or replied that all given words were suspicious which were the largest groups of answers. The remaining participants were split evenly between things such as 'money', 'winning', 'security', 'offers', 'awarded' and 'free' with about 6% between each group. The interesting recognition from this is that age played no factor in these answers as there was no clear pattern between age ranges and answers given.

Classifier Accuracy

When setting the goal for this project it was important to consider that Accuracy was not the best statistic for measuring classifiers and it had to be used in conjunction with other measurements, these measurements being Precision, Recall and F-Score. However, when comparing this paper to other papers they mostly only used Accuracy therefore meaning it must still be considered when testing classifiers, although it has been used in conjunction with the others. This allows the possibility of evaluating whether existing Accuracies give enough information to rate classifiers or if more measurements are needed. This means that a good result would be just having a high F-Score whereas having a high Accuracy can still mean you have a low F-Score as F-Score is a combination of Precision and Recall, with Precision and Recall themselves being combinations of False Positive and similar rates. This means that the use of only Accuracy and F-Score would be plentiful when measuring the success of classifiers and would give more information about its performance than using just Accuracy. The full table of data for the comparisons between classifiers can be found in the appendix which details each of these measurements for all combinations of classifiers.

Experiments were conducted to see the importance of the data set, with vast Scam and vague Spam datasets, after research showed balancing of the data set was vital. In this case just Naïve Bayes classifiers produced a 79% Accuracy for the Scam classifier but only a 17.5% Accuracy for the Spam classifier which gave both classifiers a worse Accuracy than the Spam filter found in Outlook[15]. However, as shown, Accuracy is not the best measure and therefore by calling the getMeasures function of the classifier class the Recall, Precision and F-Score were calculated.

As seen in Figure 9, the Scam classifier had a high F-Score of 80% whilst the Spam classifier was low with 30%:for reference, an ideal F-Score would be 100% meaning a classifier has perfect Recall and Precision but here Spam was very low, but Scam was shown to be high. Closer look at the Precision and Recall of both classifiers shows that Spam has maintained a Precision of 100% but a Recall of just 18%. This means no False Positives were given, that is non-Spam emails identified as Spam, however the low Recall shown means that not all emails that should have been classified as Scam were classified as Scam with only 18% of them being classified correctly. This shows how unbalancing of the datasets gives errors due to training of the classifiers with Spam emails not being registered correctly as they were too vague in definition. As for the Scam classifier, the opposite was shown with an almost perfect Recall but low Precision, which indicates it was classifying everything as Spam to obtain a high Recall. This was also shown by the Precision where 31% of emails were classified as Scam when they were not. The importance of balanced training data can therefore be seen, which proves the earlier statement that datasets must be well balanced but large enough in size to train and test the program effectively as being true.

Measure	Scam Classifier %	Spam Classifier %
Accuracy	79.000	17.500
Recall	96.875	18.000
Precision	69.144	100.00
F-Score	80.694	30.508

Figure 9: Classifier's with unbalanced datasets

After refitting the dataset cleaner to give more balanced training data, as well as increasing the amount of data for Spam, results were seen to improve. This can be seen in Figure 10 where Spam Recall, Accuracy and F-Score all increase giving a performance much closer to that found in the Outlook Spam Classifier [15] which has an Accuracy of 96%. As for the Scam classifier, it stopped classifying everything as Scam which decreased its results slightly but allowed the program to correctly classify Scam and Ham emails given to it. This displays the importance of a balanced dataset when building classifiers.

However, with these results being lower than the 96% of the Outlook Classifier, my previous hypothesis that solo Bayesian Classifiers may not be suitable for the project was also proved true; although it should be noted that Outlook's classifier does use a Naïve Bayes classifier though they have access to a much larger dataset as they have access to every Inbox on their platform. Therefore, with a big enough dataset Naïve Bayes may become viable which may be a hypothesis that could be explored in a future project.

Measure	Scam Classifier %	Spam Classifier %
Accuracy	72.818	91.611
Recall	90.400	79.787
Precision	62.431	92.5925
F-Score	73.856	85.714

Figure 10: Solo Naïve Bayes Classifiers

A solo SVM classifier (Figure 11), proved to be more capable than solo Naïve Bayes classifiers for both Scam and Spam although improvements in Spam were only minor. Spam saw a 5% increase in Recall, a 6% decrease in Precision which resulted in the same F-Score for both Naïve and SVM Spam classifiers whilst the SVM's Accuracy decreased by 1%. Meanwhile the Scam classifier saw huge improvements in all categories except Recall where there was a decrease of 3% whilst the others increased in values ranging from 19% to 36% being the largest improvements yet. This makes a solo SVM Classifier an appropriate solution for the program.

Measure	Scam Classifier	Spam Classifier
Accuracy	93.624	90.939
Recall	87.200	85.263
Precision	98.198	86.170
F-Score	92.373	85.714

Figure 10: Solo SVM Classifiers

Experiments on the solo Random Forest classifier showed it to perform more poorly than the SVM and Naïve bayes classifiers with accuracies of 46% and 67% for the Scam and Spam classifiers (figure 11). These are at least 26% lower than those found in the SVM and Naïve Bayes classifiers. However, these are not the only scores to be lower than their counter parts as every other score is also lower meaning the Random Forest classifier by itself would not be a good use for the program's function.

Measure	Scam Classifier %	Spam Classifier %
Accuracy	46.488	66.555
Recall	60.769	11.494
Precision	42.021	66.666
F-Score	49.685	19.608

Figure 11: Solo RF Classifiers

Solo C.50 classifier experiments exhibited worse accuracies than the RF classifiers with both C5.0 classifier accuracies being approximately 2-3% worse than the RFs. Although the accuracies were worse, the C5.0 Scam classifier did manage a higher Recall than the RF classifiers by 4.5%, but this was still approximately 30% lower than the Naive Bayes and SVM Scam Classifier Recalls. As for Precision, these were also worse than the Rf classifiers with F-Score being about the same (figure 12). For these reasons a solo C5.0 Classifier would not be an appropriate solution.

Measure	Scam Classifier%	Spam Classifier%
Accuracy	44.482	63
Recall	65.3846	19.54
Precision	41.626	29.825
F-Score	50.595	23.611

Figure 12: Solo C5.0 Classifiers

Experiments on the advantages of combining classifiers were run with multiple combinations trialled, with their results being shown in Figure 13 and Figure 14. For the Scam classifiers the worst combination was found to be the RF and C5.0 combination producing the same scores as a solo C5.0 scam classifier other than the Accuracy which was an average of the two classifier's accuracies. On evaluating the different combinations, several patterns emerged: firstly, whenever the Naïve Bayes is added with another classifier the Recall is always above 90% ranging from 90.3% to 100% showing that its addition is key to obtaining a high Recall. Secondly adding the SVM and RF together gave the highest Precision, other than the SVM by itself. Thirdly whenever C5.0 was combined with anything all the scores were lowered, sometimes as much as 30% making it not a good fit as part of the solution. This may be in part due to the parameters used and with more refinement of the C5.0's parameters better results may be achievable. Using these three trends the best combination of classifiers for scam classification would be Naïve Bayes, SVM and RF though SVM and RF seems to perform better with a 20% higher Precision and 9% higher F-Score. Due to F-Score being a combination of both Precision and Recall it is the key measurement to look at for deciding the best combination to use. Seeing as the SVM and RF combination possess the highest scores in all categories but Accuracy, other than the solo SVM, it must therefore be the best combination to use.

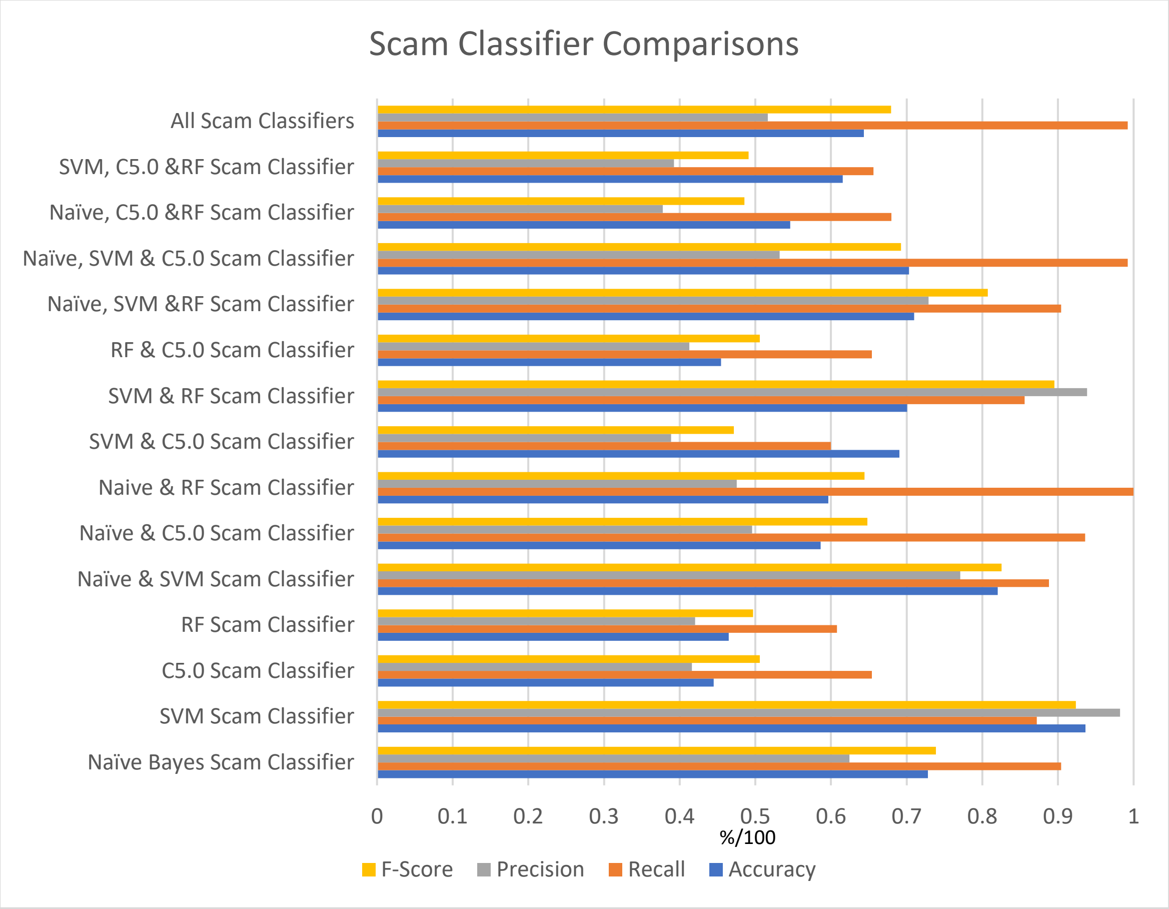


Figure 13: Scam Classifier Comparisons

When experiments were conducted on the combination of Spam classifiers, the worst combination found was RF and C5.0, the same as with the Scam classifiers. This is as RF and C5.0 combined gave the worst scores in all categories with an F-Score of 23%, being 9% lower than the next highest. This F-Score is only slightly higher than that of a solo C5.0 which was the worst performing solo classifier. On analysing the other combinations, the lowest results were always achieved whenever the C5.0 was combined with other classifiers, this can be seen with Naïve and RF. Here they achieve an F-Score of 80% but once combined with C5.0 this drops by 46% to an F-Score of 34%, this a trend that was also shown in the Scam classifier comparisons. Another trend that developed was that whatever combinations contained Naïve or SVM seemed to produce high scores. This is as a result of them having the highest scores solo and therefore when combining them with any other classifier the resulting score is only slightly worse than that of their solo scores. This is because the other classifiers perform badly solo whilst Naive and SVM do not. Therefore, the logical conclusion from this is that best combination for Spam classification should be Naïve Bayes and SVM classifiers combined with the data also backing this up. This is as combining the two gives the highest results with an F-Score of 87% which is about 2% higher than their F-Scores solo. Therefore, having these two classifiers combined is the best solution for classifying Spam Emails.

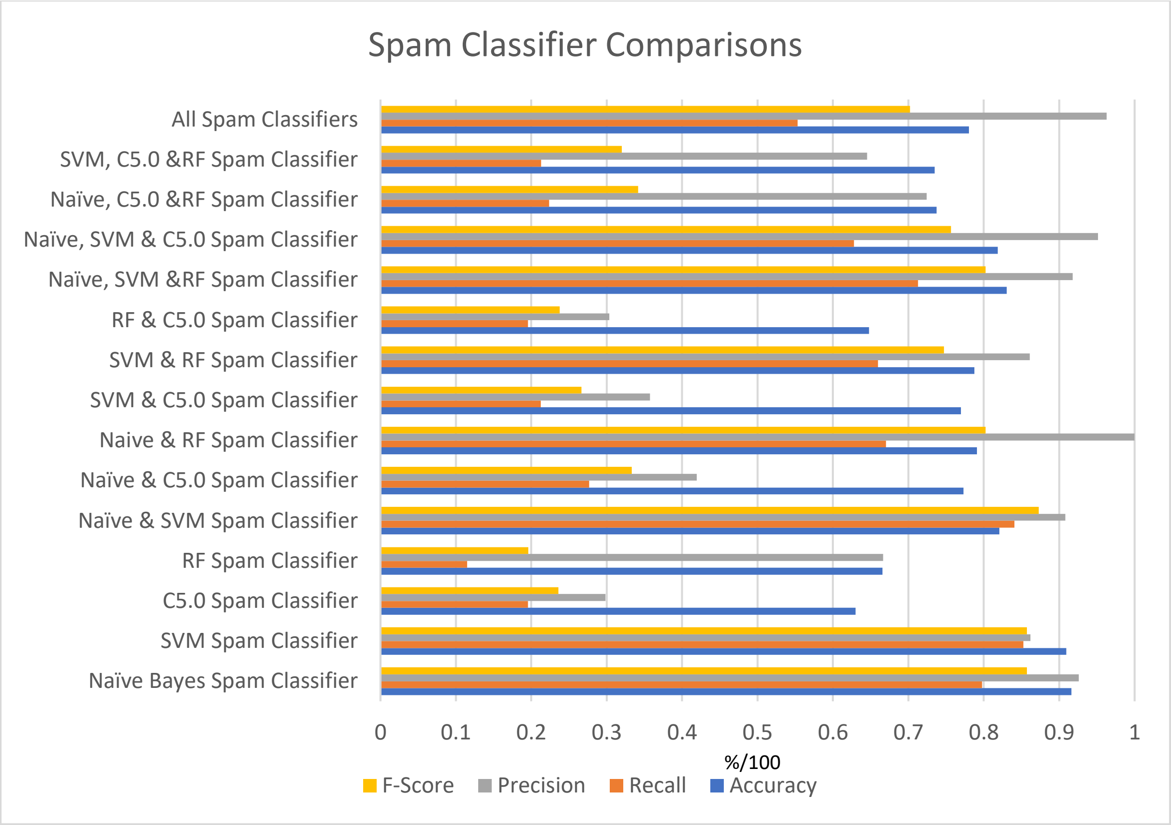


Figure 14: Scam Classifier Comparisons

On comparing the results of the Scam classifiers to the Spam it should be noted that Naïve Bayes performed better with Spam than Scam whilst for all other classifiers the opposite was true. For example, the Accuracy and F-Score of the SVM in Scam was 93% and 92% compared to 91% and 86%, whereas Naïve had 72% and 74% in Scam but 92% and 86% in Spam. This demonstrates that certain classifiers work best for different types of emails. However, it should also be noted even in combining classifiers, none of the classifiers ended up performing better than Outlook's [15]. Also, if combining classifiers was not an option SVM would be the best for a solo classifier as it has the highest scores and is used in each of the final combinations.

Questionnaire Dataset

One of the questions that came up in the creation of this project is whether a dataset of features participants thought were important could be used to better classify emails than the created dataset. Therefore, by using a Naïve bayes Scam and Spam classifier this can be determined by training it once on the original dataset, once with the Questionnaire dataset and again with both (Figure 15). The difference in performance between just the dataset and both was found to be very small with the Scam classifier getting a 3% higher F-Score with both but the Spam classifier being 7% worse when both datasets are used. This may be due to the features being given by the participants of the questionnaire being more focused on Scam email features than Spam email features. When viewing the Questionnaire data by itself though, this hypothesis is proved to be false as the Spam classifier's F-Score is seen to be 16% higher than the Scam classifier with the Spam classifier achieving a Precision of 100% and a high Recall. However, on closer inspection this is likely due to similar reasons shown in the dataset experiments where the data was too precise. This is as there is no Ham data in the questionnaire data set only features which are used for both Scam and Spam classification. Therefore, the classifiers would take anything that does not include one of these features to be 100% Ham whereas some with a few features are more likely to be Spam or Scam. In conclusion adding the Questionnaire dataset to the Original dataset should be done with the Spam classifier but not the Scam answering the earlier hypothesis.

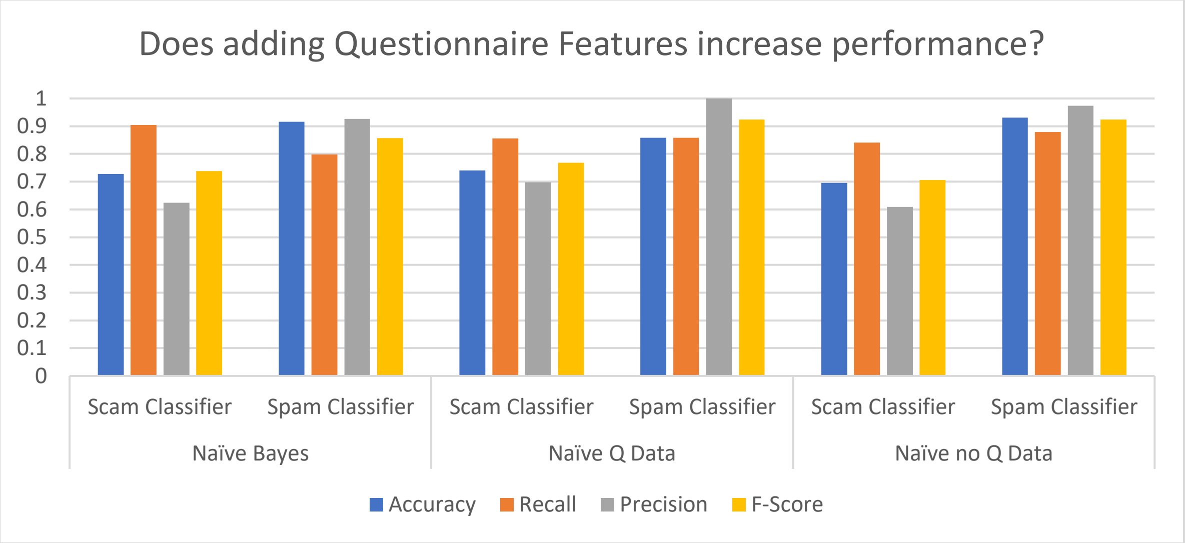


Figure 15: Does adding Questionnaire Features increase performance?

Initial User Study

An initial user study was conducted where users who asked to be kept up to date with the progress of the project were contacted and asked to participate. Participating involved sending emails to the programs inbox and then evaluating the response received from the program on aspects such as useability and Accuracy. There was also an opportunity for the initial users to give constructive criticism and feedback on how the program functions.

Of the 20 people contacted only 10 of them participated in the trial, but the participants' ages ranged from 18 to 70 giving a large range to test the usability of the software. All of the 10 participants gave the usability a score of between 8 to 10 when asked to score the usability on a scale of 1 to 10, with 80% giving the usability a 10. The other 20% gave slightly lower scores but did not comment as to why. In terms of the Accuracy of the results 80% of participants gave a score of 9 or 10 with the other 20% giving scores of 3 or 6 for varying reasons. The lower score of 3 was given as the classifiers classified an email in Spanish as Scam due to it being trained that foreign words were a key feature of Scam words. This may be an area to improve on in the future by adding in training data for specific languages other than just English. The other low results were not given a comment as to why they were marked so low. Finally, the usability of the instructions given in the return email, on how to set up a Spam filter for your inbox, was scored by the participants. For this all participants scored them between 7 and 10 with 80% of these being a 10 allowing for this to be considered a successful tutorial.

As for the additional feedback, points were made by the participants to include definitions on what Scam, Spam and Ham means in the results to allow the user to better understand what is going on. This was viewed as a valid idea and was implemented after the test to increase the email's usability and help educate users. Another comment asked to make it more discernible what email the program was talking about especially when multiple emails had been sent. Therefore, the original subject is added to the subject of the return email saying that the return email is replying to it to increase coherence. All other comments were simple messages congratulating the program on performing so well and being surprisingly useful.

Therefore, in evaluating the program on its main goal of helping vulnerable people to be less successfully targeted along with any other users of the service as well as its usability these can both be considered a success from the information shown here.

Conclusion and Evaluation

Final thoughts on the project are that it was largely a success with 91% of objectives implemented with all being researched and planned with one having to be removed for causing feature bias. All email module functions were completed with emails being retrieved from the isThisScamOrSpam@gmail.com inbox and replies being sent to the correct recipient. A useful tutorial was also placed in the reply email showing how to add an address to a spam filter. As for the classification module, keywords are calculated easily and used to generate features of Spam, Scam and Ham emails. Multiple classifiers are then trained using this data to effectively allow them to appropriately classify Spam, Scam and Ham emails correctly. As for extension tasks, the program can add retraining data to a CSV whilst retraining itself off this data and was able to modify the training dataset to create new data; however, the latter caused bias issues and was consequently removed. The last extension task was replying to Scam emails to waste the scammers time and although this was planned it was never implemented due to it being deemed outside of the scope of the project.

Final observations on the performance of the Classifier from the experiments are that even though the hypothesis that Naive Bayes classifiers would not be a good fit for the solution proved true, it was still used in the final combination for the Spam classifier. Although this was due to the combination of Naïve and SVM giving the highest results, these were still not close to the ones found in research, which were between 95% and 99%, with the combination only getting an Accuracy of 82%. This may be in part to dataset size though, as the most classifiers in the research had access to much larger datasets and may be something that the retrain function may improve. Another observation was that C5.0 was not found to be as effective as was alluded to in the research where C5.0 was shown to be in the highest performing classifiers. The reason for its ineffectiveness in these experiments may be due to its parameters not being correctly tuned, this is something that could be experimented with in the future. Similarly, this is also the same reasoning behind why the RF may not have performed so well either though it did perform better than C5.0.

In the user tests, most emails are either Spam and Ham with Spam and Ham being very close in the final probabilities with a tendency to more likely be Spam than Ham with Scam only being called when a Scam email is sent. However, the probabilities are given to the user with a message warning them of this, therefore making it possible that users can use this information to decide for themselves what the email is.

Any future work on the project would revolve around the addition of the planned automatic reply function that would reply to scammers to waste their time. This feature was already designed and would be able to be implemented but would require the refactor of the sender and pre-processor classes to allow for appropriate responses to be sent. A machine learning class would also have to be implemented to learn what these correct responses are from new Data Sets, which may have to be generated by hand, once they have been pre-processed.

Another function that would be re-added in the future was the shuffling data function which spliced emails of the same email type together to make more training data for the Classifiers. If this were to be re-added, it would need to be refactored so certain features did not become biased and more prominent thereby lowering the Accuracy and F-Score of the classifiers. A potential way to solve this would be to maintain the current ratios of features by not allowing two emails to be spliced if it would upset this ratio by a large degree.

From comparing my Naïve Bayes classifier to the likes of Outlook's it was possible to see that the reason behind theirs having such a higher Accuracy is due to them having a much larger dataset from their access to all Spam and Scam emails in their customers inboxes. Therefore, in the future a possible study could be conducted to see if increasing data increases the Accuracy of a Naïve Bayes classifier and by how much. This would involve a further study into how Outlook's classifier operates.

If this project was to be repeated from scratch it would be done with more up-to-date datasets or by collecting appropriate data myself to better, ensure its validity and that it is up-to-date with the current Scam and Spam methods. This would allow the classifiers to be more accurate as they would not be using outdated training data. These would then be fed to classifiers that use the same input for training as different inputs led to varying difficulties when building the classifier class. Therefore, if they all used the same input the process of training and classifying data would be easier.

Another feature which would allow for a better running program would be the use of TF-IDF values to register features instead of the current frequency method. This is by comparing the frequency of a term or word across all documents instead of just a single document which is how the program currently functions. This would give a bigger scope and allow the more important features to be more easily recognised rather than just comparing important features of one document with another as it currently operates. However, it is unknown whether this would improve values though it is highly likely. The use of the frequency and contents of links should also be added as this was a planned feature which was never fully developed in the implementation phase, despite them being collected.

Following this report, the program shall be run on a Raspberry Pi to run long-term and in order to complete the main goal when starting out this project. This being to help vulnerable people to be less successfully targeted along with any other users of the service and by running this program 24/7 on a raspberry PI this goal can be successfully achieved.

In conclusion, this project has proven to be a success with a useful product that performs well even if not at the expected levels found in research. Although all desired features were not implemented, these and other improvements leave room for this project to be further developed in the future or for it to be successfully re-implemented. Although even without these improvements, the main goals of the project were seen to be achieved by myself, my advisor and the initial users making the project a success.

Bibliography

Appendix & Bibliography word count = 2104 words

Ethical Considerations Form = 684 words

References

[1]Spam email statistics; https://www.propellercrm.com/blog/email-spam-statistics#:~:text=1.,falls%20into%20the%20spam%20category.

[2]Fraudulent email statistics; https://retruster.com/blog/2019-phishing-and-email-fraud-statistics.html

[3]Anti-Virus pricing; https://priceithere.com/antivirus-software-cost/

[4] Andronicus A. Akinyelu, Aderemi O. Adewumi, "Classification of Phishing Email Using Random Forest Machine Learning Technique", Journal of Applied Mathematics, vol. 2014, Article ID 425731, 6 pages, 2014. https://doi.org/10.1155/2014/425731

[5] Sarwat N, Menon N, Glasdam M, Nguyen DD (2014) Detection of fraudulent emails by employing advanced feature abundance. Egypt Inform J 15:169–174 https://www.sciencedirect.com/science/article/pii/S1110866514000280

[6] Yasin A, Abuhasan A (2016) An intelligent classification model for phishing email detection. arXiv preprint https://arxiv.org/ftp/arxiv/papers/1608/1608.02196.pdf

[7] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pp. 649–656, Alberta, Canada, May 2007; https://dl.acm.org/doi/10.1145/1242572.1242660

[8] N. Zhang and Y. Yuan, Phishing Detection Using Neural Network, Apr. 2013, http://cs229.stanford.edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork.pdf.

[9] A. ALmomani, T.-C. Wan, A. Altaher et al., "Evolving fuzzy neural network for phishing emails detection," *Journal of Computer Science*, vol. 8, no. 7, pp. 1099–1107, 2012; http://thescipub.com/abstract/10.3844/jcssp.2012.1099.1107

[10] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: a machine learning approach," in *Soft Computing Applications in Industry*, pp. 373–383, Springer, Berlin, Germany, 2008; https://link.springer.com/chapter/10.1007/978-3-540-77465-5_19

[11] Apache Software Foundation, "Spam assassin homepage," 2006; http://spamassassin.apache.org/

[12] J. Nazario, "Phishing corpus homepage," 2006; http://monkey.org/%7Ejose/wiki/doku.php?id=PhishingCorpus

[13] S. Youn, D. McLeod A comparative study for email classification, Advances and innovations in systems, computing sciences and software engineering, Springer, Netherlands (2007), pp. 387-391

https://link.springer.com/chapter/10.1007/978-1-4020-6264-3_67

[14] D.K. Renuka, T. Hamsapriya Email classification for spam detection using word stemming Int J Comput Appl, 1 (2010), pp. 45-47

http://www.academia.edu/download/34012668/pxc387241.pdf

[15] Graham P. A plan for Spam. <http://www.paulgraham.com/spam.html>

[16] S. Nizamani, N. Memon, U.K. Wiil, P. Karampelas Modelling suspicious email detection using enhanced feature selection Int J Model Optim, 2 (4) (2012), pp. 371-377

https://arxiv.org/abs/1312.1971

[17] S. Appavu, M. Pandian, R. Rajaram Association rule mining for suspicious email detection: a data mining approach Intelligence and security informatics, IEEE (2007), pp. 316-323 https://ieeexplore.ieee.org/abstract/document/4258717

[18] Chandrasekaran M, Narayanan K, Upadhyaya S. Phishing email detection based on structural properties. In: NYS cyber security conference; 2006. p. 1–7 https://www.albany.edu/iasymposium/proceedings/2006/chandrasekaran.pdf

[19] S. Nizamani, N. MemonCEAI:CCM-based email authorship identification model Egypt Inf J, 14 (3) (2013), pp. 239-249, 10.1016/j.eij.2013.10.001

[20] Sami Smadi, Nauman Aslam, Li Zhang, Rafe Alasem, M A Hossain, "Detection of Phishing Emails using Data Mining Algorithms", 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2015

[21] F. Toolan and J. Carthy, "Phishing detection using classifier ensembles," in eCrime Researchers Summit, 2009. eCRIME'09. IEEE, 2009, pp.1–9.

[22] Mayank Pandey and Vadlamani Ravi, "Detecting phishing e-mails using Text and Data mining", IEEE International Conference on Computational Intelligence and Computing Research 2012.

[23] Sunil B. Rathod, Tareek M. Pattewar, "Content Based Spam Detection in Email using Bayesian Classifier", IEEE ICCSP conference, 2015.

[24] Lew May Form, Kang Leng Chiew, San Nah Szeand Wei King Tiong, "Phishing Email Detection Technique by using Hybrid Features", IT in Asia (CITA), 9th International Conference, 2015.

[25] Tareek M. Pattewar, Sunil B. Rathod, "A Comparative Performance Evaluation of Content Based Spam and Malicious URL Detection in E-mail", IEEE International Conference on Computer Graphics, Vision, and Information Security (CGVIS), 2015.

[26] Prajakta Ozarkar, & Dr. Manasi Patwardhan," Efficient Spam Classification by Appropriate Feature Selection", International Journal of Computer Engineering and Technology (IJCET), ISSN 0976 – 6375(Online) Volume 4, Issue 3, May – June (2013).

[27] BCS The Chartered Institute for IT, "BCS Code of Conduct," 5 June 2019. [Online] [Accessed 27th October 2020] Available: https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/

[28] Difference between scam and spam;

https://www.webroot.com/gb/en/resources/tips-articles/spam-vs-phishing#:~:text=The%20difference%20between%20spam%20and%20phishing%20is%20that%2C%20while%20they,unsolicited%20emails%20to%20bulk%20lists.

[29] Official Tokenisation Definition; https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/#:~:text=What%20is%20Tokenization%20in%20NLP%3F,-Tokenization%20is%20one&text=Tokenization%20is%20essentially%20splitting%20a,smaller%20units%20are%20called%20tokens.

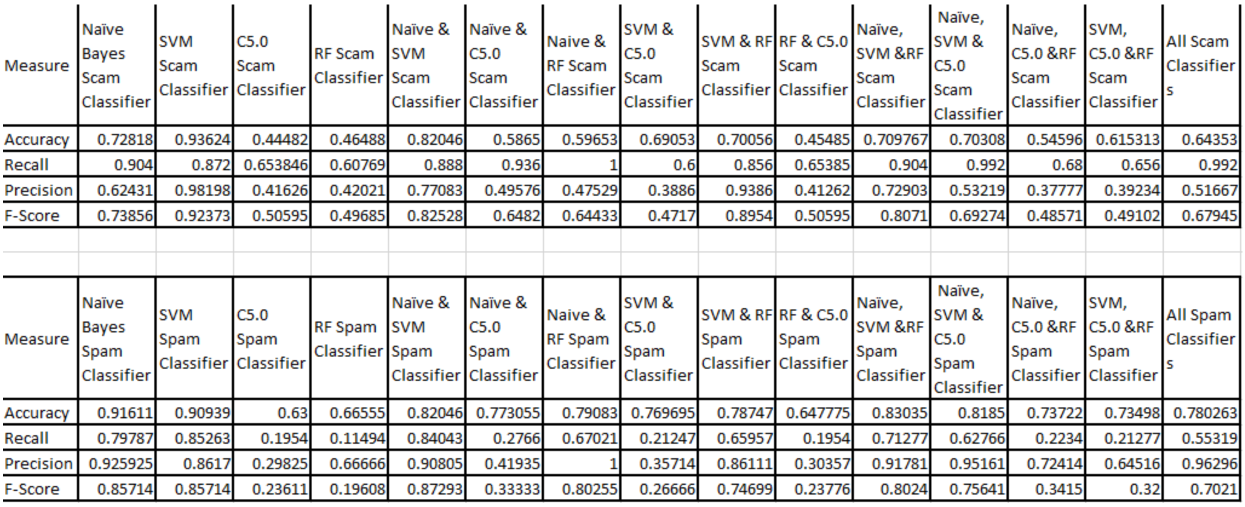
[30] Definition for Stop Words; https://whatis.techtarget.com/definition/stop-word

[31]Facebook Data; https://sproutsocial.com/insights/new-social-media-demographics/

[32] Outlook Data; https://www.lifewire.com/how-many-email-users-are-there-1171213

Appendix

Classifier Comparisons Full Data



Measure	Naïve Bayes Scam Classifier	SVM Scam Classifier	C5.0 Scam Classifier	RF Scam Classifier	Naïve & SVM Scam Classifier	Naïve & C5.0 Scam Classifier	Naive & RF Scam Classifier	SVM & C5.0 Scam Classifier	SVM & RF Scam Classifier	RF & C5.0 Scam Classifier	Naïve, SVM &RF Scam Classifier	Naïve, SVM & C5.0 Scam Classifier	Naïve, C5.0 &RF Scam Classifier	SVM, C5.0 &RF Scam Classifier	All Scam Classifiers
Accuracy	0.72818	0.93624	0.44482	0.46488	0.82046	0.5865	0.59653	0.69053	0.70056	0.45485	0.709767	0.70308	0.54596	0.615313	0.64353
Recall	0.904	0.872	0.653846	0.60769	0.888	0.936	1	0.6	0.856	0.65385	0.904	0.992	0.68	0.656	0.992
Precision	0.62431	0.98198	0.41626	0.42021	0.77083	0.49576	0.47529	0.3886	0.9386	0.41262	0.72903	0.53219	0.37777	0.39234	0.51667
F-Score	0.73856	0.92373	0.50595	0.49685	0.82528	0.6482	0.64433	0.4717	0.8954	0.50595	0.8071	0.69274	0.48571	0.49102	0.67945

Measure	Naïve Bayes Spam Classifier	SVM Spam Classifier	C5.0 Spam Classifier	RF Spam Classifier	Naïve & SVM Spam Classifier	Naïve & C5.0 Spam Classifier	Naive & RF Spam Classifier	SVM & C5.0 Spam Classifier	SVM & RF Spam Classifier	RF & C5.0 Spam Classifier	Naïve, SVM &RF Spam Classifier	Naïve, SVM & C5.0 Spam Classifier	Naïve, C5.0 &RF Spam Classifier	SVM, C5.0 &RF Spam Classifier	All Spam Classifiers
Accuracy	0.91611	0.90939	0.63	0.66555	0.82046	0.773055	0.79083	0.769695	0.78747	0.647775	0.83035	0.8185	0.73722	0.73498	0.780263
Recall	0.79787	0.85263	0.1954	0.11494	0.84043	0.2766	0.67021	0.21247	0.65957	0.1954	0.71277	0.62766	0.2234	0.21277	0.55319
Precision	0.925925	0.8617	0.29825	0.66666	0.90805	0.41935	1	0.35714	0.86111	0.30357	0.91781	0.95161	0.72414	0.64516	0.96296
F-Score	0.85714	0.85714	0.23611	0.19608	0.87293	0.33333	0.80255	0.26666	0.74699	0.23776	0.8024	0.75641	0.3415	0.32	0.7021

Interim Report

Day of Meeting	**What was Discussed**	**What should have been Completed**	**What was Completed**
01/10/20	Went over the original proposal and talked about expanding it	The Proposal	The original proposal
8/10/20	What to include in the Interim Report e.g. methodology, specification, main components What data sets to get, possible extensions	Expanded Proposal	Expanded Proposal
15/10/20	Looking through data sets, what's included in them, how old are they, bias Different papers	Work on report	Collected datasets and papers
22/10/20	What to include in related work, Bibliography, What to say about datasets	Continue work on report	Introduction, updated Tasks (methodology)
29/10/20	Creating a Questionnaire, what question the project should answer and editing datasets, How to do code of conduct and ethical section	Continue work on report	Read papers and datasets
5/11/20	The questionnaire draft, ethical considerations and BCS conduct was completed properly,	1st draft of report	Code of conduct, ethical section, questionnaire outline, 1st Draft of Report, 1st draft of Questionnaire
12/11/20	The finished report, what to do next, questionnaire	Interim Report	Submit Interim Report
29/01/21	Questionnaire Data	Questionnaire	Questionnaire
5/02/21	Controller Class Processes	Data Set Clean Up	Data Set Clean Up

		Pre-processor Class	Pre-processor Class
12/02/21	If links should be used in classifier	Controller Class	Controller Class
19/02/21	General Catchup	Retriever Class	Retriever Class
26/02/21	Classifier Class Processes	Spam Classifier	Spam Classifier
5/03/21	Classifier Types needed for research	Scam Classifier	Scam Classifier
12/03/21	Starting Dissertation intro	Classifier Class Alpha, Sender Class	Classifier Class Alpha, Sender Class
19/03/21	Dissertation results section, Fixing classifier bugs	Dissertation intro	Dissertation intro
26/03/21	Dissertation Progress, Beta Program Requirements	Dissertation tasks section	Dissertation tasks section
19/04/21	Initial user test, Dissertation improvements	1st Draft of Dissertation, Classifier Research	Dissertation, Research, Initial User test

Original Proposal

Hello and Welcome, I'm proposing to research, design, and create an Artificial Intelligence program that will be able to tell if a given email is spam or not. In theory, this would work with an email being sent to an email address that the program can access and then using Natural Language Processing techniques it shall read it to gather keywords and their meanings. These can then be fed to a Machine Learning algorithm that shall be trained on large sets of known scam and spam emails that have previously been created by researchers. This shall then return an email as a response to the sendee saying whether a given email is real or a scam/spam. A proposed extension could be to research into the creation of a function to generate and reply to these emails in order to waste scammers' time. All of the above will require research on how scams and spam emails operate as well as what words carry the largest significance and other ways an algorithm could be trained on detecting if an email or message of scam and spam. **Specification**: Ability to Receive emails, Ability to send emails, Ability to recognize what emails are Scam/Spam emails, (Extra) Ability to write emails to scammers

Cover Letter from the Questionnaire

The following questionnaire will ask you questions about how you interact with scam emails on the internet to better design an algorithm to counter them. These emails include simple spam emails to more sophisticated phishing attacks.

The information gained here shall be kept completely anonymous and will be used to inform the best direction for the program to develop by selecting what objects and patterns it should look for and not used for any other purposes. The completed program shall receive emails forwarded by users and will reply to the user to tell them whether or not the forwarded email is genuine or one of the other categories. After the program is completed your data shall not be stored and will be deleted.

Contact Details for questions and opting out:

Student	Technical Supervisor
Christopher Greer	Julie Weeds
cg394@sussex.ac.uk	juliewe@sussex.ac.uk

Ethical Compliance Form

This form should be used in conjunction with the document entitled "Research Ethics Guidance for UG and PGT Projects".

Prior to conducting your project, you and your supervisor will have discussed the ethical implications of your research. If it was determined that your proposed project would comply with **all** of the points in this form, then both you and your supervisor should complete and sign the form on page 3, and submit the signed copy with your final project report/dissertation.

If this is not the case, you should refer back to the "Research Ethics Guidance for UG and PGT Projects" document for further guidance.

1. Participants were not exposed to any risks greater than those encountered in their normal working life.

   Investigators have a responsibility to protect participants from physical, mental and emotional harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, physical hazards or discomfort, emotional distress, use of sensory deprivation (e.g. ear plugs or blindfolds), sensitive topics (e.g. sexual activity, drug use, political behaviour, ethnicity) or those which might induce discomfort, stress or anxiety (e.g. violent video games), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback.

2. The study materials were paper-based, or comprised software running on standard hardware.

   Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, mobile phones, and tablet computers is considered non-standard.

3. All participants explicitly stated that they agreed to take part, and that their data could be used in the project.

   Participants cannot take part in the study without their knowledge or consent (i.e. no covert observation). Covert observation, deception or withholding information are deemed to be high risk and require ethical approval through the relevant C-REC.

   If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, the data is to be published or there are future secondary uses of the data), then it will be necessary to obtain signed consent from each participant. Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script (see Appendix 1).

4. No incentives were offered to the participants.

   The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle. People volunteering to participate in research may be compensated financially e.g. for reasonable travel expenses. Payments *made to individuals must not be so large as to induce individuals to risk harm beyond that which they would usually undertake.*

5. No information about the evaluation or materials was intentionally withheld from the participants.

   Withholding information from participants or misleading them is unacceptable without justifiable reasons for doing so. Any projects requiring deception (for example, only telling participants of the true purpose of the study afterwards so as not to influence their behaviour) are deemed high risk and require approval from the relevant C-REC.

6. No participant was under the age of 18.

   Any studies involving children or young people are deemed to be high risk and require ethical approval through the relevant C-REC.

7. No participant had a disability or impairment that may have limited their understanding or communication or capacity to consent.

   Projects involving participants with disabilities are deemed to be high risk and require ethical approval from the relevant C-REC.

8. Neither I nor my supervisor are in a position of authority or influence over any of the participants.

   A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any study.

9. All participants were informed that they could withdraw at any time.

   All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script (see Appendix 1).

10. All participants have been informed of my contact details, and the contact details of my supervisor.

    All participants must be able to contact the investigator and/or the supervisor after the investigation. They should be given contact details for both student and supervisor as part of the debriefing.

11. The evaluation was described in detail with all of the participants at the beginning of the session, and participants were fully debriefed at the end of the session. All participants were given the opportunity to ask questions at both the beginning and end of the session.

    Participants must be provided with sufficient information prior to starting the session, and in the debriefing, to enable them to understand the nature of the investigation.

12. All the data collected from the participants is stored securely, and in an anonymous form.

    All participant data (hard-copy and soft-copy) should be stored securely (i.e. locked filing cabinets for hard copy, password protected computer for electronic data), and in an anonymised form.

Project title: Machine Learning Anti-Scam Program

Student’s Name: Christopher Frank Greer

Student’s Registration Number: 21705074

Student's Signature: CGreer

Date: 29/10/20

Supervisor's Name: Julie Weeds

Supervisor's Signature: JWeeds

Date: 10/11/2020