Bachelor of Science in Computer Science & Engineering

Content Based Email Spam Classifier as a Web Application Using Naïve Bayes Classifier

by

Arpita Chakraborty

ID: 1504109

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

May, 2021

Content Based Email Spam Classifier as a Web Application Using Naïve Bayes Classifier

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Arpita Chakraborty
ID: 1504109

Supervised by

Muhammad Kamal Hossen
Associate Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.
May, 2021

The thesis titled '**Content Based Email Spam Classifier as a Web Application Using Naïve Bayes Classifier'** submitted by ID: 1504109, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman

Muhammad Kamal Hossen
Associate Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Md. Mokammel Haque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Dr. Abu Hasant Mohammad Ashfak Habib
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Signature of the candidate

Date:

Acknowledgements

At first, I like to thank GOD for successful completion of this project. I owe to my deepest gratitude to my honorable project supervisor Muhammad Kamal Hossen, Associate Professor, Department of Computer Science & Engineering, Chittagong University of Engineering & Technology (CUET) for his valuable suggestion, constructive advices, encouragement and guidance throughout my entire project. I am indebted to all the respected teachers of the department in many ways hence it's an honor for me to thank them. Lastly I like to take this offer to express my regards and blessings to all of those who supported me in any respect during the completion of the project. Finally, I want to thank my father and mother for their unconditional love, support, encouragement, and contribution all throughout my life and academic career in every aspect along the years.

Abstract

Spam has emerged as a big issue that is endangering the reliability of existing mail networks, as it is used to distribute worms, malware, and Trojans, as well as scams of a more straightforward financial kind. Because of the large number of internet users, email has become an important means of information sharing around the world, whether for personal or commercial purposes, because it is a convenient and low-cost method of communication. Spam emails have long been a source of concern for computer security. Spam is detected using a variety of methods. We used various types of Bayesian algorithms to implement "Naive Bayesian filtering," one of the most sophisticated and common spam filtering techniques. To evaluate the content of the provided text, this algorithm employs the Bayes' theorem of probability principle. It determines the overall spam likelihood by calculating the probability of spam per each token of the received text. In the algorithm, we've introduced a few new features. Taking terms and whole sentences as tokens and attribute vectors was the most difficult function to implement. The addition of sentences to the training datasets improved the performance of spam and ham detection. We use the Natural Language Toolkit (NLTK), a language processing tool that helps us tokenize sentences and interpret the context of similar sentences to some degree. During the initial experimental phase, the filter's efficiency is primarily determined by tokenization and preparation. We suggested a spam filtering approach based on the Nave Bayes Classification Algorithm that can be deployed as a web application to separate spam from ham emails.

Key words: Email Classification, Spam Filter, Spam Detection, Naive Bayes Classification, Filtering Web Application.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Since the internet has changed our communication habits in every area of our life, it has also made revolutionized enterprise and socializing more effective and easier than ever before. Electronic mail (also known as email) has grown in popularity in recent years. These emerging inventions, on the other hand, have spawned a new set of concerns and challenges. The so-called "Spam" is one of the big issues that nearly all e-mail users face. Spam is unsolicited commercial e-mail that is usually delivered in bulk to thousands, if not millions of subscribers. While most email clients may see spam as a new virus. Spam can be seen as the outcome of various kinds of email-borne infections, including Trojans or zombies. Similarly, some people can mark e-mail from a business in which they have done business as spam [1].On the other hand, spam that appears in our e-mail inboxes is almost unanimously recognized and some also predict that it will lead to the death of e-mail. A variety of spam detection and filtering approaches have been created to deal with the issues. This initiative studies, tests and implements cutting-edge anti-spam technology.

1.2 Framework/Design Overview

The main concept of our project is building a spam filter for emails. So this is our main goal. Now to design a spam filter we have many more options like :

Content filters – Consider the content of messages in this filter, searching for keywords widely found in spam emails or messages.

Header filters – Look for unusual details in the email header source (such as

spammer email addresses).

Rules-based filters – Use guidelines created by the company to block emails from particular senders or those with specific terms in the subject or body..

URL-based filters – This is the most recent email filtering strategy. Different researchers have done different works in this technology to distinguish between desired and unwanted email connections. In a different form of this method, spam is filtered 24 hours a day, seven days a week, and the networks are reviewed as a result.

We can consider our spam filter using **content based** techniques using machine learning concepts. So the basic task is represented by a figure below:

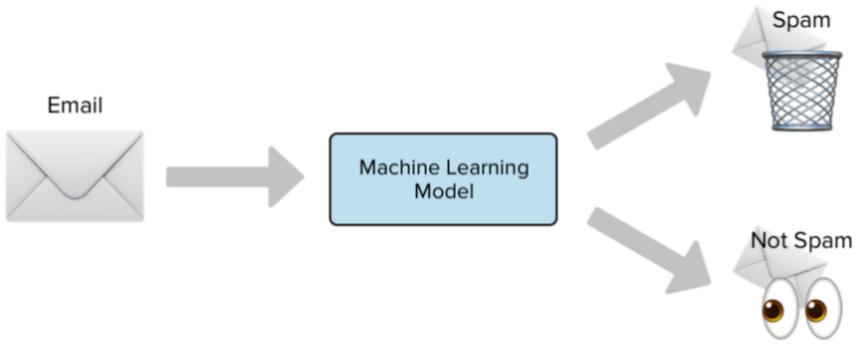

Figure 1.1: Common architecture

Now, we can explain the common workflow of our project in briefly. The overall system of email spam filtering goes through the following steps:

1. At first, we collected different types of mails and labelled them in ham or spam according to their content.
2. Then we trained a machine learning model which can be trained to identify these two types of email.
3. To train this model, we have to define some criteria or features for prediction.
4. After that, we tested a new email without labeling ham/spam.
5. Finally, this model is given our desired result.

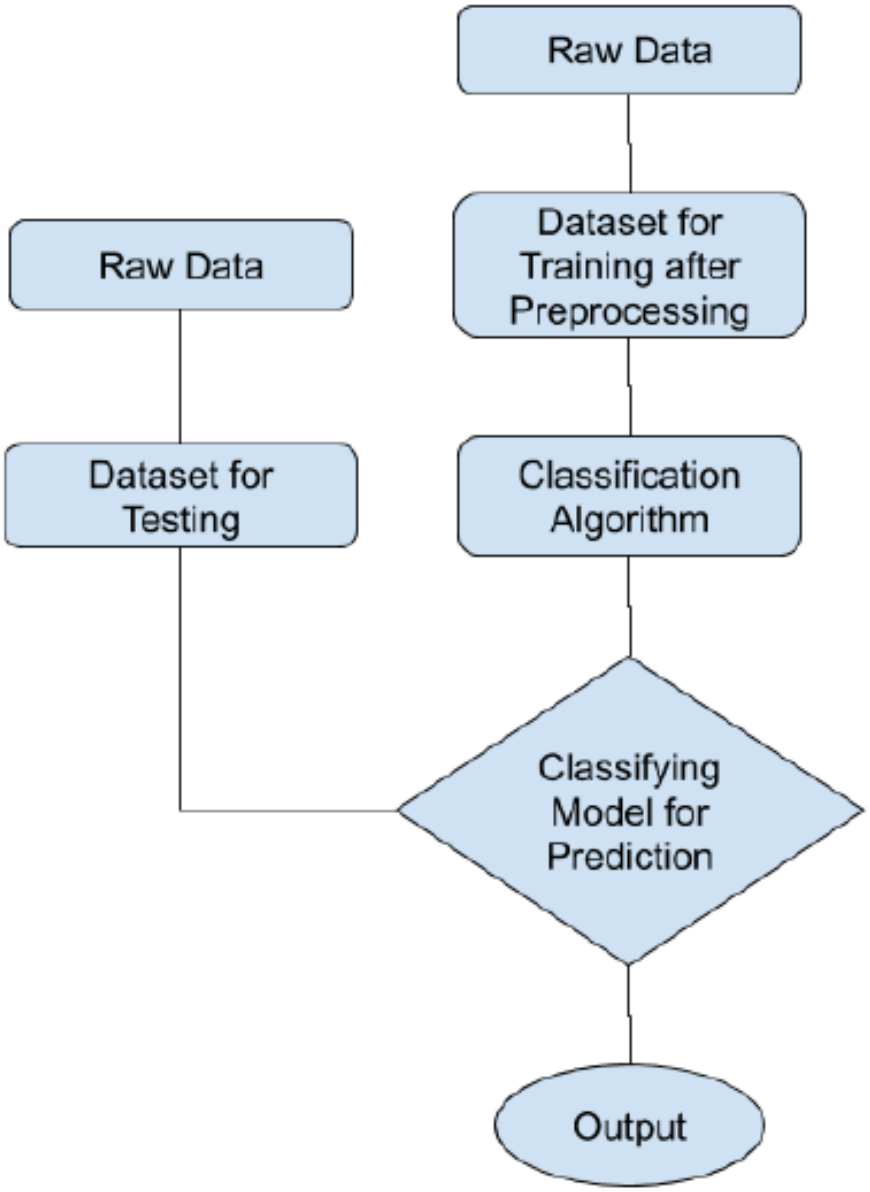

Figure 1.2: The workflow of the system

1.3 Difficulties

Spam differs from other cyber risks in many ways. It may seem to be more of an annoyance than a hazard at first glance, but it does have an effect. We might end up with hundreds of spam messages in our inbox if our spam filter is 95 percent accurate[2]. This is a lossy consistency situation, and it has a direct and immediate impact on consumer loyalty. What was once considered state-of-the-art precision is now woefully insufficient. It's not just that people's aspirations have risen, there's still a lot more spam now than there used to be. That is the unspoken explanation why spam filter accuracies must be improved on a regular basis. Furthermore, no discussion of accuracy or efficacy will be accurate without including false positives. Users are understandably enraged as they find genuine messages in their spam archive. As a result, developing a spam filter that can produce more reliable results while reducing false positive rates is far more complex [3]. We attempted to solve this issue by lowering the error rate in this project.

1.4 Applications

There are many applications in real life for this project.

- Any business companies use spam filter to protect their clients information [4].
- The use of spam filtering methods is applied to all incoming email as well as outgoing email.
- This technology helps internet service providers secure the data of their clients by removing unwanted or suspicious email by using various spam-filtering techniques.
- We can use spam filter to block threats.
- Spam filtering techniques are used to filter legitimate emails.
- We can use it to meet data regulations.

- Anyone can use this to protect their business reputation.

1.5 Motivation

People nowadays have work and studies completely reliant on the use of the internet, or don't enjoy this high standard of living because of the amount of time spent online. They have a higher standard of living because they spend so much time online. No one today will live a total lifestyle without having a phone, and the internet. Email is the most cost-effective and reliable mode of communication. The exponential advancement in information and communication technologies has brought with it both benefits and drawbacks, such as spam. For internet users, spam has been a nightmare. It clogs up inboxes, takes up disk space, and wastes resources due to the dial-up link. Spam messages take a long time to uninstall, and users waste a lot of time doing so. Furthermore, spam notifications use a lot of bandwidth and use up a lot more database capacity, leaving the internet more congested and sluggish. Over the last few years, spam sending processing has advanced significantly. Anti-spam apps did the same thing. Anti-spam software developers save businesses millions of money, and people who have vulnerabilities for internet shopping, gaming, and other activities support them[1]. However, no anti-spam software currently in development is a complete solution to the spam crisis. Both knowledge about spam messages and their effects on individuals, as well as strategies for overcoming problems created by spam, were discovered in order to begin developing the whole project. Reading and analyzing scholarly articles and books yielded relevant material[5] . For instance, Henry Stern wrote an article about the three most common spam sending tools and their improvements in his article "A Survey of Modern Spam Tools." He explained how spammers use the [6] approach to get around content filtering. Professors at Georgia Tech wrote a paper on the impact of denial-of-service attacks. They highlighted two separate spam targets to attack: humans and computers. Another research paper, written by a University of Cambridge PhD undergraduate, describes the real behavior of spam attacks in the United Kingdom [7]. The University of Michigan and Microsoft Research Silicon Valley collaborated to create one of the functional

articles. They discussed triangular spamming, which is the most recent tactic used by spammers to get through filters [8]. Because of the knowledge obtained from research papers, the need to develop a creative anti-spam software package for our project has grown.

1.6 Contribution of the thesis

There are several ways to combat spam, and clever spammers get through any given anti-spam measures; no one approach can effectively filter it because spammers are acutely aware of current spam-filtering tactics and skilled enough to mitigate it. The primary aims of this project are listed below:

1. We built a system which can minimize the error rate that means ham messages will not be considered as spam.
2. Modern email system contains HTML data with the message. To detect intelligent spam, we also considered HTML tags and comments as token.
3. We took the words and sentences simultaneously in the trained dataset. So when evaluating the content of an e-mail, the filter performs text analysis as well as checking for grammar and language, getting a more productive outcome.
4. Built a spam filtering model which can be implemented as a web application that works both offline and online system.

1.7 Thesis Organization

The following chapters will go through the different aspects of the project works in details. Various aspects and types of spam filters, their comparative analysis and their structure will be discussed in details. Our goal will be discussed from various points of view. This chapter provides an introductory concept of the work done. In chapter 2, we give a brief description about previous works that is already implemented in this field and also mention their limitations. In chapter 3, Naïve Bayesian filter will be discussed along with Bayesian formula of probability

and we will see how it can be used to detect spam. The building of Bayesian model will be discussed. And at the last the comparative analysis among filters will be done. In chapter 4, we will discuss about our project and analysis of its output. And finally in chapter 5, we will discuss about possible development in the future on the project and draw a conclusion.

1.8 Conclusion

In this chapter, we have discussed our aims and objectives. The main design concept, different types of challenges and for what reasons we chose this topic are also described here. The chapter gives a basic idea of the project.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we will describe the related work in this particular field. After analyzing their shortcoming, we will describe how the limitations can be improved by the new system. Later, we will describe briefly about the resources that are used in this project for full functionality.

2.2 Related Literature Review

There are various papers have been published showing various methods for email spam classification.Using a python method (Naive Bayes) that integrates semantic and keyword analysis with machine learning can expand Naive Bayes by two hundred percent[1]. The calculation was also actualized and tried progressive conditions over the Internet.

This article[8] uses a Naive Bayes and Hidden Markov models in tandem to accurately determine how people's moods impact on the outcome of various issues. One of the many possible ways in which we might use is to define just the portion of the email as text and then extract it from the text. In this article, terms and sentences with respect to relative aspects are examined in detail

This email classifier was first built as a year ago as described in [9] for users to install and upload their own Bayesian software and apply a basic threshold. Author [10] recommended using Bayesian filters as a year ago as part of a naive classifier, but later web services provide them. In order to properly apply these methods, we need a huge data collection.

Some previous works based on spam filtering using different types of models and

the advantage and disadvantage of this existing solutions are described briefly in the following table:

Table 2.1: Related Works and its advantage, disadvantage

Author	Techniques used	Advantage	Disadvantage
S. Peng et al.[5]	Naïve Bayes classification algorithm.	Most simple to implement and less complex.	The speed and accuracy of the system are less than the other system.
S. Wang et al.[1]	Fuzzy-SVM and k-means	The precision increases as the compression ratio increases.	Complex to implement and required large sample data is required for training.
Sebastian Romy Gomes et al.[6]	Naïve Bayes and Hidden Markov Model	The accuracy of the system improved.	The system is not capable to counter poison attack.
Seongwook Youn et al.[9]	Two-level Ontology-based classifier.	The system is suitable when the requirement of the classification model is customized according to the need of the user.	Need focus on the misclassified legitimate emails and also on the overfitting of the filter.
Weiweng Yang et al.[8]	Various combination of classification algorithms.	This experiment by the author gives various methods for classification	Performed poor in some case.

Every approach discussed above is efficient, but each strategy has limitations. It's impossible to think of a solution that has complete accuracy of 100 percent, so one must compromise on any number of false positives or the amount of false positives

2.2.1 What is Spam?

The Australian Communication Authority defines spam as following “Unsolicited commercial electronic messaging”[11].The word “Spam” as applied to email means Unsolicited Bulk Email (UBE). Until approval has been given by the letter may be received, the receiver, it may be regarded as unsolicited. The substance of the item is shared with other similar items with a similar to it would be submitted along with the whole set of messages, making the message even bigger. An e-mail address is considered spam, if it is both unwanted and if it is unsolicited. Inquiries made with no one asks for, referred to as usual ones (or popular ones), and queries that gather plenty of responses are referred to as automated ones (or channeled inquiries (example: subscriber newsletters, customer communications, discussion lists) [12].

2.2.2 A Brief History of Spam

The new phenomenon of email spam first came to public attention in 1978, but has been very prominent over the last decade or so, referring back to the years between 2002 and 2012. In early 2002, about 16% of all email sent via the internet was junk; by the start of 2008, junk emails represented between 87% and 95% of all internet traffic. But, Ironically, the proportion of email that has already been classified as spam subterfuge a bigger problem. The amount of spam being received each day elaborate expanding. In other words, for instance, during the year 2002, there were less than a hundred billion spam emails. Currently, there are tens of billions of spam emails sent every day. Furthermore, spam volumes will increase dramatically in a brief amount of time, as shown by the doubling of spam volume from May to November 2006, as well as spam "spikes" during the Christmas season and other apparently unpredictable periods [13].

2.2.3 Reason of sending spam

Some people send junk email because it makes more financial sense to do so. If the campaign costs just 500, a spammer will send tens of millions of emails due to the little work required (it’s that easy to hack other people into their computers),

the price is completely zero. If only one person out of ten thousand decides to buy a product, then the spammer will become wealthy.

2.2.4 Cost of Spam

There are variety of problems caused by spam:

1. Avoiding a problem of spam in an organization's network consumes resources that it can then be used for valuable purposes.
2. Spam creates unnecessary storage requirements in an organization
3. Organization that does not have an adequate spam filter may face a huge loss of employee productivity. Which is a major issue in competitive organizations.
4. Breaching of security using spam is also a major factor. Some spammers uses phishing attempts to enter a one's confidential information by sending spam[2].

2.2.5 Techniques of Spammers

The keys to why the spam problem is so bad and is getting worse can be distilled down to three factors[12].

1. Most of the spammers are smart at what they do.
2. Spammers can make money even with extraordinary low conversion rates.
3. They have at their disposal lots of money for development of new delivery techniques.

Given that the money is being made by spammers, which totals tens of millions of dollars per year, can fund newer and better techniques for distributing spam, this provides a ready source of finding for spam development[14]. Among the techniques that spammers use to distribute their contents are:

1. **Filter-circumvention techniques:** Spammers use simple technique to avoid filters by misspelling keywords, introducing valid text like Bible verses

into spam messages, using various HTML techniques to trick filters. For example, they use misspellings of word like replacing an "I" with "1".

2. **Botnets:** Traditionally, the spam spammers would have used less and more visible channels for sending email messages in order to make it easier to trace. More than half a million computers have been rendered "zombie" by a worm, malware, or some other internet evil actor into spambots that can be used by a remote abuser so these initiatives are met with zombie computers that help to overwhelm botnets and also for instance, thousands of computer viruses known as entity attachments can assist The benefits to the spammers may get from using botnets is that they can prevent detection by internet providers and anti-based analysis programs by sending a large amounts of messages from individual IP addresses are that have a lot.

3. **Newer types of spam:** A new wave of spammers had only recently started using more sophisticated spamming tactics to combat new spam filtering technology starting in 2006-2007. For example:

****Image-based spam:** Computers' traditional understanding of text recognition technology is countered by such innovations as image text that has been blurred, multicolored, sideways letters, "strikethoscopes," and various word/letter combinations that makesiness in order to increase textual identification difficulties for the spam. Because of the relatively large size of each post, image spam is even more trouble for those it is usually consumes more bandwidth and costs the sender a lot more money, as well.

****Spam with attachments:** Like image spam, but with material in a ZIP files as the payloads, this spam carrying content in a ZIP files

2.2.6 Various Spam Filters

2.2.6.1 Signature-based Filter

A lot of businesses implement heuristic (signature-based) content blocking, which is the most often used method for spam detection.It does some basic filtering to retain only the messages relevant to the requested commodity, before saving it

to a copy of it to a disk, which then verifies some patterns against the copy. The score is given out of how good the pattern fit the inventory is, or out of how many fitting it is. The greater the risk of an undesired email attachment or whatever that being filed in the subject line above.

One of the techniques used to assign a score is a pattern signature[7]. As one types in "unauthorized email", "unwanted email" or "bother email" will be collected by lab technicians will take a large number of expansions on the keyboard to appear. There are next there is a need for humans equipped with undesired email and signature gathering techniques to read the mail to go through it and collect signatures that are then sent to the customers (usually via very frequent product domains). A drawback to this solution is the significant latency that exists between when consumers get a signature update and when it is being done to ensure no false positives, however, the customers have to do a semi-manual job to catch backdated/malicious/phishing messages in between when it happens and when it is reported so those unauthenticated messages won't be missed.

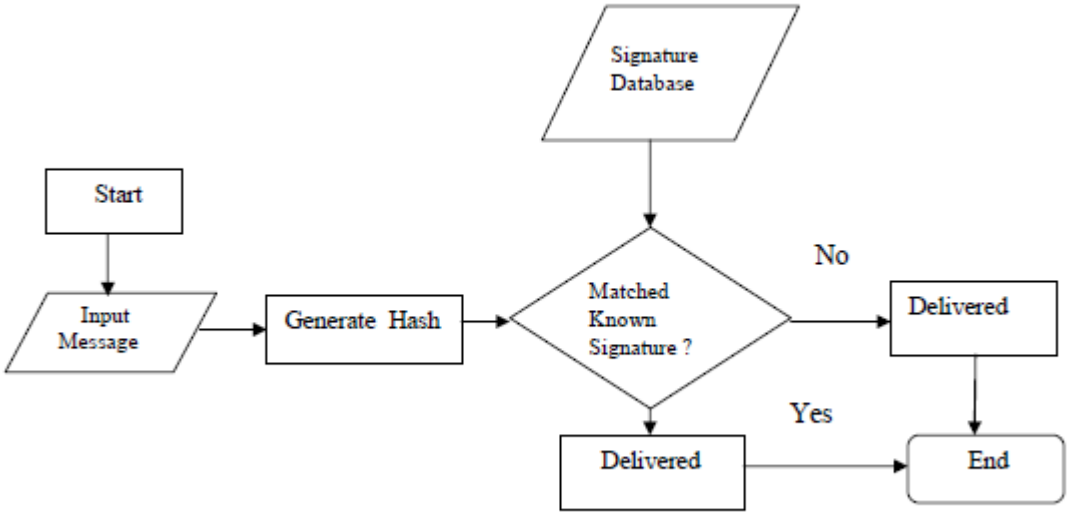

Figure 2.1: Signature based filtering

2.2.6.2 Learning-based Filter

When it is first loaded, the filter is conditioned by gathering and processing the examples of spam; afterwards, it expands the database from the examples itself

on a periodic basis. The method used here is a dynamic and our Bayesian spam filter is a bayesian one which is trained dynamically. The mail is extracted and all the device returns one token that is assigned to a one token value from each mailbox address that exists in the database. It can be seen in figure 2.2 that with the use of a Bayesian filter, a dictionary of terms is assembled, and the likelihood that one of them will be in a spam message is calculated.[7]. The more frequently a few words are found in spam communications, the less likely they are to be in actual emails, and/it is assumed that such words will appear in your target group's vocabulary. In one respect, Bayesian filtering is more closely resembles reputation-based methods than machine learning. We have the ability to tweak the method for the specific needs of the email system. As far as Bayesian filtering is concerned, the process of building the filter first, the initial training stage takes a significant amount of time. Upon the start of this training, the Bayesian monitor waits for inbound and outbound emails, and simultaneously collects a database (or modifies a database from a provider) that is specially made for that email traffic. A particular user can manually added spam can help to improve the database's (the e.g. assigned) Bayesian engine's) learning by incrementally moving its posterior probability closer to 0. After the training period, the Bayesian engine begins tagging spam. Because the training period introduces the engine to the organization's typical mail flow, the system often starts with significantly low false-negative and false-positive rates. Because the probability database is so important to a Bayesian filter, the vendor supplies a standard ham database and only spam is accepted for training[15]. Tuning the filter correctly in this case would have a huge impact on the performance because it is done incorrectly. Expanding the filter's capability to process to make use both ham and spam is beneficial since it will enable it to be more specific and reliable False positive is genuine mail that is detected as spam and false negative that is missed by the filtering. In order to keep the chance database up to date, the engine would be able to respond to a shift in the mail traffic. Such regular update activity in the model ensures that Bayesian engines can easily respond to the evolving of spam techniques. A classic countermeasure for trying to circumvent the effectiveness of a signature-based filter is to use arbitrary terms

and phrases at the bottom of each email address. However, Bayesian networks are susceptible to these kinds of modifications and alterations[16]. Since the entire email is checked rather than specific keywords requiring a match against a given rule, Bayesian filters are substantially more effective than rule-based filters. A rule-based scheme, on the other hand, can have a lower false-positive rate. The use of more processors on email systems is a major disadvantage of Bayesian filtering, particularly if the anti-spam engine is hosted on the mail server itself.

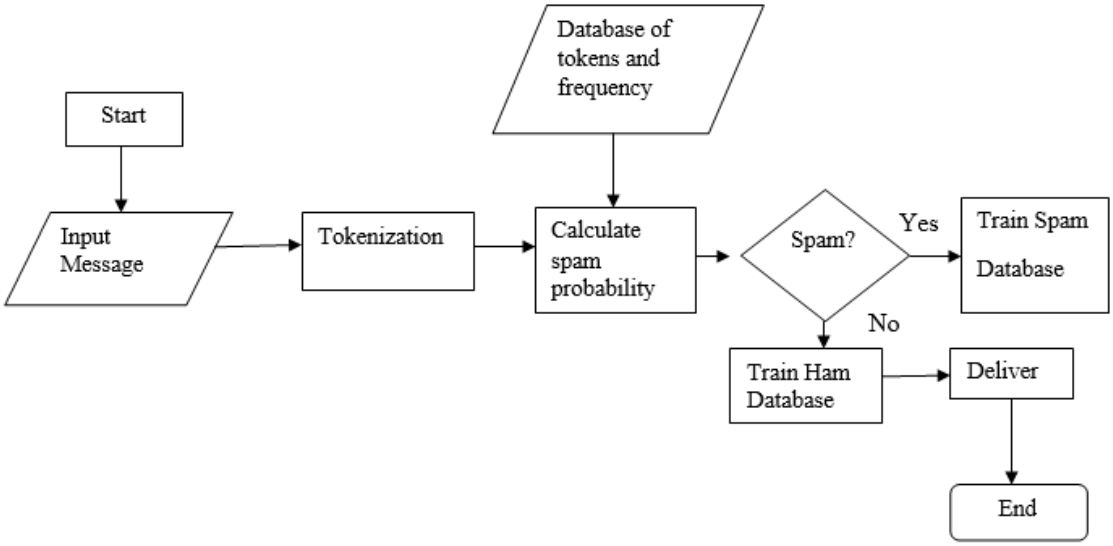

Figure 2.2: Learning based filtering

2.2.6.3 Rule-based Filter

With rule-based filtering, the user sets a list of keywords that will be used in the search results. When the words “cancer”, “danger”, or “spam” appear in an incoming mail, it is classified as such. Each keyword is stored in the database, and all the terms in the body are then compared to see if they are equal.That is seen in figure 2.3, which illustrates how the filter operates based on rules. Because of the false negative and positive errors in these algorithms, these expanders have a higher prevalence of false positives and false negatives. In addition to general keywords, that is because of the unique keywords.

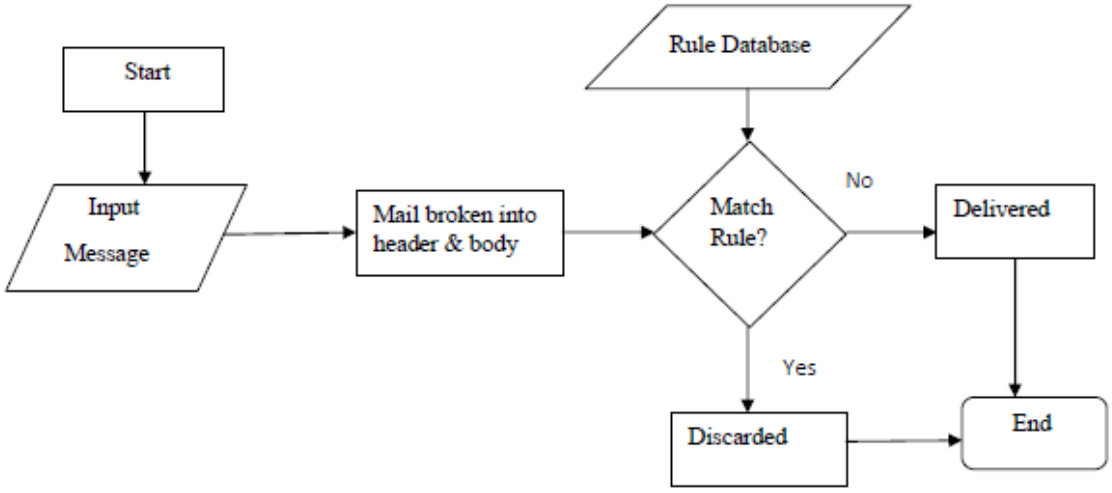

Figure 2.3: Rule based filtering

2.2.7 Bayes' Formula

Bayes' theorem is a procedure for calculating conditional probabilities. The formula also computes the posterior probabilities (as opposed to the prior probabilities) are generally known prior to the observations. Often, a hypothesis can be proposed to account for a few observations and Bayes' rule can be used to calculate the likelihood of that hypothesis being valid. We illustrate this idea with details in the following example:

Example: Mammogram posterior probabilities Approximately 1% of women aged 40-50 have breast cancer. A woman with breast cancer has a 90% chance of a positive test from a mammogram, while a woman without has a 10% chance of a false positive result. What is the probability a woman has breast cancer given that she just had a positive test?[14] Translate into the meaning of probability, let B = "the woman has breast cancer" and A = "a positive test". We wish to calculate P(B|A). Similar to what we did last time, we have:

$$P(B|A) = \frac{P(B \cap A)}{P(A)} = \frac{P(B \cap A)}{P(B \cap A) + P(B^c \cap A)}$$

To compute the probabilities on the right side, we use the multiplication rule.

$$\begin{aligned} P(B \cap A) &= P(B)P(A|B) = 0.01 \cdot 0.9 = 0.009 \\ P(B^c \cap A) &= P(B^c)P(A|B^c) = 0.99 \cdot 0.1 = 0.099 \end{aligned}$$

Therefore,

$$P(B|A) = \frac{0.009}{0.009 + 0.099} = \frac{9}{108}$$

This answer is somewhat surprising. Indeed, when ninety-five physicians were asked this question their average answer was 75%. The two statisticians who had done this survey indicated that physicians were better able to see the answer when the data was presented in frequency format. 10 out of 1000 women have breast cancer. Of these 9 will have a positive mammogram. However, of the remaining 990 women without breast cancer 99 will have a positive test, and again we arrive at the answer 9/(9 + 99)=9/108. We now state the Bayes' Formula: First, we have a partition B_1,B_2,...,B_n of a probability space, namely, their disjoint union is the total space. In the test example, we have

$$P(B_1|A) = \frac{P(B_1 \cap A)}{P(A)} = \frac{P(B_1 \cap A)}{\sum_i P(B_i \cap A)}$$

To evaluate the probability, observe that by the multiplication rule,

$$P(B_i \cap A) = P(B_i)P(A|B_i)$$

From this, we have the important Bayes' Formula:

$$P(B_1|A) = \frac{P(B_1 \cap A)}{P(A)} = \frac{P(B_1)P(A|B_1)}{\sum_{i=1}^{n} P(B_i)P(A|B_i)}$$

Understanding Bayes' formula can greatly enhance our ability to examine in real

life chance problems.

2.2.8 Naïve Bayes Formula in Spam Filtering

Bayesian email filters take advantage of Bayes' theorem. Bayes' theorem is used several times in context of spam:

1. Firstly, to compute the probability that the message is spam, knowing that a given word appears in this message.
2. Secondly, to compute the probability that a message containing word is spam.

Consider the word "replica" in the suspected message. Most people who aren't used to getting e-mail recognize this message as spam, specifically a solicitation to sell counterfeit watches of well-known brands. However, spam detection software cannot "know" such facts; it can only compute probabilities. The formula used by the software to determine that, is derived from Bayes' theorem

$$\mathbf{Pr}(S|W) = \frac{\mathbf{Pr}(W|S) \cdot \mathbf{Pr}(S)}{\mathbf{Pr}(W|S) \cdot \mathbf{Pr}(S) + \mathbf{Pr}(W|H) \cdot \mathbf{Pr}(H)}$$

where:

- Pr(S|W) is the probability that a message is a spam, knowing that the word "replica" is in it.
- Pr(S) is the overall probability that any given message is spam.
- Pr(W|S) is the probability that the word "replica" appears in spam messages.
- Pr(H) is the overall probability that any given message is not spam (is "ham").
- Pr(W|H) is the probability that the word "replica" appears in ham messages.

$$\mathbf{Pr}(S|W) = \frac{\mathbf{Pr}(W|S)}{\mathbf{Pr}(W|S) + \mathbf{Pr}(W|H)}$$

This quantity is called spamicity of the word "replica", and can be computed. The number Pr(W|S) used in this formula is approximated to the frequency of messages containing "replica" in the messages identified as spam during the learning phase. Similarly, Pr(W|H) is approximated to the frequency of messages containing "replica" in the messages identified as ham during the learning phase. For these approximations to make sense, the set of trained messages needs to be big and representative enough. It is also advisable that the learned set of messages conforms to the 50% hypothesis about repartition between spam and ham, i.e. that the datasets of spam and ham are of equal size. Of course, determining whether a message is spam or ham based only on the presence of the word "replica" is error-prone, which is why bayesian spam software tries to consider several words and combine their spamicities to determine a message's overall probability of being spam. Calculating individual probabilities We can calculate the overall probability of the mail to be a spam regarding all of the tokens or a set of tokens of the mail using the following formula:

$$p = \frac{p_1 p_2 \cdots p_N}{p_1 p_2 \cdots p_N + (1 - p_1)(1 - p_2) \cdots (1 - p_N)}$$

Where:

- p is the probability that the assumed message is spam.
- p1 is the probability Pr(S|W1) that it is a spam knowing it contains a first word (for example "replica").
- p2 is the probability Pr (S|W2) that it is a spam knowing it contains a second word (for example "watches").
- pN is the probability Pr (S|WN) that it is a spam knowing it contains an Nth word (for example "home").

  The result p is usually compared to a given threshold to decide whether the message is spam or not. If p is lower than the threshold, the message is considered as likely ham, otherwise it is considered as likely by spam.

2.2.9 Why Bayesian Filter?

All the techniques available to detect spam are static except the Bayesian filter. It is easy for the smart spammer to evade the static spam filter easily by making some modifications. But the Bayesian filter works on the content of the mail which is everything the spammer want to deliver. While other techniques are common to all type of users, Bayesian filter can be customized for individual user[12].

2.2.10 Natural Language Tool Kit (NLTK)

Our first task is to build a spam filter that takes an email as input and extracts the words and sentences from the message body. So that spam likelihood can be calculated using both word-to-word and sentence-to-sentence comparisons. We used the Natural Language Tool Kit (NLTK) in Python programming for this approach[6] The NLTK module is a large toolkit designed to assist us with all aspects of Natural Language Processing (NLP). NLTK will help us with everything from separating sentences from paragraphs to splitting up phrases, understanding the part of speech of those words, highlighting the key subjects, and even assisting your computer in comprehending the text[17]. The main purpose of using NLTK in our project is the tokenization of sentences and words[18]. NLTK can be used for the following purposes:

1. **Tokenizing words and sentences:** NLTK converts the sentences into a vector and also separates the words and stores them in a database called data dictionary

2. **Stop words with NLTK:** Words like “the”, “an”, “a”, “this” , “there” which are not necessary for identifying spam or ham. While tokenizing we won’t count the words by using the stop words list.

3. **Part of speech tagging:** The NLTK module will do Part of Speech tagging for users, which is one of its more powerful features. This involves identifying terms in a sentence as nouns, adjectives, verbs, prepositions, and so on.

4. **Stemming the word:** The idea of stemming is a sort of normalizing method. Many types of words carry the same meaning, other than when tense is involved. For example :" I was taking a ride in the car." And "I was riding in the car". The meaning are almost same. Stemming the word try to arrange this sentences as same.

5. **Lemmatizing with NLTK:** Lemmatizing is a process that is somewhat close to stemming. The main difference is that stemming will also produce non-existent words, while lemmas are real words, as we saw earlier. So, while our root stem (the word we end up with) isn't something we can look up in a dictionary, we can look up a lemma.

6. **Scikit-Learn Sklearn with NLTK:** Scikit-Learn is the method to call for the Library in python compiler. In NLTK, we can also use this. A new dataset can be tested corresponding to training dataset using Scikit-Learn as it provides access to many classifiers[19].

7. **Naïve Bayes classifier with NLTK:** We can choose many algorithms for training and testing. Naïve Bayes is one of the common and widely used algorithms. After seeding database, we can apply Naïve Bayes classifier for test classifications in NLTK.

2.3 Conclusion

The best practices outlined in this chapter became the most widely used techniques for classifying spam. However, it may take a global undertaking, and the process will take a long time. Users now need to defend themselves, and statistical filters are currently the most promising tool for doing so. They have superior efficiency, can instantly adjust as spam improves, and are computationally effective in many ways.

2.3.1 Implementation Challenges

Dealing with memory and defining keyword contexts were two of the obstacles encountered when building the filter. To store all of the possible words after

training the filter with a large number of texts, there must be substantial free space in the brain. The decision to solve this problem was to use a database, but accessing the database took time. As a result, at the conclusion of the tests, the decision was made to store all of the words in the hash table, allowing them to be saved in the main memory. As opposed to querying data from a database, accessing information from the main memory does not take long [2].

Chapter 3

Methodology

3.1 Introduction

This chapter gives a detailed outline of the software development methodology used in the project. The structured way and procedure of the project are defined and described. The system has generally many functional and non- functional requirements that are necessary to interact between different roles and the whole system.

3.2 Diagram/Overview of Framework

3.2.1 Sample Features Used in this Project

We divide the spam features into two groups based on the study of the features used in similar studies in the literature: features in the header, features in the payload(body)[13]. Figure 3.1 shows the hierarchy of the features categories. In the following subsections, we provide a description of each category and its related features.

Header features: The email header is a crucial component of any email address, since it contains a number of key features that aid email delivery. These characteristics are divided into two categories: email metadata and body. Table 3.1 lists twenty-five header features with succinct explanations culled from the literature.

Payload features: These features are categorized into three sub categories: Email body features, Readability features, and Lexical diversity features[20].

1. ***Email body features:*** The e-mail body contains unstructured data such

as images, HTML tags, text,and other objects. This features set contains 59 features,which are described in Table 3.2.

2. ***Readability features:*** In email readability terms, differentiating a section characteristics (i.e. complexity, weight, length, substance, style, focus) identify the capability of reading email sections. features are derived based on syllables — voice sounds — and are used to separate plain and complicated words this feature collection of features evaluates the body readability is greater than the other similar tests we have seen[20].

The readability features are discussed as follows:

- Simple word features (with and without stopwords) include the number of syllables in each word.
- Features of complex words (with and without stopwords), such as the use of three or more syllables in each word
- The number of syllables and the number of words in the text are used to create word length features (with and without stopwords).
- The Fog Index (FI) features (with and without stopwords), which are the most often used readability metric to quantify the years of education experience required to comprehend the text at a glance[20].
- Features of the Flesch Reading Ease Score (FRES) (with and without stopwords) are used to assess textual complexity.
- Features of the SMOG index (with and without stopwords), which are used to assess the difficulty of writing a letter.
- Features of the FORCAST index (with and without stopwords), which are used to assess the reading abilities of texts with a high proportion of plain words.
- Features of the Flesch-Kincaid Readability Index (FKRI) (with and without stopwords), which are similar to the previous features but weighted differently[20].

- Easy Word FI functions (with and without stopwords), which are similar to the Fog Index but use simple words instead..

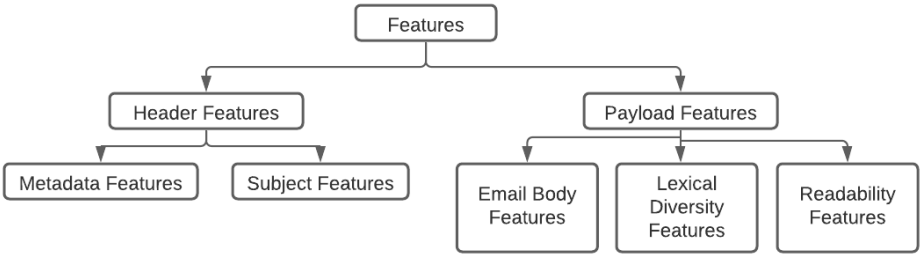

Figure 3.1: Features hierarchy

Table 3.1: The Email Header Features

ID	Feature Details	Type	Studies	ID	Feature Details	Type	Studies
1	Year	Metadata	[15]	26	Replay to MIL?	Metadata	[15]
2	Month	Metadata	[15]	27	Replay to Yahoo?	Metadata	[15]
3	Day	Metadata	[13] , [15]	28	Replay to AOL?	Metadata	[15]
4	Hour	Metadata	[13] , [15]	29	Replay to Gov?	Metadata	[15]
5	Minute	Metadata	[13] , [15]	30	X-Mailman-Version	Metadata	[15]
6	Second	Metadata	[13] , [15]	31	Exist Text/Plain?	Metadata	[15]
7	From Google?	Metadata	[15]	32	Exist Multipart/Mixed?	Metadata	[15]
8	From AOL?	Metadata	[15]	33	Exist Multipart/Alternative?	Metadata	[15]
9	From Gov?	Metadata	[15]	34	Number of characters.	Subject	[13]
10	From HTML?	Metadata	[15]	35	Number of capitalised words.	Subject	[13]
11	From MIL?	Metadata	[15]	36	Number of words in all uppercase.	Subject	[13]
12	From Yahoo?	Metadata	[15]	37	Number of words that are digits.	Subject	[13]
13	From Example?	Metadata	[15]	38	Number of words containing only letters.	Subject	[13]
14	To Hotmail?	Metadata	[15]	39	Number of words containing letters and number.	Subject	[13]
15	To Yahoo?	Metadata	[15]	40	Number of words that are single letters.	Subject	[13]
16	To Example?	Metadata	[15]	41	Number of words that are single digits.	Subject	[13]
17	To MSN?	Metadata	[15]	42	Number of words that are single characters.	Subject	[13]
18	To Localhost?	Metadata	[15]	43	Max ratio of uppercase letters to lowercase letters of each word.	Subject	[13]
19	To Google?	Metadata	[15]	44	Min of character diversity of each word.	Subject	[13]
20	To AOL?	Metadata	[15]	45	Max of ratio of uppercase letters to all characters of each word.	Subject	[13]
21	To Gov?	Metadata	[15]	46	Max of ratio of digit characters to all characters of each word.	Subject	[13]
22	To MIL?	Metadata	[15]	47	Max of ratio of non-alphanumeric characters to all characters of each word.	Subject	[13]
23	Count of "To" Email	Metadata	[13]	48	Max of the longest repeating character.	Subject	[13]
24	Replay to Google?	Metadata	[15]	49	Max of the character lengths of words.	Subject	[13]
25	Replay to Hotmail?	Metadata	[15]	-	-	-	-

Table 3.2: The Email Body Features

ID	Feature Details	Studies	ID	Feature Details	Studies
1	Count of Spam Words	[14] , [15] , [16]	31	Number of question marks	[15] , [16]
2	Count of Function Words	[14] , [16]	32	Number of multiple question marks	[15]
3	Count of HTML Anchor	[13] , [14] , [15] , [16]	33	Number of exclamation marks	[15] , [16]
4	Count of Unique HTML Anchor	[13] , [15]	34	Number of multiple exclamation marks	[15]
5	Count of HTML Not Anchor	[14]	35	Number of colons	[15] , [16]
6	Count of HTML Image	[16]	36	Number of ellipsis	[15] , [16]
7	Count of HTML All Tags	[14] , [16]	37	Total number of sentences	[14] , [15] , [16]
8	Count of Alpha-numeric Words	[13] , [14] , [16]	38	Total number of paragraphs	[15]
9	TF-ISF	[14]	39	Average number of sentences per paragraph	[15]
10	TF•ISF without stopwords	[14]	40	Average number of words pre paragraph	[15]
11	Count of duplicate words.	[16]	41	Average number of character per paragraph	[15]
12	Minimum word length	[16]	42	Average number of word per sentences	[15]
13	Count of lowercase letters	[16]	43	Number of sentence begin with upper case	[15]
14	Longest sequence of adjacent capital letters	[15] , [16]	44	Number of sentence begin with lower case	[15]
15	Count of lines	[15] , [16]	45	Character frequency "$"	[15] , [16]
16	Total number of digit character	[15]	46	Number of capitalized words.	[13]
17	Total number of white space	[15]	47	Number of words in all uppercase.	[13]
18	Total number of upper case character	[15] , [16]	48	Number of words that are digits.	[13]
19	Total number of characters	[13] , [15]	49	Number of words containing only letters.	[13]
20	Total number of tabs	[15]	50	Number of words that are single letters.	[13]
21	Total number of special characters	[15]	51	Number of words that are single digits.	[13]
22	Total number of alpha characters	[15]	52	Number of words that are single characters.	[13]
23	Total number of words	[15] , [16]	53	Max ratio of uppercase letters to lowercase letters of each word.	[13]
24	Average word length	[15] , [16]	54	Min of character diversity of each word.	[13]
25	Words longer than 6 characters	[15]	55	Max of ratio of uppercase letters to all characters of each word.	[13]
26	Total number of words (1 - 3 Characters)	[15]	56	Max of ratio of digit characters to all characters of each word.	[13]
27	Number of single quotes	[15] , [16]	57	Max of ratio of non-alphanumeric characters to all characters of each word.	[13]
28	Number of commas	[15] , [16]	58	Max of the longest repeating character.	[13]
29	Number of periods	[15] , [16]	59	Max of the character lengths of words.	[13] , [16]
30	Number of semi-colons	[15] , [16]	-	-	-

3.2.2 Feature extraction Tool

We provide an open source and scalable tool that can be used to remove the previously described functionality from any EML-formatted email corpus2 [21]. EML (or .EML) is one of the most popular email file extensions such as Outlook, Yahoo, and Gmail usage, is used on many email programs. the key part of the emails are "From", "To", "CC", "BCC", and "Subject" <> to a file, save every part of the text, and its HTML content as a production. The extraction and export tool then moves the relevant features from the email component and saves them to another CSV file. If any error occurred during the extraction process, an error file will be generated[10].

Tab pages are used to group and organize feature selection options in a feature. They have tabs that correspond to features that helps us to choose features, each of which has check boxes from which to pick the ones you want. To finish the extraction process, a folder and a browser is used to pick the corpus, which makes the extraction results visible as a place of interest, while an updated counts progressbar indicates progress [14]

3.2.3 Building a model for Dataset Training

Step 1: Retrieving spam and ham email messages:

Before we expand our model with additional sample data, we need a lot of ham and spam emails to work with[8]. We retrieve all the messages one by one and take it into the machine.

Step 2: Tokenization:

Tokenization is done into two steps. Each email is split into two parts, words and sentences are separately taken. Words are directly listed into a database which is called data dictionary. Using NLTK the sentences are converted to a vector and then stored to the data dictionary.

Step 3: Feature Extraction:

Feature extraction measures the frequency of the spam and ham emails. Generally, each of the word is labeled 0 if it comes from the ham email and labeled 1 if it is from spam email. In this method the vectors generated from sentences are also labeled 0 and 1 in accordance with spam and ham messages. In this way, model calculates the frequency of the tokens.

Step 4: Shuffle the data dictionary:

An alternative approach to token labelling is used that uses a 2D array to store all the tokens, then proceeds to randomize the placement of them in the dataset. These spam emails are included in the spam data dictionary without any other messages to contain them.

Step 5: Generate .pkl file:

After doing all of the above steps, the training process is complete, the dataset creates a .pkl file format.

Step 6: Seeding database: We have to seed the data dictionary with spam and ham messages as many as possible. The more data added, the more accuracy of a mail being ham or spam will be achieved.

Step 7: Test method (Input message): After training the model with enough datasets of spam and ham emails, we go to testing. We take a sample email body to test.

Step 8: Tokenization of test message: To maximize message expansion, each message is subdivides into two separate files.Some sentences can be expanded and

other sentences are converted into words.

Step 9: Frequency calculation: After tokenizing the test message, we calculate the frequency of each token and vectors in the token list.

Step 10: Retrieve Spam and Ham frequency: In this step, we will retrieve the spam and ham frequency from the training datasets of our model.

Step 11: Retrieve Spam and Ham count: We will also retrieve spam and ham count from the datasets.

Step 12: Calculate Spam probability: Calculate the spam probability of each token that are found in the database using the following formula: Spam probability= (spam frequency in database)/ (number of spam mail in database) If the probability is greater than 1, set it to 1.

Step 13: Calculate Ham probability: Calculate the ham probability of each token that are found in the database using the following formula: Ham probability= (Ham frequency in database)/ (number of ham mail in database) If the probability is greater than 1, set it to 1.

Step 14: Calculate Spamicity: Calculate the spamicity of each token that are found in the database using the output of step 13 using the following formula: Spamicity= (spam probability)/ (spam probability + ham probability)
Spamicity= (ham probability)/ (spam probability + ham probability).

Step 15: Total spam probability: Calculate the total spam probability using the output of step 14, It can be found as: Total probability= ((spamicity1....*spamicity2*.....spamicityn))+ ((1-spamicity1* (1-spamicity2*..... (1-spamicityn))

Step 16: Decision: If the output of the total probability for spam is greater than the total probability for ham, then the incoming test mail is considered as spam, otherwise it is a ham.

3.2.4 Flowchart of building a model

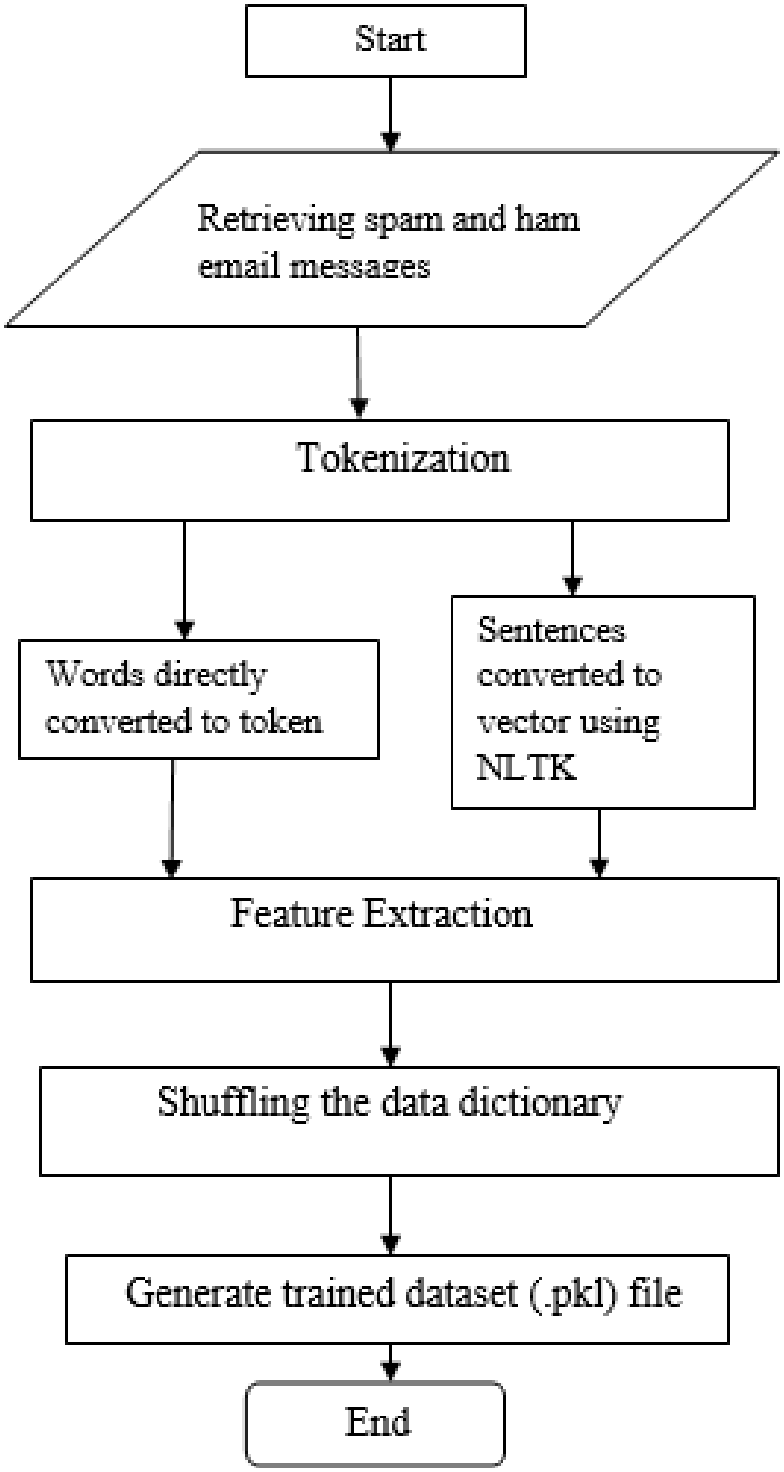

Figure 3.2: Flowchart representation for building a model.

3.2.5 Flowchart of spam classification techniques

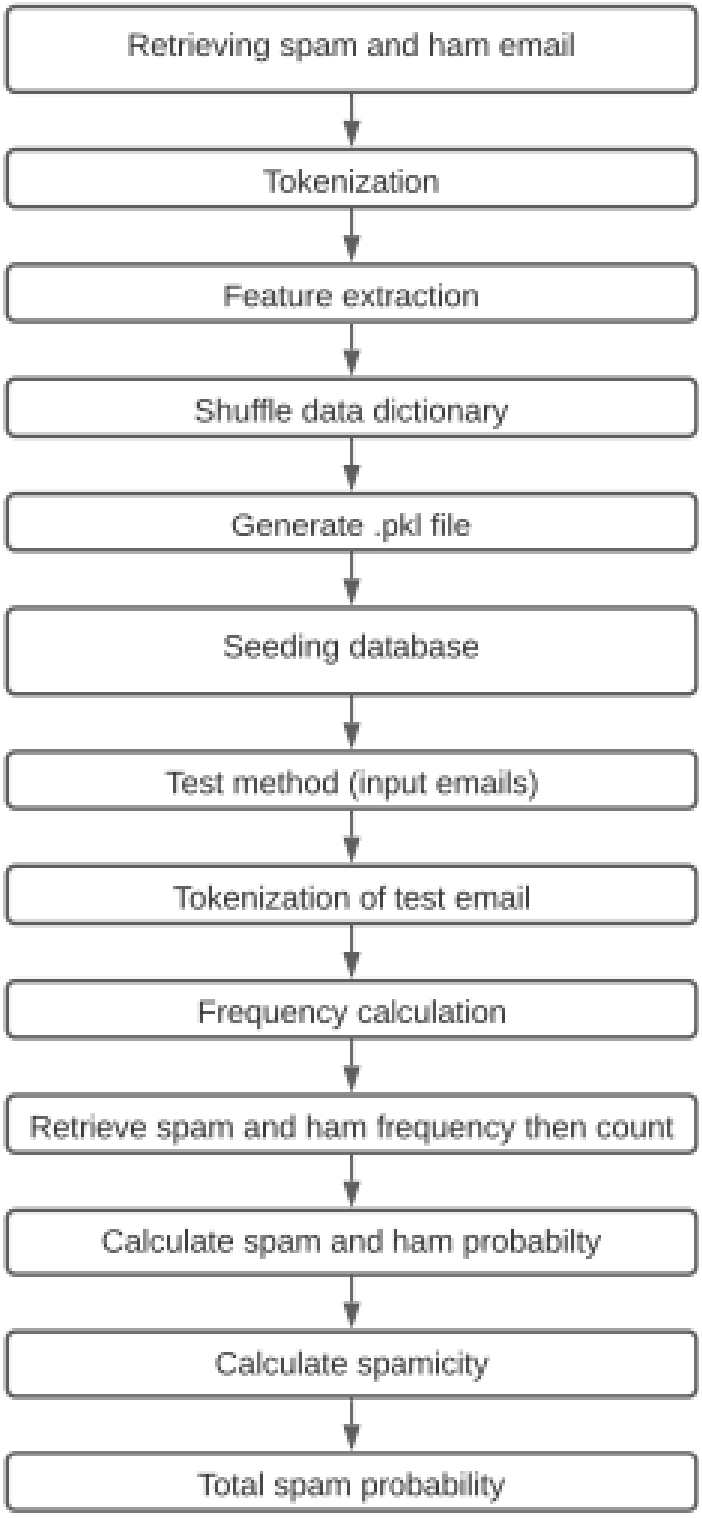

Figure 3.3: Flowchart representation for whole system

3.3 Detailed Explanation

3.3.1 Implementation

3.3.1.1 A sample spam message

Let us see what happened when the filter receive a mail and how the filter process the mail. As an example we can use a well known "enron datasets" spam message. The message is given below: From: BUMA SARO WIWA <bsarowiwa@incamail.com> To: imukaviva@incamail.com Subject: dobmeos with hgh my energy level has gone up ! stukm Introducing doctor – formulated hgh. human growth hormone - also called hgh. It is referred to in medical science as the master hormone . it is very plentiful when we are young, but near the age of twenty - one our bodies begin to produce less of it . by the time we are forty nearly everyone is deficient in high , and at eighty our production has normally diminished at least 90 - 95 advantages of high : - increased muscle strength. - loss in body fat. - increased bone density. - lower blood pressure. - quickens wound healing. - reduces cellulite. - improved vision. - wrinkle disappearance. - increased skin thickness texture. - increased energy levels. - improved sleep and emotional stability. - improved memory and mental alertness. - increased sexual potency. - resistance to common illness. - strengthened heart muscle. - controlled cholesterol. - controlled mood swings. - new hair growth and color restore. Subscribe today.

3.3.2 Processing steps

3.3.2.1 Tokenization

After receiving the mail the Bayesian spam filter first split the message into tokens. At the same time, sentences will be separated too by counting the comma (,), full stop (.), exclamatory sign (!), interrogative sign (?) etc punctuation marks.

Table 3.3: Token List (Words)

Dobmeos	with	hgh
My	energy	level
Has	gone	up
Stukm	human	growth
Hormone	also	called
Hgh	Is	referred
To	in	medical

Master	hormone	It
Is	very	plentiful
When	we	are
Young	but	near
The	Age	of
Twenty	one	our
Bodies	Begin	to
Produce	less	of
It	by	the
Time	we	are
Forty	nearly	everyone
Is	deficient	of
In	hgh	and
At	eighty	our
Production	Has	normally
Diminished	at	least
90-95%	advantages	of
Hgh	increased	muscle
Strength	loss	in
Body	fat	increased
Bone	density	lower
Blood	pressure	quickens
Wound	healing	reduces
Cellulite	improved	vision

Wrinkle	disappearance	increased
Skin	thickness	texture
Increased	energy	levels
Improved	sleep	and
Emotional	stability	improved
Memory	and	mental
Alertness	Increased	sexual
Potency	resistance	to
Common	illness	Strengthened
Heart	muscle	controlled
Cholesterol	controlled	mood
Swings	new	hair
Growth	and	color
Restore	Read	more
At	this	website
www	hmboe	com
Subscribe	today	

Table 3.4: Token Lists for sentences

dobmeos with hgh my energy level has gone up
Introducing doctor – formulated hgh
human growth hormone - also called hgh
It is referred to in medical science as the master hormone
it is very plentiful when we are young
but near the age of twenty - one our bodies begin to produce less of it
by the time we are forty nearly everyone is deficient in hgh

and at eighty our production has normally diminished at least 90 - 95 % . advantages of hgh
increased muscle strength
loss in body fat
increased bone density
lower blood pressure
quickens wound healing
reduces cellulite
improved vision
wrinkle disappearance
increased skin thickness texture
increased energy levels
improved sleep and emotional stability
improved memory and mental alertness
increased sexual potency
resistance to common illness
strengthened heart muscle
controlled cholesterol
controlled mood swings
improved memory and mental alertness
Read more at this website www
Subscribe today

3.3.2.2 Wordcloud

Wordcloud is a useful visualization tool for us to have a rough estimate of the words that has the highest frequency in the data that we have.

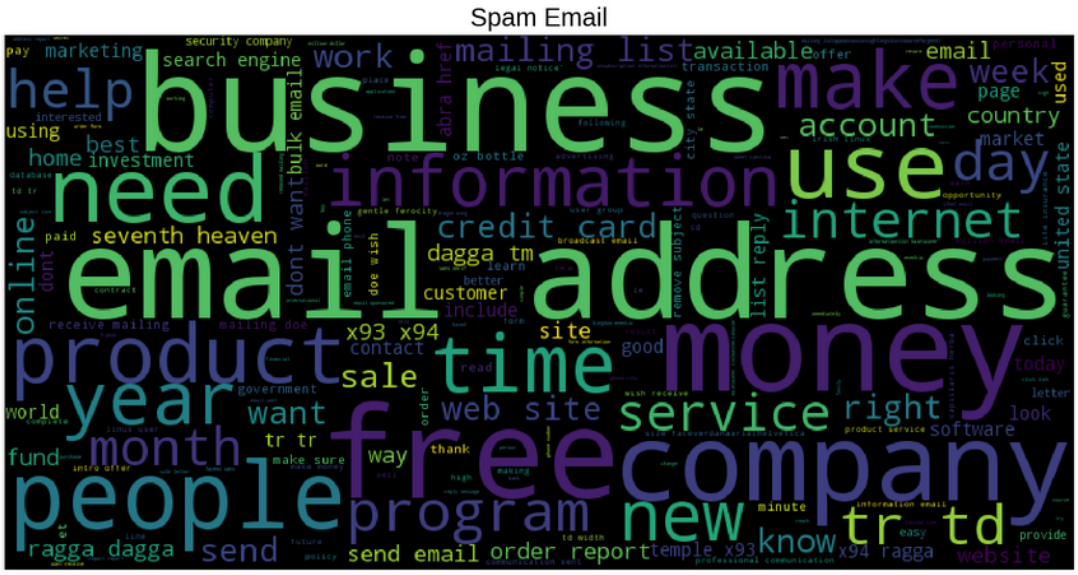

Figure 3.4: Word dictionary for spam emails

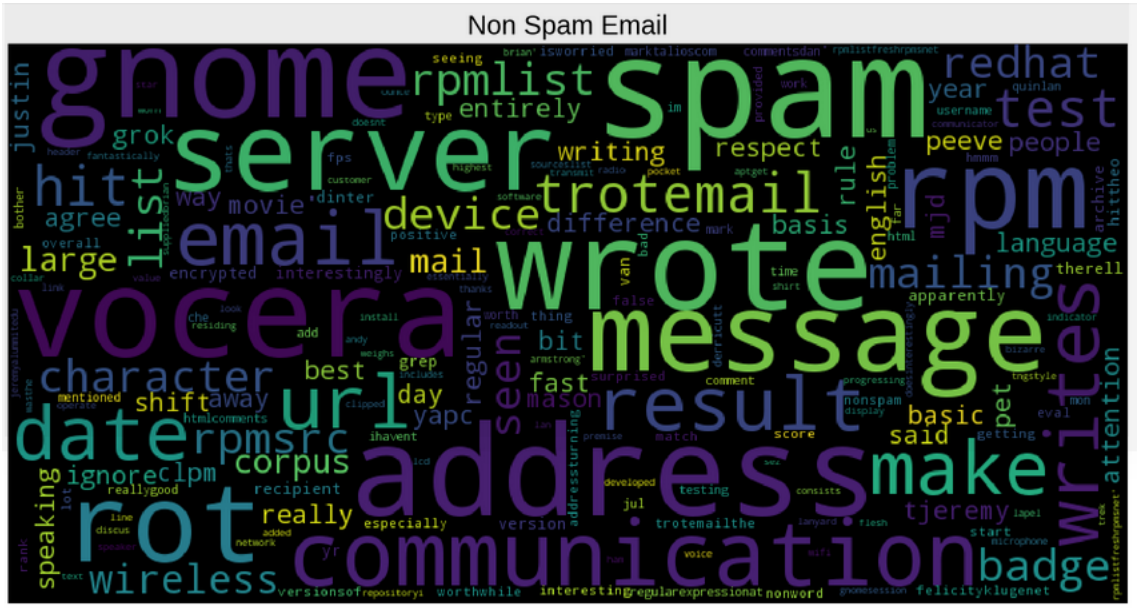

Figure 3.5: Word dictionary for ham emails

From this visualization, we are able to see that the email seems to be highly dense. Some of the terms they are using frequently in their text are pomposity words such as free, money, commodity, etc. It is possible that a sense of depth

and intelligence about spam will benefit us when it comes to our spam system's architecture. There is an important point to remember as we expand the term cloud that the cloud does not evaluate the importance of each of the terms. In order to properly present data, one must process, such as word stopwords and character encoding must be applied to the data before all other modifications have been made to it.

3.3.2.3 Frequency Calculation

By means of stop words list, we eliminate the unnecessary words that's not effective in spam detection. Then we calculate the frequency of single tokens. After that, with the help of NLTK, the sentences are converted into feature vectors and then their frequency is also calculated.

3.3.2.4 Retrieve Spam and Ham Frequency

Now we go to Naive Bayes and retrieve the data dictionary where the spam and ham frequency is stored.

3.3.2.5 Retrieve spam and ham count

After getting the spam and ham frequency, we now go through the spam and ham count, Here number of spam and ham messages from the data dictionary are found.

3.3.2.6 Calculate spam and ham probability

In this step the spam and ham probability of each tokens, and vectors generated from sentences are calculated using the formula discussed above. If either spam or ham probability is found greater than 1, then it is set to 1.

3.3.2.7 Spamicity Calculation

The spamicity of each token is calculated by using spam and ham probability using the formula discussed above.

3.3.2.8 Total spam probability

In our test mail, there are 122 tokens for words, and 27 vectors generated from sentences. It is really a lengthy calculation, so it would take some time in compiler to show result Using Naïve Bayes formula the total probability of the message to be a spam can be calculated as: Term 1: (0.99911) (0.654) (0.756) (0.889) (0.776) (0.445) (0.667) (0.334) (0.554) (0.233) (0.556)..... ..(0.912) (0.814) (0.823) Term 2: (1-0.99911) (1-0.654) (1-0.756) (1-0.889) (1-0.776) (1-0.445) (1-0.667) (1-0.334) (1-0.554) (1-0.233) (1-0.556)..... ..(1-0.912) (1-0.814) (1-0.823) Total spam probability = Term1/(Term1+Term2)=0.74

In similar way, we have also calculated total ham probability which is 0.33. These last values (which show the final probability of the message being spam) give us a good idea of the message's spam likelihood. We have already defined our measurement value. We're working to identify relevant information for this approach If the likelihood is higher than the other, then the message will be classified as spam, while if it is less, then we will regard it as something else. . Hence the designed Naïve Bayesian filter considers it as a **spam** message because here ham probability is less than spam probabillity.

3.3.3 Implement this output as a web application

During the development of the machine learning models, we mostly focused on generating a numerical prediction based on the sample test dataset. But to make the real-world application it is important to deploy these models on the server to use in the form of applications. But in my opinion, both the model generating and deployment part is equally important. As we have discussed the proposed solution in the above section now, we will implement this proposed solution as a web application using Python's Micro Flask Framework for web development which takes new email message as input and predicts whether the given input is a spam or a ham as an output.

This system consists of two major parts. The first is to train the classifier with a dataset which consists of spam and ham emails and generation of the classification model[22]. The second part is to deploy this model as a web service on a

server. In the first part, we generate the classification model which defines the rules or criteria on which the classification of the email takes place. The process starts with importing the dataset in the system then the process of pre-processing on the dataset set takes place which removes various factors which degrades the quality of dataset like removal of punctuation, whitespace and converting upper case words into lower case words and replacing the email addresses, URLs, phone numbers, other numbers with the regular expressions

Then the features are extracted from the dataset and after this for the training purpose, the dataset is split into two parts i.e. training part and test part. Now with the help of training dataset, we train our model and after the required amount of training, we can save our model in the .pkl file format so we did not have to train the model again and again. This process of generating a classification model is known as —persist model in a standard format, that is, models are persisted in a certain format specific to the language in development. The whole process in this first part can be done offline[23].

The second is using our classifier model from the first part as a web application for this we have use python's Flask framework for the development of the web application. For this first, we install the Flask in our repository where our previous model is stored and then start building the webpage which takes email as an input and which can be named as —index.html and another webpage which displays the output and can be named as —result.html. These two webpages are developed with the help of HTML and CSS in the Flask framework. It is a good practice to begin the development in the virtual environment during the development of the web application which uses various libraries which can collide with the working of different application on our system.

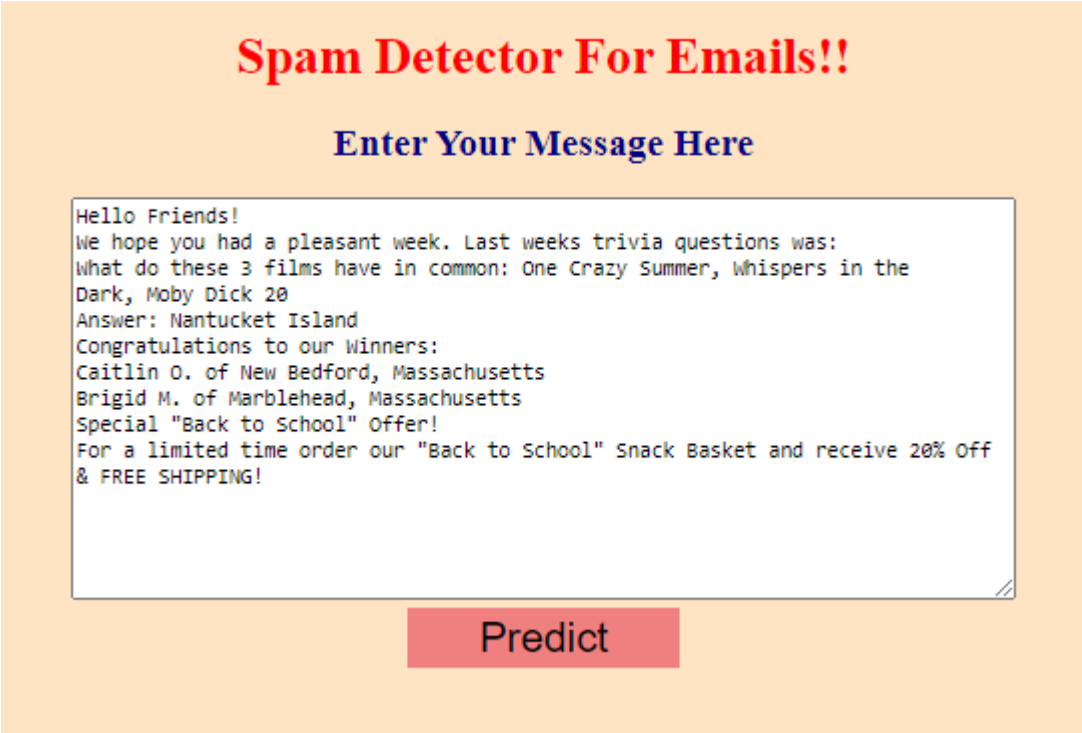

Figure 3.6: Sample input for ham message

Figure 3.7: Sample output for ham message

Figure 3.8: Sample input for spam message

Figure 3.9: Sample output for spam message

3.4 Conclusion

The whole projects methodology is described in this chapter. What kinds of algorithm and techniques we used, why used and how we used this are also described here. We can easily understand the basic concepts and design frameworks through this chapter.

Chapter 4

Results and Discussions

4.1 Introduction

In the previous chapter, a detailed explanation of the proposed framework for Content Based Spam Classification as a Web Application was given. This chapter examines the performance of the proposed framework. This framework was implemented in Jupyter Notebook environment with Intel Core i5 processor and 8GB RAM. For this thesis work, 'Spambase Dataset' [14] dataset was used. This dataset was collected from UCI Machine Learning Repository, which is an open source data repository[24].

4.2 Dataset Description

For successfully completing a project using machine learning algorithm we have to choose one thing most importantly which is known as dataset.As our email spam filtering project is implemented using machine learning algorithm we selected our dataset from UCI Machine Learning Repository which contains more than 5500 labelled emails. The detailed information about our dataset is described below:
Data Set Characteristics: Multivariate
Attribute Characteristics: Integer, Real
Number of Instances: 5757
Number of Attributes: 67
Missing Values? Yes
Associated Tasks: Classification
Area: Computer
Date Donated : 2009-07-01

Number of Web Hits: 576887

As it is a machine learning problem, we classified our dataset into two categories: training dataset and testing dataset. We split our 5757 email datas into 80% for training data and 20% for testing data.

Table 4.1: Size of the training and testing dataset

	No of samples	**Percentage**
Training data	**4,605**	**80%**
Testing data	**1,152**	**20%**

4.3 Impact Analysis

4.3.1 Experimental Result

We also examined the efficiency of the spam filter by comparing it to the spam-massin corpus of the Apache Foundation. As an example, when the naive Bayesian filter runs on an email sample of text, the output of the Naive Bayes classifier is listed in the Table. The corresponding graph of the output is given below:

Table 4.2: Comparative analysis of performance

Steps	Total no of email (input)	Number of Spam (input)	Number of Ham (input)	Number of Spam (output)	Number of Ham (output)	False Positive (Output)	False Negative (Output)	Accuracy (%)
initially	150	65	85	-	-	-	-	-
1	40	15	25	15	24	1	0	99.4
2	75	30	45	29	45	1	0	99.29
3	120	45	75	44	74	1	1	99
4	150	45	75	44	73	1	2	98.9

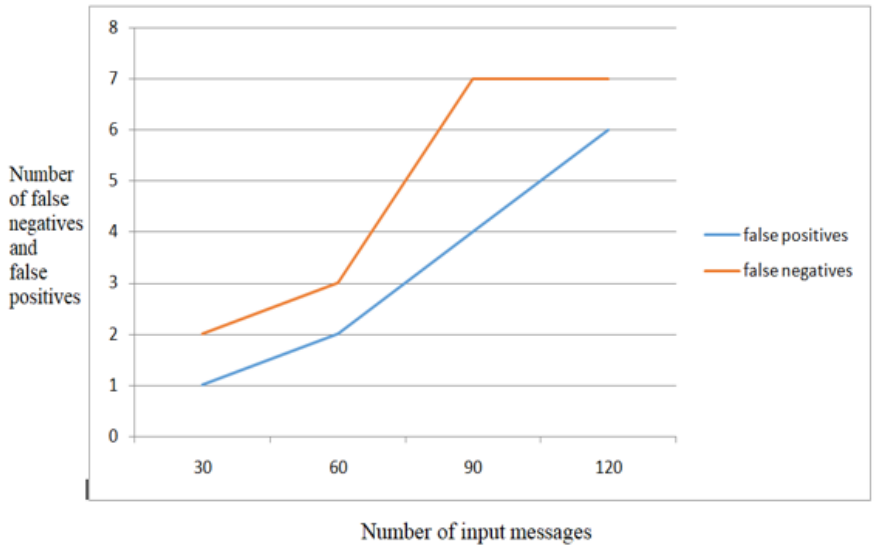

Figure 4.2: Performance curve of Naïve Bayesian Filters

We considered total 150 emails among them 65 spam and 85 ham/non spam. The input emails are provided in four steps. At step-1 and step-2, step-3 and step-4 total 40, 75, 120, 150 emails are used respectively.

During the experiment for step-1 and step-2 we used one spam and one ham alternatively. This is done so that there is a balance between the spam and ham database. For the next two steps no such order was maintained.

In figure 4.2 we have shown the performance curve of spam filter based on the number of false positives and false negatives with the total number of input messages during the experiment. The blue curve shows the false positives and the orange curve represents the false negatives. The less the number of false positive and false negative the greater the performance and vice versa.

Now I want to check my system whether it shows right output or not.

Ham: This looks like a normal email reply to another person, which is not difficult to classified as a ham:

Sample input: This may be a little cumbersome, but may be useful for our purposes If you have an internal zip drive (and BIOS accepts zipping as a Floppy disk drives, that is), you can use a bootable zip drive with all your DOS software.

Sample Output: Great! This is not a spam message.

Hard Ham (Ham email that is trickier): Hard Ham is indeed more difficult to differentiate from the spam data, as they contain some key words such as limited time order, Special “Back to School” Offer, this make it very suspicious !
Sample input: So welcome all of my mates! we would like to thank you for having a good week. These questions were all from last week's quiz contest: To what extent do these 3 films share is the same denominator Many secrets will be revealed to you that year. One confusing, puzzling, utterly fascinating summery year, one "crazy" summer, Moby-Dick That expands for two-twenty-five more has the island of Nantucket Congratulations to our Winners, everyone!
Dr. Caitlin O'Hara of New Bedford, MA and Dr. Maribeth M. Bragg of Marblehead,We have an Exclusive Back to School promotion for you! Extendangered and Get S&0.20% off AND FREEMACIEMN at 20-on!!"back to school" , "school snack EXTEND0 20 in bazaar" and recoup free Get up to 20% off retail price!"
Sample Output : Great! This is not a spam message.

Spam: One of the spam training data does look like one of those spam advertisement email in our junk folder:
Sample input: Congrats! You've won 1.0111 btc in our spring giveaway from exchange!

1. Open the exchange
2. Go the settings tab
3. Write your Promo: pSygpl1laHRb
4. Have fun with bitcoin tech!

Sincerely yours, Space coin Manager Martin Ryan
Sample Output: Be Careful!! this is a spam message.

4.3.2 Social and Environmental Impact

Some social impact for our project is given below:

1. Spam filter can prevent social chaos by filtering false and fake news that can be shared on social platforms for destructive purposes.
2. Women and children can stay protected from many kinds of cyber bullying.

3. It would be difficult for Hackers to spread virus through spam messages.

Some environmental impact of spam filtering is given below:

- 135 kWh of power a year is saved by spam filtering. and it is about as many cars as excluding 13 million vehicles from the road.
- If every inbox is equipped with a state-the-of-the-the-the-the-the-art filter, companies and individuals could use 75 million (and about 25 trillion) less kilowatt (or about the same amount of gasoline, as 2.3 million vehicles per year.

4.3.3 Ethical Impact

Sending spam that means sending same advertisement or message or email to millions of people. People waste their time deleting spams as spam filter is able to identify it though.Using spam filtering we can overcome this problem which can save peoples money and time.

4.4 Evaluation of Framework

Naïve Bayesian spam filters are more effective to detect spam than other types of filter such as signature based, rule based or list based filters. One major reason would be that they have the mechanism of continuously incorporating and adjusting the spam/ham probability value during the processing new email. While other filters cannot increase their accuracy because they are confined to some fixed rule and properties.

A primary explanation for the outstanding success of Naïve Bayesian spam filters is because they are open to the general public and that they were tailored to particular consumers, who use them to create more tailored filter settings. Some kinds of filters can't accommodate features such as text-based and list-based filtering are used on the server side, but other features like signatures are used in small-scale (mostly on the client) filters.The comparative analysis is shown in below table.

Table 4.3: Comparative analysis of Filtering Systems

Filter type	Fast	Reliable	Dynamic	Application Area
Signature-based	Yes	Yes	No	Server
Rule-based	Yes	Sometimes	No	Both server, client
Learning systems	No	Yes	Yes	Both server, client
List-based	Yes	Sometimes	No	Server

The learning method is slower due to the increased in use of a large numbers of tokens and a database. Retrieving, inserting and updating the archive is time-consuming each time. At other organizations, though, a certain amount of business is conducted using a database-driven structure, than try to expand on or interpret the content. Both, it interacts with the email content of the message instead of either expanding on the sender or the message delivery network This causes the spammer to be unable to do one of the two things he wants to do, getting around the filter by sending spam from a different address or from a new location or network. These techniques expand the effectiveness of learning processes, whereas rule-based and list-based systems are less effective. Learning system is dynamic where others are static because, other systems need to be tuned by specialists after a period of time while the tuning process if learning system is done automatically after receiving the email. Learning system can be implemented in both client side and server side of the mailing system with suitable configuration where list based and signature based systems are implemented only in server side. From all the discussions and comparisons, we can easily understand that the next generation spam filtering will be absolutely learning based. It is the most effective system till now in spam filtering technique.

4.5 Evaluation of Performance

As our main task is to classify between ham and spam emails.So, at first we showed top 10 ham emails after cleaning that means after removing stopwords.This is shown using a bar chart below:

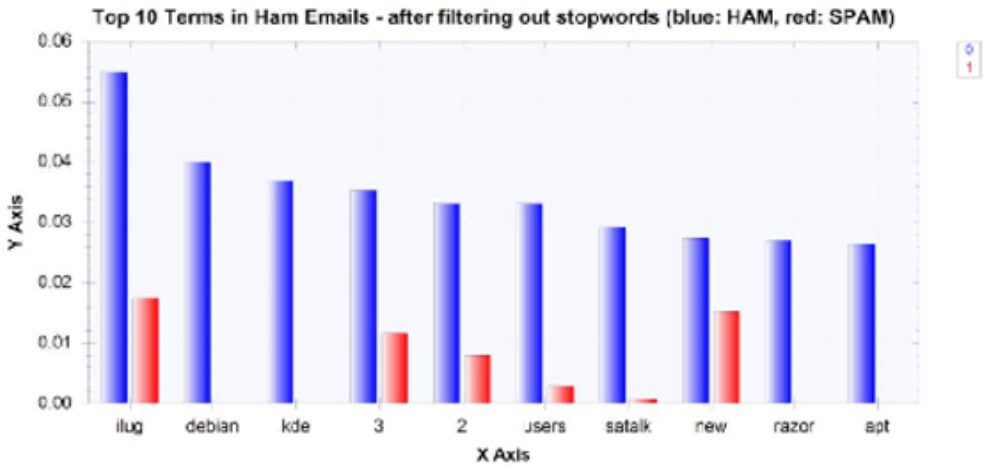

Figure 4.1: Top 10 ham emails bar chart

Now similarly,we showed top 10 spam emails after cleaning means after filtering and removing stop words.This is also shown using a bar chart below:

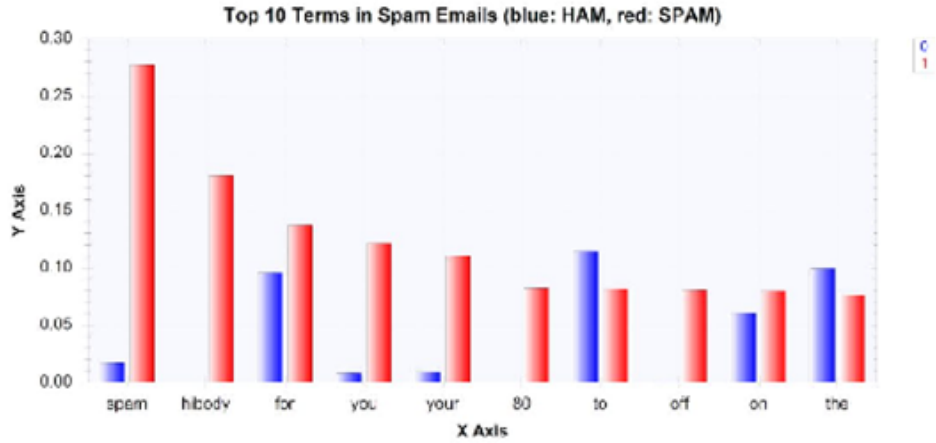

Figure 4.2: Top 10 spam emails bar chart

From the confusion matrix, a set of value of true positive (tp), true negative (tn), false positive (fp) and false negative (fn) is derived.The predicted classes

are represented in the columns of the matrix, whereas the actual classes are in the rows of the matrix. We then have four cases:

True positives (TP): The cases for which the classifier predicted 'spam' and the emails were actually spam.

True negatives (TN): The cases for which the classifier predicted 'not spam' and the emails were actually real.

False positives (FP): The cases for which the classifier predicted 'spam' but the emails were actually real.

False negatives (FN): The cases for which the classifier predicted 'not spam' but the emails were actually spam.

So the confusion matrix for our project is shown below:

		Predicted	
		Non-spam(ham)	Spam
Actual	Non-spam(ham)	True Positive=1455	False Negative=0
	Spam	False Positive =10	True Negative=207

Table 4.4: Confusion Matrix

From the confusion matrix, we see that the Naive Bayes classifier got the following results:

From the 1455 actual instances of 'ham' (not spam), it predicted correctly 1455 of them; From the 217 actual instances of spam, it predicted correctly 207 of them.

In the multi-class problem, there are two types of measure, namely, micro and macro. F1-score is the harmonic mean of precision and recall. F1-score depicts the overall system performance.The support of the class in the given dataset indicates how many examples of that class appear in the results. Structural imbalances

in the training data suggest an uneven support for the classifier, which may or may not reflect the ratings, depending on the structure, but indicates the need for sub-based sampling or rebalancing.

The second phase of implementation is measuring the accuracy of the proposed framework. Based on the parameters and training data, a model is generated by the framework. Then testing data is supplied to the model. The model returns predicted label for the samples of testing data set. Using the values of the parameter, we can determine precision, recall and F1-score for the proposed system.

The **precision** for this model is calculated as:

Precision = True Positives / (True Positives + False Positives)

Precision = 1455 / (1455 + 10)

Precision = 1455 / 1465

Precision = 0.99

Similarly, the **recall** value for this model is calculated as:

Recall = True Positives / (True Positives + False Negatives)

Recall = 1455 / (1455 + 0)

Recall = 1455 / 1455

Recall = 1.00

The **f-score** for this model is calculated as:

F-score = 2 * precision * recall / (precision + recall)

F-score = 2*0.99*1 / (0.99 + 1.00)

F-score = 1.98 / 1.99

F-score = 0.99

The **accuracy** for this model is calculated as:

Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Negatives + False Positives)

Accuracy = 1455+207 / (1455 + 207+ 0+ 10)

Accuracy = 1662 / 1672

Accuracy = 0.994 = 99%

Table 4.5: Classification Report

	Precision	Recall	F1-score	Support
Ham	0.99	1.00	0.99	1459
Spam	1.00	0.92	0.96	213
Accuracy			0.99	1672
Macro average	0.98	0.88	0.92	1672
Weighted average	0.97	0.97	0.97	1672

Accuracy Measurement for different algorithms : We chose to use Naive Bayes Classification model for our project. The reason for choosing this model mainly is its greater accuracy performance. Comparing with other models such as Gradient Boosting, Logistic Regression, Support Vector Machine, SGD, Random Forest we show the accuracy score in below table:

Table 4.6: Comparison between different algorithms and accuracy measurement

Model	Accuracy
Naive Bayes	0.977991
Gradient Boosting	0.962332
Logistic Regression	0.959641
SVM	0.946188
SGD	0.937220
Random Forest	0.927354

4.6 Conclusion

In this chapter, we have shown the detailed explanation with result and evaluate the performance of framework and explained the reason for choosing our algorithm which is used to implement our project.This chapter is the most important part of the whole project report.

Chapter 5

Conclusion

5.1 Conclusion

In this report we have discussed entirely about the Naïve Bayesian spam filter using different types of Bayesian filter. We have also discussed about various types of spam filtering techniques and compare them with our filter. The Bayesian filter is completely based on statistics. Accuracy is shown by some real-time experimentation during the experiment process and we've raised the standard to 98%.From the conducted project experiment, we van say that the use of Naive Bayes classifier is the best for email classification as it has high accuracy and precision score. Due to the easy in implementation of this classifier, it is suitable to use in creating the web application for classification of emails. As well as the speed of communications, the quality of the database also depends on the order in which they are sent.In the other hand, the drawback of a singular filters is that they can all be distinct. When using filters have varying probability values, any of the ones that are being applied could be disregarded, because either one or a few are employed at the same time. Additionally, it is more important to examine the information than any other tactics in order to take down spammers, since they can only communicate with the intended recipients based on the evidence they have to gain their confidence. They are creating spam message with shadow of ham so that it becomes really hard to detect by spam filter. To effectively build a spam filter, one should consider content as well as the huge amount of data.

5.2 Future Work

Even the most advanced spam filters cannot ensure 100% filtering effectiveness. Thus, our enhanced spam filter does not preclude the presence of vulnerability, but rather incorporates different aspects of the filter. With this, therefore, there are further developments which are possible. The opportunities that are available could include:

- Spammers now also used image based spam to confuse the users. Image based spam detection can be developed to detect image based spam.
- Now-a-days spam in bangla language is also seen With language processing technique, it is possible to detect spam emails of bangla language.
- Adding various modules in the current filter to increase such as optical image recognization for image classification and this application can also be implemented in the form of a mobile application for further use[25].

References

[1] S. R. Gomes, S. G. Saroar, M. Mosfaiul, A. Telot, B. N. Khan, A. Chakrabarty and M. Mostakim, 'A comparative approach to email classification using naive bayes classifier and hidden markov model,' in *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, 2017, pp. 482–487. DOI: 10.1109/ICAEE.2017.8255404 (cit. on pp. 1, 5, 8).

[2] S. Wang, X. Zhang, Y. Cheng, F. Jiang, W. Yu and J. Peng, 'A fast content-based spam filtering algorithm with fuzzy-svm and k-means,' in *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2018, pp. 301–307. DOI: 10.1109/BigComp.2018.00051 (cit. on pp. 4, 11, 22).

[3] T. Mehrotra, G. K. Rajput, M. Verma, B. Lakhani and N. Singh, 'Email spam filtering technique from various perspectives using machine learning algorithms,' in *Data Driven Approach Towards Disruptive Technologies: Proceedings of MIDAS 2020*, Springer Singapore, 2021, pp. 423–432 (cit. on p. 4).

[4] A. Adetunji, J. Oguntoye, O. Fenwa and N. Akande, 'Web document classification using naıve bayes,' *Journal of Advances in Mathematics and Computer Science*, pp. 1–11, 2018 (cit. on p. 4).

[5] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed and M. A. Al-Garadi, 'Email classification research trends: Review and open issues,' *IEEE Access*, vol. 5, pp. 9044–9064, 2017 (cit. on p. 5).

[6] W. You, K. Qian, D. Lo, P. Bhattacharya, M. Guo and Y. Qian, 'Web service-enabled spam filtering with naïve bayes classification,' in *2015 IEEE First International Conference on Big Data Computing Service and Applications*, 2015, pp. 99–104. DOI: 10.1109/BigDataService.2015.19 (cit. on pp. 5, 20).

[7] W. Peng, L. Huang, J. Jia and E. Ingram, 'Enhancing the naive bayes spam filter through intelligent text modification detection,' in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 849–854. DOI: 10.1109/TrustCom/BigDataSE.2018.00122 (cit. on pp. 5, 13, 14).

[8] A. Zamir, H. U. Khan, W. Mehmood, T. Iqbal and A. U. Akram, 'A feature-centric spam email detection model using diverse supervised machine learning algorithms,' *The Electronic Library*, 2020 (cit. on pp. 6, 8, 27).

[9] R. K. Kumar, G. Poonkuzhali and P. Sudhakar, 'Comparative study on email spam classifier using data mining techniques,' in *Proceedings of the*

International MultiConference of Engineers and Computer Scientists, vol. 1, 2012, pp. 14–16 (cit. on p. 8).

[10] A. Zaid, J. Alqatawna and A. Huneiti, 'A proposed model for malicious spam detection in email systems of educational institutes,' in *2016 Cybersecurity and Cyberforensics Conference (CCC)*, IEEE, 2016, pp. 60–64 (cit. on pp. 8, 26).

[11] A. Rodan, H. Faris, J. Alqatawna *et al.*, 'Optimizing feedforward neural networks using biogeography based optimization for e-mail spam identification,' *International Journal of Communications, Network and System Sciences*, vol. 9, no. 01, p. 19, 2016 (cit. on p. 10).

[12] H. Faris, I. Aljarah and B. Al-Shboul, 'A hybrid approach based on particle swarm optimization and random forests for e-mail spam filtering,' in *International Conference on Computational Collective Intelligence*, Springer, 2016, pp. 498–508 (cit. on pp. 10, 11, 20).

[13] D. Gudkova, M. Vergelis, N. Demidova and T. Shcherbakova, 'Spam and phishing in q2 2016,' *Kaspersky Lab*, vol. 18, 2016 (cit. on pp. 10, 23).

[14] W. Hijawi, H. Faris, J. Alqatawna, A. M. Al-Zoubi and I. Aljarah, 'Improving email spam detection using content based feature engineering approach,' in *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, 2017, pp. 1–6. DOI: 10.1109/AEECT.2017.8257764 (cit. on pp. 11, 16, 26, 41).

[15] J. Kolluri and S. Razia, 'Text classification using naıve bayes classifier,' *Materials Today: Proceedings*, 2020 (cit. on p. 14).

[16] B. Al-Shboul, H. Hakh, H. Faris, I. Aljarah and H. Alsawalqah, 'Voting-based classification for e-mail spam detection.,' *Journal of ICT Research & Applications*, vol. 10, no. 1, 2016 (cit. on p. 15).

[17] S. A. Khamis, C. F. M. Foozy, M. F. Ab Aziz and N. Rahim, 'Header based email spam detection framework using support vector machine (svm) technique,' in *International conference on soft computing and data mining*, Springer, 2020, pp. 57–65 (cit. on p. 20).

[18] H. Bhuiyan, A. Ashiquzzaman, T. I. Juthi, S. Biswas and J. Ara, 'A survey of existing e-mail spam filtering methods considering machine learning techniques,' *Global Journal of Computer Science and Technology*, 2018 (cit. on p. 20).

[19] H. Zhang, N. Cheng, Y. Zhang and Z. Li, 'Label flipping attacks against naive bayes on spam filtering systems,' *Applied Intelligence*, pp. 1–12, 2021 (cit. on p. 21).

[20] R. Nayak, S. A. Jiwani and B. Rajitha, 'Spam email detection using machine learning algorithm,' *Materials Today: Proceedings*, 2021 (cit. on pp. 23, 24).

[21] J. Alqatawna, A. Hadi, M. Al-Zwairi and M. Khader, 'A preliminary analysis of drive-by email attacks in educational institutes,' in *2016 Cybersecurity and Cyberforensics Conference (CCC)*, IEEE, 2016, pp. 65–69 (cit. on p. 26).

[22] A. A. Alurkar, S. B. Ranade, S. V. Joshi, S. S. Ranade, P. A. Sonewar, P. N. Mahalle and A. V. Deshpande, 'A proposed data science approach for email spam classification using machine learning techniques,' in *2017 Internet of Things Business Models, Users, and Networks*, IEEE, 2017, pp. 1–5 (cit. on p. 36).

[23] D. Gaurav, S. M. Tiwari, A. Goyal, N. Gandhi and A. Abraham, 'Machine intelligence-based algorithms for spam filtering on document labeling,' *Soft Computing*, vol. 24, no. 13, pp. 9625–9638, 2020 (cit. on p. 37).

[24] N. F. Rusland, N. Wahid, S. Kasim and H. Hafit, 'Analysis of naıve bayes algorithm for email spam filtering across multiple datasets,' in *IOP conference series: materials science and engineering*, IOP Publishing, vol. 226, 2017, p. 012 091 (cit. on p. 41).

[25] T. Verma and N. S. Gill, 'Email spams via text mining using machine learning techniques,' *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 4, pp. 2535–2539, 2020 (cit. on p. 53).