SPAM MAIL DETECTION

AN INTERNSHIP PROJECT REPORT

`

By
Jainu Varsha Priya

(Regd. No. 20JG1A0529)

Under The Esteemed Guidance of

Mrs. R. Archana

Assistant Professor

CSE Department

Department of Computer Science and Engineering

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN

[Approved by AICTE NEW DELHI, Affiliated to JNTUK Kakinada]

[Accredited by National Board of Accreditation (**NBA**) for B.Tech. CSE, ECE & IT – Valid from 2019-22 and 2022-25]

[Accredited by National Assessment and Accreditation Council (**NAAC**) – Valid from 2022-27]

Kommadi, Madhurawada, Visakhapatnam–530048

2023–2024

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the internship project report titled **"SPAM MAIL DETECTION"** is a bonafide work of following IV B.Tech. student in the Department of Computer Science a nd Engineering, Gayatri Vidya Parishad College of Engineering for Women affiliated to JNT University, Kakinada during the academic year 2023-2024 Semester-1.

Jainu Varsha Priya (Regd. No. 20JG1A0529)

Mrs. R. Archana
Assistant Professor
(Internal Guide)

Dr. P. V. S. Lakshmi Jagadamba
Professor
(Head of the department)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We would like to extend our heartfelt gratitude and a sincere thanks to **Prof. PS Avadhani**, Professor, Department of CSE, GVPCEW, and former Director IIIT Agartala, former Principal AU College of Engineering (A), Andhra University Visakhapatnam, Senior Professor Department of CSE College of Engineering (A), Andhra University, for his valuable guidance and providing necessary help whenever needed.

We feel elated to extend our sincere gratitude to **Mrs. R. Archana,** Assistant Professor for encouragement all the way during analysis of the project. Her annotations, insinuations and criticisms are the key behind the successful completion of the thesis and for providing us all the required facilities.

We express our deep sense of gratitude and thanks to **Dr. P. V. S. Lakshmi Jagadamba**, Professor and Head of the Department of Computer Science and Engineering for her guidance and for expressing her valuable and grateful opinions in the project for its development and for providing lab sessions and extra hours to complete the project.

We would like to take this opportunity to express our profound sense of gratitude to Vice Principal, **Dr. G. Sudheer** for allowing us to utilize the college resources thereby facilitating the successful completion of our project. We are also thankful to both teaching and non-teaching faculty of the Department of Computer Science and Engineering for giving valuable suggestions for our project.

We would like to take the opportunity to express our profound sense of gratitude to the revered Principal, **Dr. R. K. Goswami** for all the help and support towards the successful completion of our project.

TABLE OF CONTENTS

ABSTRACT

SMS (Short Message Service) is still the primary choice as a communication medium even though nowadays mobile phone is growing with a variety of communication media messenger applications. However, nowadays along with the SMS tariff reduction leads to the increase of SMS spam, as used by some people as an alternative to advertise and fraud. Therefore, it becomes an important issue as it can bug and harm the users and one of its solutions is with automatic SMS spam filtering.

One of most challenging in SMS spam filtering is its accuracy. In this research we proposed to enhance SMS spam filtering performance by combining two of data mining task association and classification. FP-growth in association is utilized for mining frequent pattern on SMS and Naive Bayes Classifier is used to classify whether SMS is spam or ham. Training data was using SMS spam collection from previous research. The result of using collaboration of Naive Bayes and FP-Growth performs the highest average accuracy of 90%. FP-Growth for dataset SMS Spam Collection and improves the precision score; thus, the classification result is more accurate.

1. INTRODUCTION

1.1 Problem Statement:

Spam Detection system is proposed which will classify the data into spam and ham. A typical data can be classified by filtering its content. The process of spam detection is based on the assumption that the content of the spam is different than the legitimate or ham.
Spam is also referred to as junk and is unsolicited messages sent in bulk by spamming. Spammers collect email addresses from chat rooms, websites, customer lists, newsgroups, and viruses that harvest users' address books.
On top of mocking-up delivery and storing the results you also need to have human analysis of the accuracy of filtering (flagging as spam, deleting, filing to a "Junk E-mail" folder, etc.). It probably goes w/o saying, but a human need to verify the results of filtering since the whole point of this exercise is to test the computer's ability to filter results. (I know, that went w/o saying... but I can just hear somebody saying "But wait! We'll write a script to test the accuracy...")

The filtering analysis is problematic foremost, to my mind, as a matter of scale. For any large group of end-users, filtering analysis is going to be beyond the abilities of an individual or small group of testers (or, more likely, they'll just "spot check" and hope for the best). Moreover, the tester may have a different idea of what should be considered "spam" than the real receiving user (who really *does* want to receive those stupid promotional emails from their car rental company, or the "vocabulary word of the day"). Identifying obviously failed messages is probably pretty easy (pornographic emails in the "Inbox", messages from known Customers in the "Junk E-mail" folder, etc.), but there's certainly the potential for subtlety.
I don't think there's a 100% substitute for reality-- you're just going to have to get close and hope that your testing setup reflects reality closely enough to give you an accurate assessment of the filter's performance.

1.2 Objectives:

The objective of a spam mail detection project is to detect and score word quickly and accurately. The project's system will use machine learning algorithms to extract spam emails. Key objectives include:

- Improving user productivity
- Increasing system performance
- Reducing security risk
- Protecting user security
- Improving user experience

2. SYSTEM SPECIFICATIONS

Hardware requirements:

- System : Intel CORE i3
- Hard Disk : 40 GB.

Software requirements:

- Operating system : Windows 7/8/10
- Coding Language : Python
- IDE : PyCharm, Python IDLE 2.7/3.6
- Database : CSV File, TSV File.

3.SYSTEM ANALYSIS

3.1 Existing System:

Now a day's true caller is that existing system which can block these senders message whose messages are annoying you but we have a control over the sender but not ones the type of messages. So we need such technology/system which can block the particular kind of messages.

In the existing system attackers used shortened malicious URL's that redirect users to external attack server to cope with malicious tweets several twitter spam detection have been proposed these scheme can be classified into account feature-based, relation feature based , message feature based schemes.

Disadvantages:

Suppose it a user don't want any promotional message and it he knows which all users can send him these kinds of messages then he/she can block these senders. But if these blocked users any informational message to the user then user will not be able to receive the message. Great likelihood of unresponsive recipients. Waste of time for sales staff. Waste of money for marketing department. Damaged company reputation. Drop in open and click-through rates for marketing campaigns. The most effective method of email marketing is opt-in email. This is just what it sounds like. It is commercial email sent to people who have specifically indicated a willingness to receive it by opting in somehow. It is specifically targeted to a willing audience, making it the Cadillac of email marketing strategies.

3.2 Proposed System:

We are using Machine Learning algorithm (Naïve Bayes Algorithm) to eradicate such problem. In this algorithm model will train the machine by its 70% and 30% of dataset. Through this 70% data our machine will be trained enough to decide which is the SPAM message or which is the HAM message. We have tried to create a system which not only identifies email as a spam or non-spam (also known as Ham) but also which can find a concept present in the e-mail along with the category it falls the most into. We have used different lexical, semantic and syntactic features in order to create the system. We have created a database of emails.

The proposed algorithm is consisting of six main steps.

1. Collect the test emails.
2. Remove the following from emails:

a. Html tags
b. Symbols
c. Unimportant word and remaining are converted in dictionary file

3. Now here using the less algorithm to find the repeated word and list out the total number of word count from the dictionary file.
4. Total number of word count is checked which word are spam and which are ham/non-spam.
5. Collect spam and ham mail are put in the unigram weighted formula to find out whether mails are spam or ham.
6. Total number of words count we process it in the word net dictionary which gives the definition of that word and defined the category.

3.2.1 Advantages:

We can easily block the unnecessary messages compare to existing system. Then the proposed system will distinguish between SPAM & HAM.

We are not supposed to block the users we can just oppose or block that type of least important message without blocking the user. So, the users can send any important message.

Electronic spam is the most troublesome Internet phenomenon challenging large global companies, including AOL, Google, Yahoo and Microsoft. Spam causes various problems that may, in turn, cause economic losses. Spam causes traffic problems and bottlenecks that limit memory space, computing power and speed. Spam causes users to spend time removing it.

Various methods have been developed to filter spam, including black list/white list, Bayesian classification algorithms, keyword matching, header information processing, investigation of spam-sending factors and investigation of received mails.

This study describes three machine-learning algorithms to filter spam from valid emails with low error rates and high efficiency using a multilayer perceptron model. Several widely used techniques include C4.5 decision tree classifier, multilayer perceptron and Naïve Bayes classifier, all of which are used for training data whether in the form of spam or valid emails. Finally, the results are discussed, and outputs of considered techniques are examined in relation to the proposed mode

3.3 Feasibility Study:

The feasibility of the system has been studied from the various aspects like whether the system is feasible technically operationally and economically. The present technology has found to be sufficient to meet the requirements of the system. This system is believed to work well when it is developed and installed hence operational feasibility is achieved since the requirement for the project are easily available where headed with the intention to use the available resource to fulfill the system requirement.

3.3.1 Technical Feasibility:

The technology needed for the proposed system that we are going to develop is available we can work for the project is done with current equipment existing tools like python we can develop our system using this technology if needed to upgrade in future if we want to use new technology like android app of our system it is possible hence the system that we are going to develop will successfully satisfy the needs of the system

3.3.2Economic Feasibility:

Since the system is develop as a part of project work there is no manual cost to spend for the proposed system also all the resources are already available it gives an indication the system is economically possible for the development economic justification is generally the bottom-line consideration for most systems the cost of conduct a full system investigation is negotiable because required information is collected. We can run our system in our normal hardware like desktop, laptop, mobiles. This system won't require extra specific software to use it. The project that we are going to develop won't require enormous amount of money to be developed so it will be economically feasible.

3.3.3 Operational Feasibility:

The user interface will be user friendly and no training will be required to use the application the solution proposed for our project is operationally workable and most likely convenient to solve the irrelevant document and fraud message.

3.4 Modules:

Importing the Libraries
Load Data sets
Data Preprocessing
Feature Extraction (FP-Growth)
Vector Creation
Classification
Naïve Bayes Algorithm
Finding the accuracy
Output

DATA SET AND RUNNING ENVIRONMENT:

We have manually collected a British English data set consisting of 425 SMS spam messages from the GrumbleText website.The GrumbleText website receives SMS spam reported by volunteer users. However, because of privacy concerns, private data, such as name, address and phone number, have been removed. The SMS are not chronologically sorted. We argue that our data set is reliable for this research. Compared with another public data set [25], ** our data set's messages are in the same language (British English) and from the same society (Britain).

FEATURE EXTRACTION:

An English stop word is used to remove meaningless words because these words exist in both e-mail spam and e-mail ham. However, removing such words will reduce the number of useful features to analyze because SMS is very short. As a simple example, we have two SMS spam: "$$$ buy free Viagra!" and "free SMS!", also one SMS ham: "buy book: p". We obtain five words: "buy", "free", "Viagra", "SMS", and "book". These words then enter the word vocabulary (|v|).

VECTOR CREATION:

Vector creation is a process used to map a raw tokenized word into numerical data that is ready to be classified by the text classification algorithm. We propose to use Word Occurrences because of its simplicity as our proposed approach was designed for mobile phones. We just need to count the number of words in each SMS message.

NAÏVE BAYES:

Once the word occurrences table is built, we can apply the Naïve Bayes approach to filter unknown incoming SMS. e the probability of SMS ham is higher than the probability of SMS spam, we can say that the SMS is ham. Intuitively, a human will agree that the SMS is ham, not spam, because there is no word, such as “free” or “Viagra”, present that is usually present in SMS spam. However, if the vocabulary count is high and the number of words is high, then the probability value will be too low for the processor to run the mathematical calculation, especially in a mobile phone. This problem is termed the underflow problem.

UPDATING FILTERING METHOD:

The feature extraction and vector creation steps are part of the training process. The filtering process step is part of the classification process. If we receive a new SMS and want to update the filtering system, we just need to repeat the feature extraction and vector creation steps. If the word already exists in the word occurrences table, we will just update the word occurrences table. If the word does not exist.

One common method of preventing automated spam postings is to employ a module that injects a captcha into common Drupal forms, like the user registration, comment and contact pages. If the captcha, a visual or auditory challenge, can be completed with the correct response, the user is assumed to be human and not an automated spam-posting process. See the captcha documentation or project pages for two modules that provide this functionality.

Mollom was a popular spam filtering service that analyzes content in real-time for spam-like characteristics. Content identified as spam is blocked, while clean content is processed normally. If Mollom is unsure whether the content is spam, it directs your site to require a successful CAPTCHA completion before the post is accepted. Mollom end of life has been announced. As of 2 April 2018, Acquia will no longer actively support or maintain Mollom. After that point in time, the Mollom service will no longer be available.

For filtering without any external services, the Spam module integrates a Bayesian filter along with some other filtering methods. Spam is only available for Drupal 6 (or below).

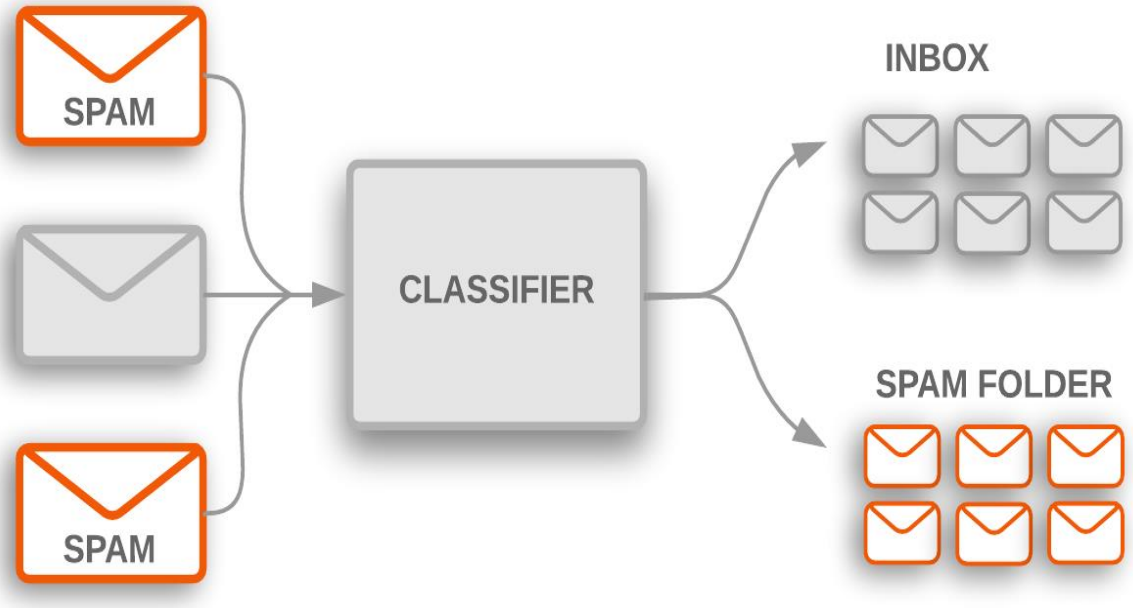
For Drupal 8 the Statistical Spam Filter module exists as an alternative.

Honeypot and modules like it use simpler techniques to filter spam without any user interaction, using a series of tests on the form itself.

These more passive modules are often a good first step towards spam prevention as they don't require any extra user interaction or steps to complete existing forms.

Anti-Spam module by Clean Talk to protect Drupal sites from spam bot registration and spam comments publications thru comment and contact forms. Clean Talk is a SaaS spam protection service for Web-sites. Clean Talk uses protection methods which are invisible for site visitors.

Using Clean Talk eliminates needs in CAPTCHA, questions and answers, and other methods of protection, complicating the exchange of information on the site. Clean Talk has an advanced option "Spam Firewall". This option allows blocking the most active spam bots before they get access to your website. It prevents spam bots from loading website pages so your web server doesn't have to perform all scripts on these pages.


SPAM
SPAM
CLASSIFIER
INBOX
SPAM FOLDER

4. SYSTEM DESIGN

4.1 Input Design:

Inaccurate input data are the most common causes of sms error in data processing. Errors entered by data entry operators can be controlled by the Input design. "Input design is the process of converting user originated inputs to computer-based formats". It consists of developing specification and procedure for data preparation.

a) Controlling amount of input: Due to so many reasons, design should control the quantity of data for input. Reducing the data requirement can lower cost by reducing labour expenses. By reducing input requirement, the analyst can speed the entire process from data capture to providing results to the users.

b) Avoiding delay: A processing delay resulting from data preparation or data entry operator is called bottleneck. Avoiding bottleneck should always be one objective of the analyst while designing output.

c) Avoiding errors in data: The rate at which sms errors occurs depends on the quantity of data, i.e. smaller the amount of data to input the fewer the opportunities for errors.

d) Keeping the process simple: Simplicity works and is accepted by the users. Complexity should be avoided when there are simple alternatives.

4.2 Output Design:

The term output necessarily implies to information on printed or displayed by an information system. Following are the activities that are carried out in output design stage.

Identification of specific output required to meet the information requirements.

Selecting of methods for processing outputs.

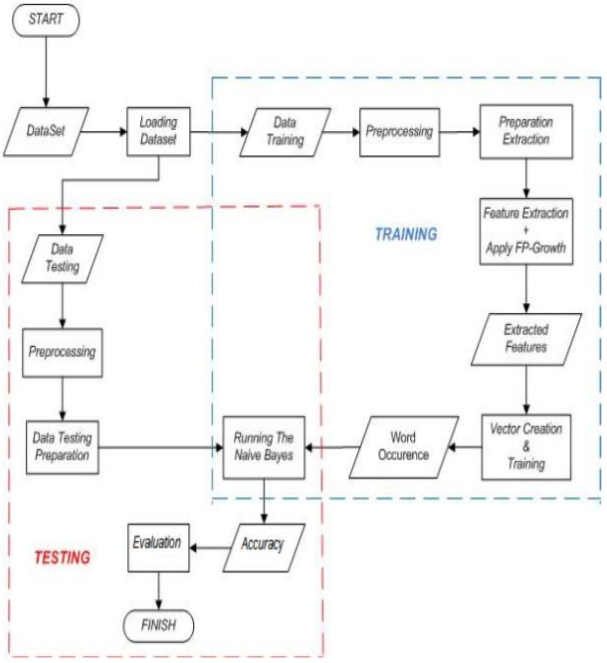
Designing of reports, formats or other documents that acts as a carrier of information.

4.2.1 Output Design Activities:

The output design of an information system must meet the following objectives:

1. The output design should provide information about the past, present or future events. The operational control level outputs provide operations of the past and present events. On the other hand, strategic planning level provides information of the future events.
2. The output design should indicate the important events smserror, opportunities and problems.
3. The output design should be designed keeping in mind that an action must be triggered in response to some event. A set of rule is pre- designed for such trigger.
4. The output design should produce some action to the transaction. For e.g. when the telephone bill is generated, a receipt is printed.

4.3 System Architecture:



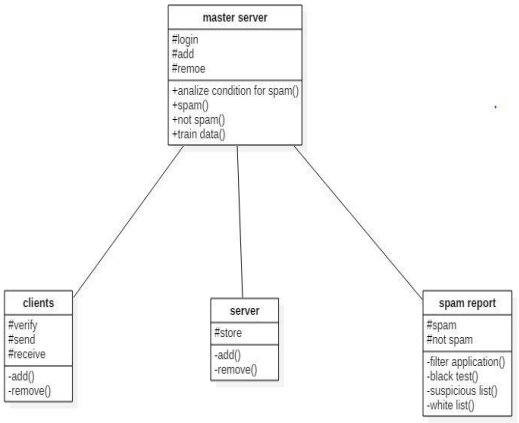
4.4 UML Diagrams: A UML diagram is a diagram based on the UML with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

4.4.1 Class Diagram:The class diagram is the main building block of object oriented modeling.It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code.It contain three compartments:

1)The top compartment contains the name of the class.

2)The middle compartment contains the attributes of the class.

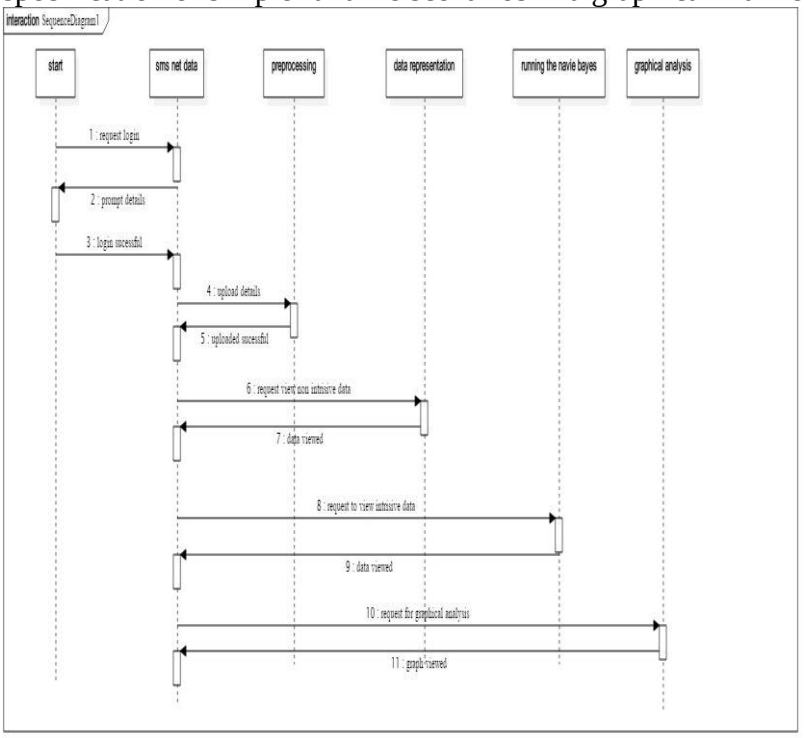
3)The bottom compartment contains the operations the class can execute.



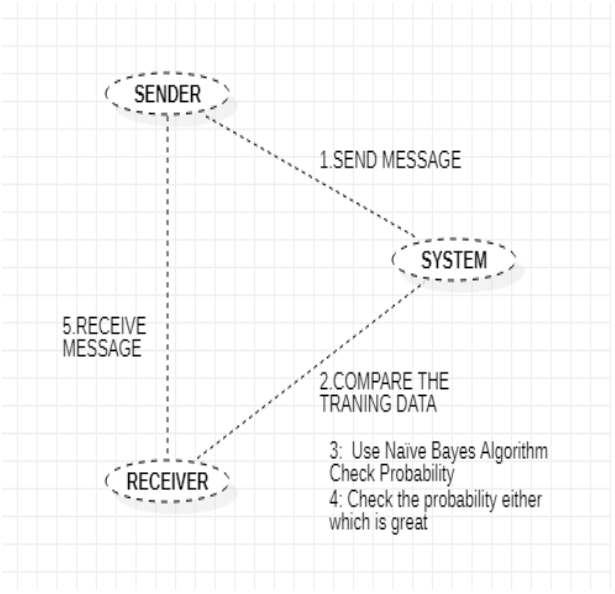
4.4.2 Use case Diagram: A use case diagram is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

4.4.3 Sequence Diagram: A sequence diagram shows object interactions arranged in time sequence. It

depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
A sequence diagram shows, as parallel vertical lines(lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

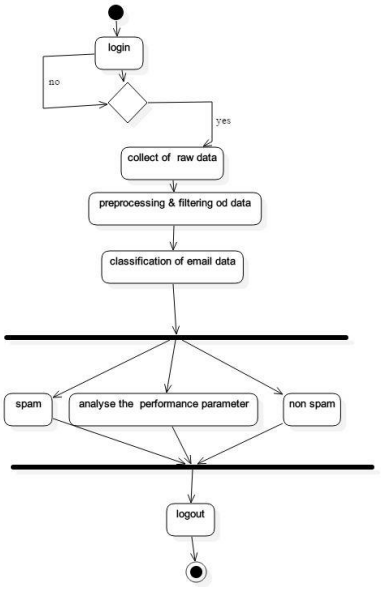


4.4.4 Collaboration Diagram: A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the UML. These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction.

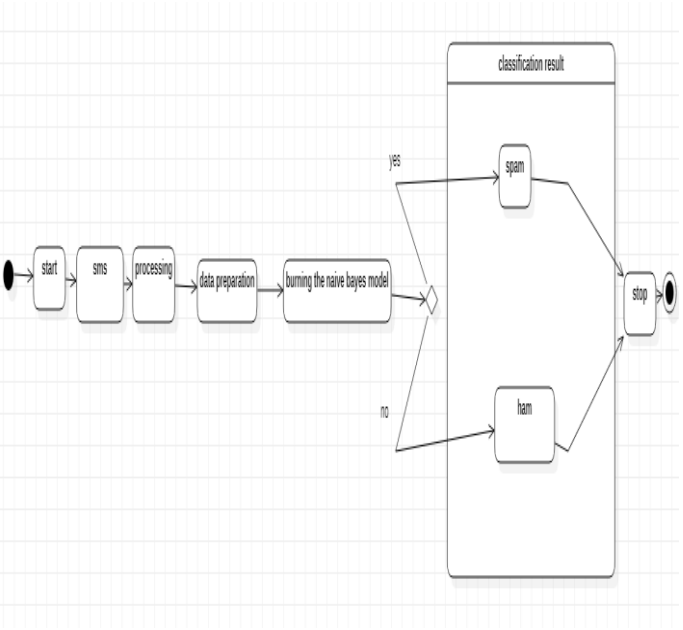


4.4.5 Activity Diagram: Activity diagram describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow

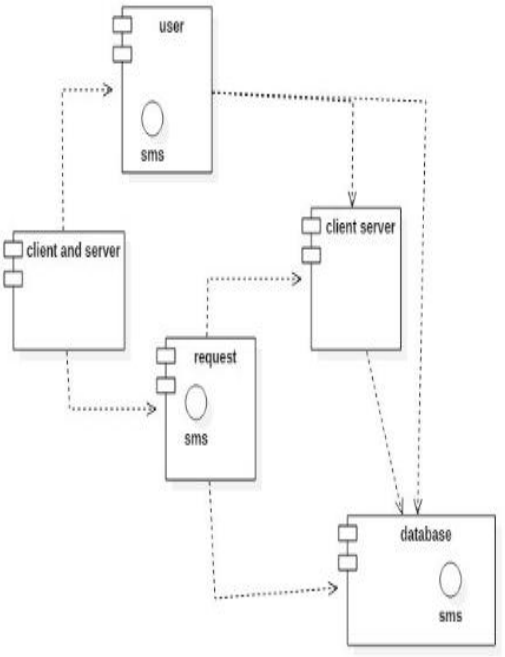
can be sequential, branched, or concurrent.
The basic purposes of activity diagram are it captures the dynamic behavior of the system. They are also used to construct the executable system by using forward and reverse engineering techniques.



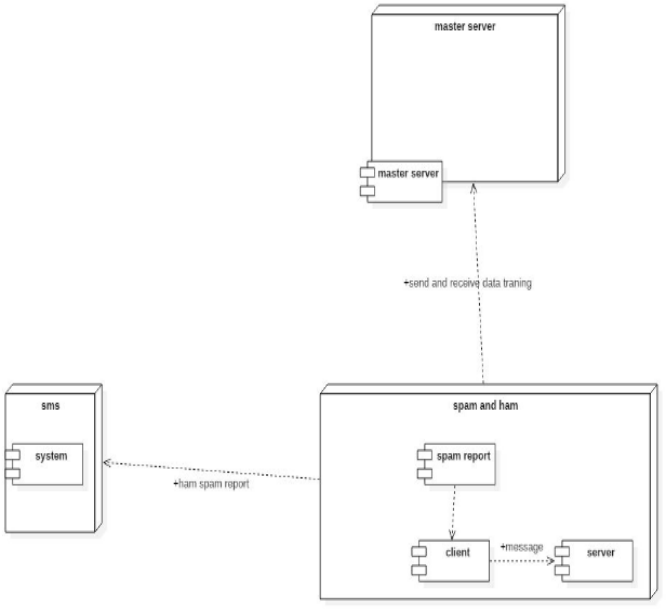
4.4.6. State Chart Diagram: State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. State chart diagram describes the flow of control from one state to another state.



4.4.7 Component Diagram: A component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems. These diagrams are also used as a communication tool between the developer and the stakeholders of the system.

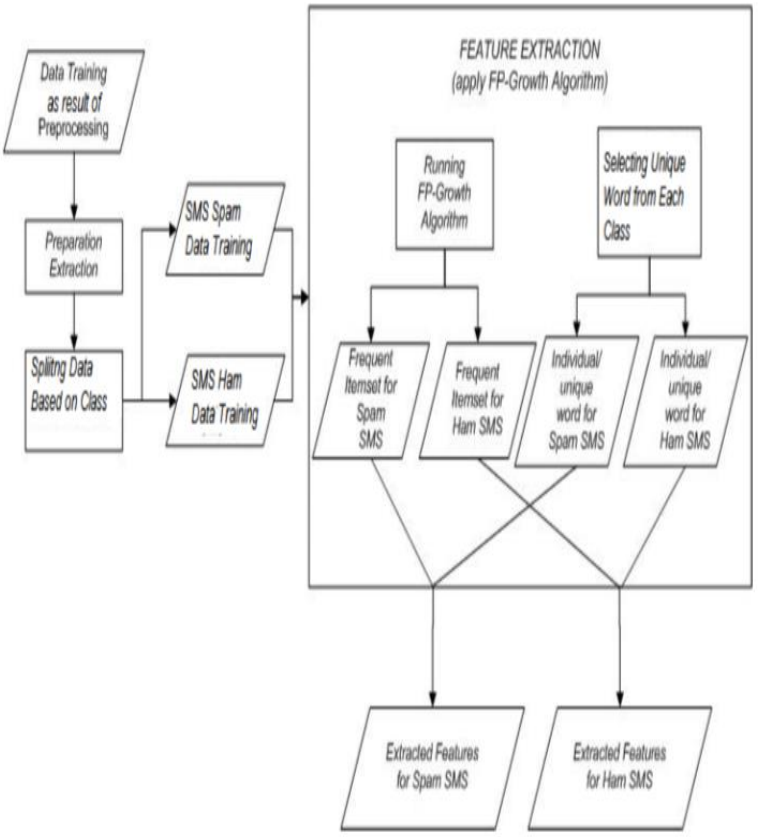


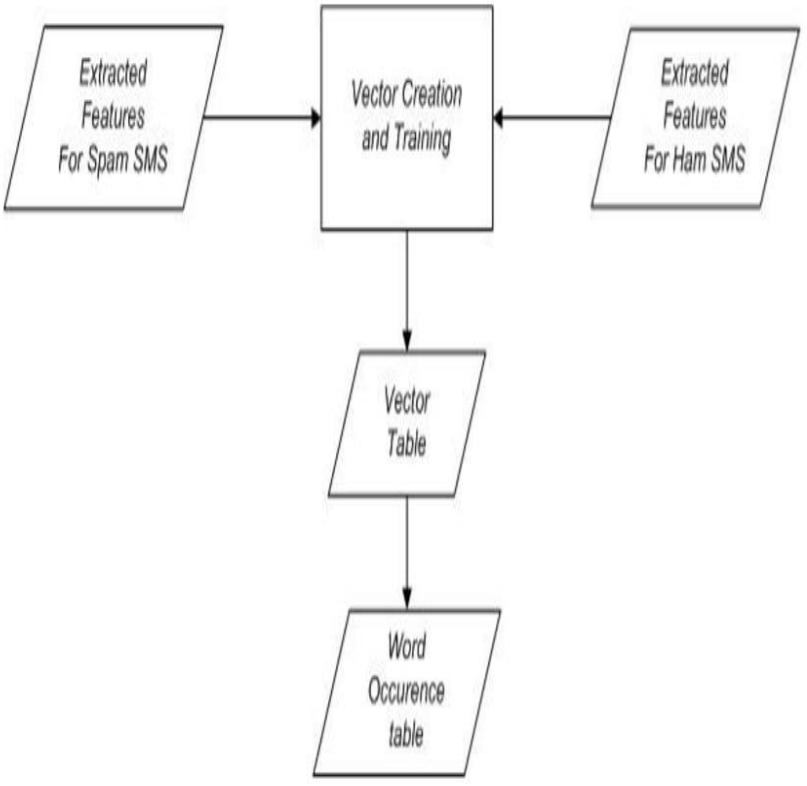
4.4.8 Deployment Diagram: A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. This diagram shows what hardware components exist, what software components run on each node and how the different pieces are connected.

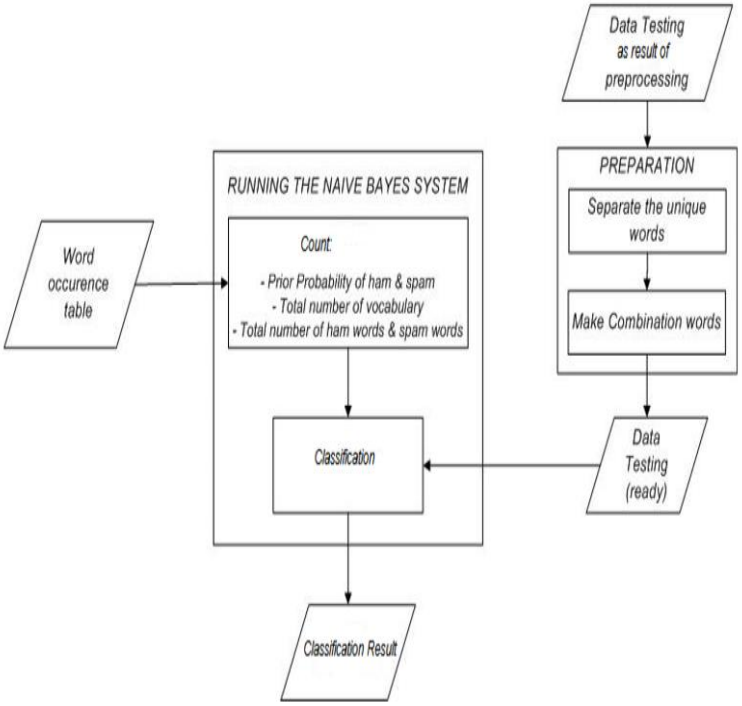


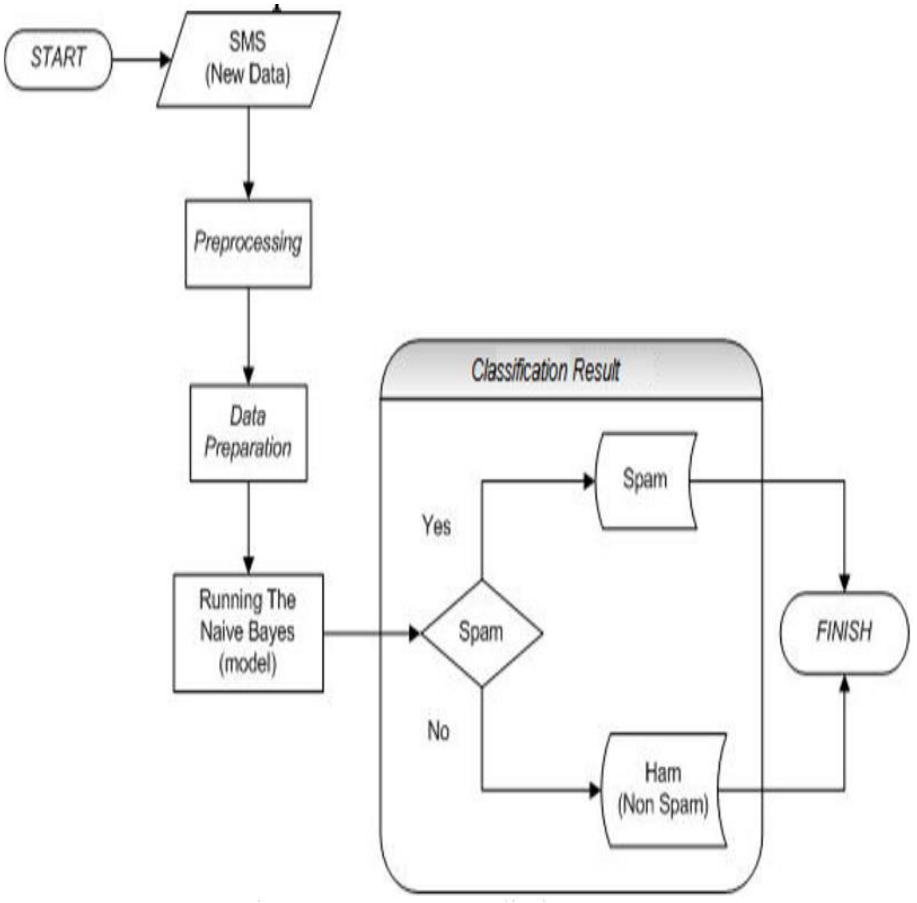
Data Flow Diagram:

Feature Extraction






Data Testing as result of preprocessing
PREPARATION
Separate the unique words
Make Combination words
RUNNING THE NAIVE BAYES SYSTEM
Word occurence table
Count:
- Prior Probability of ham & spam
- Total number of vocabulary
- Total number of ham words & spam words
Classification
Data Testing (ready)
Classification Result


START
SMS (New Data)
Preprocessing
Data Preparation
Running The Naive Bayes (model)
Classification Result
Spam
Yes
Spam
No
Ham (Non Spam)
FINISH

5. TESTING AND DEBUGGING

5.1 Introduction to Testing:

Test for the spam messages:

It’s one of the most dreaded words in the email marketing industry.
We’ve talked a lot about testing your design, but it’s also important to run spam tests to check your email’s deliverability potential.
A spam test checks your email to see whether certain spam filters will flag it and move it out of a subscriber’s inbox. The test looks at the content of your email and where you are sending it from. There are a couple of different ways to check your email for spam potential. If you’re using Email on Acid to test your emails, you can run a spam test with a seed list, run it through your smtp server, or send the test from our servers. This test is more thorough in checking for different aspects of your email that could flag it as spam; the test will not only review your content and subject line; it will also check if your IP address or domain is blacklisted. Seed list testing also provides more accurate test results.
We tested the database and found out that which message is spam and which message is non-spam indicated as 0 and 1 respectively we calculated the feature vector to know whether it is spam or non-spam using that feature vector naive Bayes algorithm works by comparing the trained data to test the data.
Data set: Data set is the collection of data or related information that is composed for separate elements a collection of datasets for the message spam contains spam and non-spam messages
Specific details of the SMTP protocol (did the sender do an HELO, EHLO, and what with name or IP, etc)
Attributes of the sending domain's DNS (does it resolve, SPF records, domain-keys records)
Reachability of the sender's MX (for reverse-path verification)
Source IP address of the server delivering the message
Content of the message
An "opinion" that third-party databases or "signatures" give about any of the above (DNSBLs, proprietary "reputation" databases, etc)
Mocking up delivery of messages to a filtering tool becomes less representative of reality if any of these factors aren't the same as what would occur during "real" delivery. Sending captured incoming email to a filtering rig in a manner that doesn't simulate the source IP address of the sending server, for example, impedes that filtering product's ability to act on that factor (running it by a DNSBL, etc). Likewise, sending delayed messages to a filtering (say you have a "canned" corpus of messages to test with) will give a false impression of behavior because the attributes of the sender's DNS and of any third-party databases or signatures may have changed since the time the message was originally sent (sender altered their SPF records to prevent false positives, a third-party service "blacklisted" the sender, etc).
I'd argue that it's impossible to completely mock-up reality, as it comes to delivery. Getting close going to be fairly difficult if you intend to simulate the sending server's IP address (and I'm not aware of any "off the shelf" solution that does that... and having gotten fairly good results from DNSBLs, I'd be concerned about *not* simulating the sending server's IP address.) Ultimately, you'll just have to get as close as you can afford.

Storage:

You've got to store the filtered email somewhere if you intend to analyze the results. Without storage of some kind there's no good way to actually view the results. Sure, the filtering software generates statistics, but it would certainly improve my comfort-level if I could see the filtered messages.
Some filtering products have a storage capability built-in (like MailMarshal, for example). Other products expect to have an email system to deliver into and don't have any storage capability. To test those products, and so as not to disrupt your production email system, you'll have to create up some kind of secondary email infrastructure to store the test filtering results.
If licensing expense is a concern, you can use free and open-source tools to prevent incurring licensing expense. That may present a learning curve for the testing personnel.

More convoluted "groupware"-type email systems may present a challenge in mocking-up because of their dependencies on other services. Exchange Server will require you to mock-up an Active Directory infrastructure to host the mailboxes. Other "groupware"-type email systems (Notes, Groupwise, etc) will have their own associated degrees of difficulty in creating parallel infrastructures.
Scroll through your inbox and look for junk email to see whether your spam filter is performing as expected. If you find spam, use the management tools in your particular program to flag those emails and then create new spam filter rules to manage them.
Check the Spam, Junk and Trash folders to determine if your spam filter is moving email designated as spam to the correct areas.
Send yourself an email that contains one or more words or phrases in the Subject line or message body that you previously set your spam filter to identify as spam, such as “Free,” “Please Read This Immediately!” or “You’ve Won!” Wait a minute or two and then check to see if the spam transferred to the correct folder.
Set up a new spam filter rule using different words or phrases. Send yourself another test message and check the folders again
Use a second email provider to test if your spam filter is handling blocked email addresses and domains correctly. Set your spam filter to block a secondary email account address or domain. Send an email to your primary account from that secondary account and look to see if your email client filtered the spam correctly
Browse to websites that offer free email testing tools such as Byte plant and GFI (links in resources). Follow the on-screen instructions to test your spam filter.

Difference between verification, validation and testing:

Verification is a process in which a design is tested (or verified) against a given specification before manufacturing. This can be done using several methods like software simulations, static formal analysis and FPGA/hardware emulation.

Validation is a process in which the manufactured design is tested for all functional and electrical correctness in a lab set up. This includes having the real chip assembled on a test board or a reference board, running real software/applications and making sure that all features work well.

Testing (or Manufacture testing) mostly involves running certain reliable test patterns on each chip before volume shipment. Once a product is validated and is ready for shipment in large volumes, test patterns are run to identify any defective products during fabrication as well as for “binning” parts for different skews.

Some companies use these terminologies interchangeably. For example, in certain companies Verification is also known as Pre-Silicon Validation and Validation is known as Post-Silicon Validation.

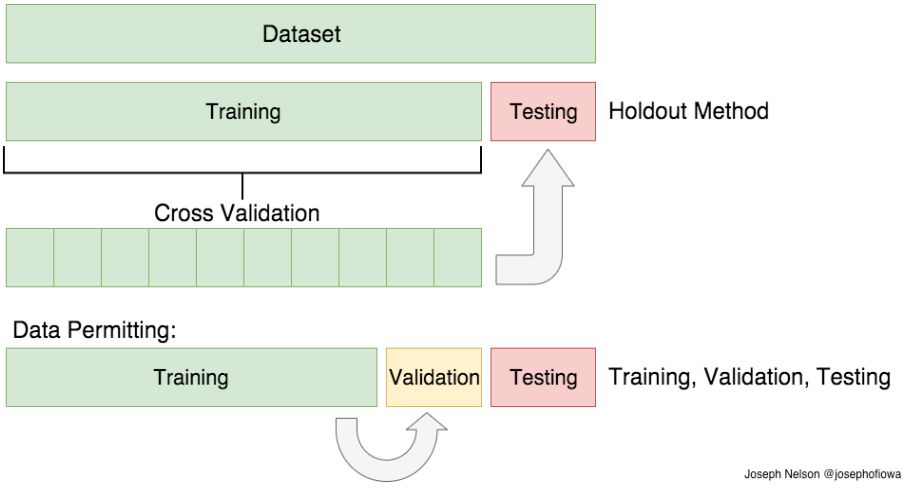


Fig 1: testing, training and validation

Cross Validation:

In the previous paragraph, I mentioned the caveats in the train/test split method. In order to avoid this, we can perform something called cross validation. It's very similar to train/test split, but it's applied to more subsets. Meaning, we split our data into k subsets, and train on k-1 one of those subsets. What we do is to hold the last subset for test. We're able to do it for each of the subsets.

Train/Test Split:

As I said before, the data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset.

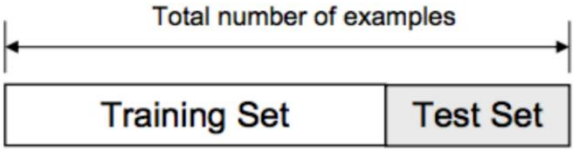


Fig 2: example for train/test split

Training:

The training data is an initial set of data used to help a program understand how to apply technologies like neural networks to learn and produce sophisticated results. It may be complemented by subsequent sets of data called validation and testing sets.
Training data is also known as a training set, training dataset or learning set.
The training set is the material through which the computer learns how to process information. Machine learning uses algorithms – it mimics the abilities of the human brain to take in diverse inputs and weigh them, in order to produce activations in the brain, in the individual neurons. Artificial neurons replicate a lot of this process with software – machine learning and neural network programs that provide highly detailed models of how our human thought processes work.
With that in mind, training data can be structured in different ways. For sequential decision trees and those types of algorithms, it would be a set of raw text or alphanumerical data that gets classified or otherwise manipulated. On the other hand, for convolutional neural networks that have to do with image processing and computer vision, the training set is often composed of large numbers of images. The idea is that because the machine learning program is so complex and so sophisticated, it uses iterative training on each of those images to eventually be able to recognize features, shapes and even subjects such as people or animals. The training data is absolutely essential to the process – it can be thought of as the "food" the system uses to operate.
Training data is the main and most important data which helps machines to learn and make the predictions. This data set is used by machine learning engineer to develop your algorithm and more than 70% of your total data used in the project.

Difference between testing and trained data:

Training set is the one on which we train and fit our model basically to fit the parameters.
Test data is used only to assess performance of model. Training data's output is available to model whereas testing data is the unseen data for which predictions have to be made.

Training phase:

A phase where you are basically training the algorithms to create the right output. In the in the learning phase, you are having the input parameters. ... While training, the algorithm modifies the training parameters. It also modifies the used data and then you are getting to an output.
Software Testing is a process of evaluating the functionality of a software application to find any software bugs. It checks whether the developed software met the specified requirements and identifies any defect in

the software in order to produce a quality product.
This is basically executing a system in order to identify any gaps, errors, or missing requirements contrary to the actual requirements. It is also stated as the process of verifying and validating a software product. With this, you can check whether the software product:
Meets the business and technical requirements that guided its design and development
Works as per the requirement
Can be implemented with the same characteristics

Software Testing Strategies:

An efficient software testing or QA strategy requires testing of all technology stack levels to ensure that every part, as well as the entire system, works without breaking down. Some of the Software Testing Strategies include:

5.2 Dimensions of Testing:

1)Demonstrate, Check, Confirm, Verify, Validate:

First of all testing is about demonstrating in a constructive way that something works, at least to some extent. Further it is about checking and confirming artifacts like requirements, features or use cases. And it is also about verifying and validating.

2)Detect, Search

In testing we try to detect bugs as early as possible in a destructive way, and we search for unknown and therefore unspecified behavior in the system under test**.**

3)Mitigate, Reduce Risks, Investigate, Explore

Any testing should be based on risks in the system under test, so this is also a very important dimension in testing. Thereby we also investigate and explore in order to look deep inside the system under test.

4)Measure, Assess, Evaluate and Predict

During testing we collect data and measure, for example to address quality attributes like performance, reliability or availability. Based on these data we know something about the current status, and we can hopefully make a good forecast for the future.

5.3 Stages of Testing:

Testing Stage 1 – Test Plan
Software testing should always begin with establishing a well-thought-out test plan to ensure an efficient execution of entire testing process. Efficient test plan must include clauses concerning the amount of work to be done, deadlines and milestones to be met, methods of testing and other formalities like contingencies and risks.
Testing Stage 2 – Analysis
At this stage, functional validation matrix is made. In-house or offshore testing team analyzes the requirements and test cases which are to be automated and which are to be tested manually.
Testing Stage 3 – Design
If the testing team has reached this stage, it means that there is no confusion or misunderstanding concerning the test plan, validation matrix or test cases. At Designing Stage testing team makes suitable scripts for automated test cases and generates test data for both automated and manual test cases.
Testing Stage 4 – Development
At this stage, scripting is provided. In particular cases, development stage also includes unit tests and generating of performance and stress test plans. Usually, it happens when testing starts together with the software development process.
Testing Stage 5 – Execution
As soon as the entire scripting have been made its execution begins. Firstly, testing team executes unit tests, and then functionality tests. They identify bugs on the superficial level and report to the
software developers. After that the detailed testing is carried out. Execution stage is completed when test and bug reports are made up.

Testing Stage 6 – Bug fixing
When testing team identifies the bugs, they send it to IT development team. If development team considers to fix the bugs', testing team has to retest the software in order to check that no new bugs have been created while fixing.
Testing Stage 7 – Software is implemented
This is the final stage of the software testing when all test cases are executed and all procedures are carried out. The software is delivered to the end user who tests it and reports if any bugs take place.

5.4 Types of Testing:

5.4.1 Unit Testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.4.2 Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

5.4.3 System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.4.4 Interface Testing:

When an application or a software or a website is developed, then there are several components of it. Those components can be server, database etc. The connection which integrates and facilitates the communication between these components is termed as an Interface. It verifies that communication between the systems are done correctly.

3 phases of Interface Testing:

Configuration & development: When the interface is configured, and once the development starts, the configurations need to be verified as per the requirement.

Validation: When the development is completed, the interface needs to be validated and verified, this can be done as a part of unit testing also.

Maintenance: When we start developing an interface, we need to make sure that we are not introducing any defects in our code and hence tests need to be run on the interface.

5.4.5 User Acceptance Testing:

Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery

1. **Alpha Testing:** This is to assess the Product in the development/testing environment by a specialized testers team usually called alpha testers.

2. **Beta Testing:** This is to assess the Product by exposing it to the real end-users, usually called beta testers/beta users, in their environment. Continuous feedback from the users is collected and the issues are fixed.

5.4.6 White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

5.4.7: Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6. IMPLEMENTATION

Source Code:

Spam Mail Prediction:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

df = pd.read_csv("mail_data.csv")

print(df)

data = df.where((pd.notnull(df)), "")

data.head(10)

data.info()

data.describe()

data.columns

data.shape

data.loc[data['Category'] == 'spam', 'Category',] = 0
data.loc[data['Category'] == 'ham', 'Category',] = 1

X = data['Message']
Y = data['Category']

print(X)

print(Y)

X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size = 0.2, random_state =3)

print(X.shape)
print(X_train.shape)
print(X_test.shape)

print(Y.shape)
print(Y_train.shape)
print(Y_test.shape)

feature_extraction =TfidfVectorizer(min_df = 1, stop_words = 'english', lowercase = 'True')
X_train_features = feature_extraction.fit_transform(X_train)
X_test_features = feature_extraction.transform(X_test)
Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')
```

```
print(X_train)

print(X_train_features)

model = LogisticRegression()

print(model.fit(X_train_features, Y_train))

prediction_on_training_data = model.predict(X_train_features)
accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)

print('Acc on training data:', accuracy_on_training_data)

prediction_on_test_data = model.predict(X_test_features)
accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)

print('acc on test data:', accuracy_on_test_data)

input_your_mail = ["This is the 2nd time we have tried to contact you. You have won the prize. To claim is easy, just call 12345567890 now!"]
input_data_features = feature_extraction.transform(input_your_mail)
prediction = model.predict(input_data_features)
print(prediction)
if(prediction[0] == 1):
   print("Ham mail")
else:
   print("Spam mail")

input_your_mail = ["Today I'm going to class, so can't attend the session"]
input_data_features = feature_extraction.transform(input_your_mail)
prediction = model.predict(input_data_features)
print(prediction)
if(prediction[0] == 1):
   print("Ham mail")
else:
   print("Spam mail")
```

7. RESULTS

7.1 Outputs:

```
     Category                                            Message
0         ham  Go until jurong point, crazy.. Available only ...
1         ham                      Ok lar... Joking wif u oni...
2        spam  Free entry in 2 a wkly comp to win FA Cup fina...
3         ham  U dun say so early hor... U c already then say...
4         ham  Nah I don't think he goes to usf, he lives aro...
...       ...                                                ...
5567     spam  This is the 2nd time we have tried 2 contact u...
5568      ham              Will ü b going to esplanade fr home?
5569      ham  Pity, * was in mood for that. So...any other s...
5570      ham  The guy did some bitching but I acted like i'd...
5571      ham                        Rofl. Its true to its name

[5572 rows x 2 columns]
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Category  5572 non-null   object
 1   Message   5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

	Category	Message
count	5572	5572
unique	2	5157
top	ham	Sorry, I'll call later
freq	4825	30

```
Index(['Category', 'Message'], dtype='object')
```

```
(5572, 2)
```

```
0       Go until jurong point, crazy.. Available only ...
1                           Ok lar... Joking wif u oni...
2       Free entry in 2 a wkly comp to win FA Cup fina...
3       U dun say so early hor... U c already then say...
4       Nah I don't think he goes to usf, he lives aro...
                              ...
5567    This is the 2nd time we have tried 2 contact u...
5568                 Will ü b going to esplanade fr home?
5569    Pity, * was in mood for that. So...any other s...
5570    The guy did some bitching but I acted like i'd...
5571                            Rofl. Its true to its name
Name: Message, Length: 5572, dtype: object
```

```
0       1
1       1
2       0
3       1
4       1
       ..
5567    0
5568    1
5569    1
5570    1
5571    1
Name: Category, Length: 5572, dtype: object
```

```
(5572,)
(4457,)
(1115,)
```

```
(5572,)
(4457,)
(1115,)
```

```
3075                  Don know. I did't msg him recently.
1787    Do you know why god created gap between your f...
1614                          Thnx dude. u guys out 2nite?
4304                                       Yup i'm free...
3266    44 7732584351, Do you want a New Nokia 3510i c...
                              ...
789     5 Free Top Polyphonic Tones call 087018728737,...
968     What do u want when i come back?.a beautiful n...
1667    Guess who spent all last night phasing in and ...
3321    Eh sorry leh... I din c ur msg. Not sad alread...
1688    Free Top ringtone -sub to weekly ringtone-get ...
Name: Message, Length: 4457, dtype: object
```

```
  (0, 5413)     0.6198254967574347
  (0, 4456)     0.4168658090846482
  (0, 2224)     0.413103377943378
  (0, 3811)     0.34780165336891333
  (0, 2329)     0.38783870336935383
  (1, 4080)     0.18880584110891163
  (1, 3185)     0.29694482957694585
  (1, 3325)     0.31610586766078863
  (1, 2957)     0.3398297002864083
  (1, 2746)     0.3398297002864083
  (1, 918)      0.22871581159877646
  (1, 1839)     0.2784903590561455
  (1, 2758)     0.3226407885943799
  (1, 2956)     0.33036995955537024
  (1, 1991)     0.33036995955537024
  (1, 3046)     0.2503712792613518
  (1, 3811)     0.17419952275504033
  (2, 407)      0.509272536051008
  (2, 3156)     0.4107239318312698
  (2, 2404)     0.45287711070606745
  (2, 6601)     0.6056811524587518
  (3, 2870)     0.5864269879324768
  (3, 7414)     0.8100020912469564
  (4, 50)       0.23633754072626942
  (4, 5497)     0.15743785051118356
  :     :
 (4454, 4602)  0.2669765732445391
 (4454, 3142)  0.32014451677763156
 (4455, 2247)  0.37052851863170466
 (4455, 2469)  0.35441545511837946
 (4455, 5646)  0.33545678464631296
 (4455, 6810)  0.29731757715898277
 (4455, 6091)  0.23103841516927642
 (4455, 7113)  0.30536590342067704
 (4455, 3872)  0.3108911491788658
 (4455, 4715)  0.30714144758811196
 (4455, 6916)  0.19636985317119715
 (4455, 3922)  0.31287563163368587
 (4455, 4456)  0.24920025316220423
 (4456, 141)   0.292943737785358
 (4456, 647)   0.30133182431707617
 (4456, 6311)  0.30133182431707617
 (4456, 5569)  0.4619395404299172
 (4456, 6028)  0.21034888000987115
 (4456, 7154)  0.24083218452280053
 (4456, 7150)  0.3677554681447669
 (4456, 6249)  0.17573831794959716
 (4456, 6307)  0.2752760476857975
 (4456, 334)   0.2220077711654938
 (4456, 5778)  0.16243064490100795
 (4456, 2870)  0.31523196273113385
```

```
LogisticRegression()
```

```
Acc on training data: 0.9670181736594121
```

```
acc on test data: 0.9659192825112107
```

```
[0]
Spam mail
```

```
[1]
Ham mail
```

8. CONCLUSION

Based on the analysis of the tests performed in this research, it can be concluded that:
Both methods used in this research, the performances of both methods are equally well for SMS classification with average of the accuracy above 90%. The use of collaboration methods, Naive Bayes and FP-Growth, is superior to the average accuracy for each dataset.

The Accuracy best average is obtained when the SMS Spam Collection v.1 dataset with the 9%minimum support is used and the implementation of the FP-Growth has accuracy up to 98.506%.
The use of datasets with varied training data is agreeable to be applied by using the FP-Growth. By implementing the FP-Growth for feature extraction, it can elevate the score of precision. Thus, the system becomes more precise in providing the information requested by the users in response to the SMS classification.

9. FUTURE SCOPE

1. Though, thesis has made efforts towards solving the problem of Spam E-mail using legislative, behavioral and technological measures, the solution proposed are not complete solutions.
2. The problem of Spam E-mail and Anti-Spam solution is game of cat and mouse since, every day Spammer will come up with new techniques of sending Spam E-mails. This work has given the potential direction for classification of the Spam E-mails.

The future efforts would be extended towards:

1. Achieving accurate classification, with zero percent (0%) misclassification of Ham E-mail as Spam and Spam E-mail as Ham.
2. The efforts would be applied to block Phishing E-mails, which carries the phishing attacks and now-days which is more matter of concern.
3. Also, the work can be extended to keep away the Denial-of-Service attack (DoS) which has now, emerged in Distributed fashion called as Distributed Denial of Service Attack (DDoS).
4. Getting caught in the spam folder is an email marketer's nightmare, but it's more common than you might think. If you're not watching and paying attention, it's easy to get your emails blocked or blacklisted and not know it. Or there might be issues with your infrastructure or the content of your email that can trigger spam filters, preventing your messages from being delivered to the inbox.
5. But how do you know if your email is at risk of being sent to the spam folder? Spam filter testing tools are instrumental in getting this critical visibility.
6. We asked thousands of marketers if they run their emails through spam filter tests before sending—and which tools they use to do so. Here's what we found.
7. 53.8% of all marketers say they rely on spam filter testing tools to test their emails for potential spam issues before they send**,** according to our State of Email research. Third-party spam testing tools are the most popular, with 37.7% of brands using them. Another 16.1% of brands rely on the spam testing tools provided by their email service providers (ESPs) for pre-send spam testing.
8. Marketers who use third-party spam filter testing tools are 50% more likely to report being blacklisted (22% vs. 15%) than those who don't.
9. But that doesn't mean that those brands get blacklisted more often. It's safe to assume that most of these differences are due to tool users having much better visibility into their deliverability. After all, you can't report or resolve a blacklisting if you aren't aware of it.
10. Here's another stat that supports that theory: Brands who use spam filter tests before sending emails reported an ROI of 51:1, while those who don't report an ROI of 39:1. Having better visibility into blocks and black listings means you can take action more quickly—and that leads to better email program performance**.**
11. The most popular spam filter testing tools
12. Many brands use third-party tools in addition to their ESP's functionality to prevent blocking and blacklisting. Out of the 600+ marketers that said they use third-party spam filter testing tools, over half (59.6%) relies on Litmus Spam Testing to ensure their emails pass spam filter tests prior to sending an email.

10. REFERENCES

https://ieeexplore.ieee.org/abstract/document/7811442/
https://onlinelibrary.wiley.com/doi/full/10.1002/sec.577

Books to refer:

Advance image-based spam detection and filtering techniques

- Dhavale, Sunita Vikrant

Spam assassin

-alan schwartz

CERTIFICATE

राष्ट्रीय लघु उद्योग निगम–तकनीकी सेवा केन्द्र
एन एस आई सी
N S I C
THE NATIONAL SMALL INDUSTRIES CORPORATION LTD.
TECHNICAL SERVICES CENTRE
(भारत सरकार का उद्यम / A Government of India Enterprises)
ई.सी.आई.एल एक्स रोड, कुशाईगुडा, हैदराबाद – 500062, तेलंगाना, भारत
E.C.I.L X Road, Kushaiguda, Hyderabad - 500062, Telangana, India.

क्रमांक / S.No. 203817

दिनांक / Date: 14/06/2023

Certificate

This is to certify that Mr. / Ms. Jainu Varsha Priya
son/daughter of Mr. Jainu Santha Kumar pursuing BTech in CSE from
(College Name) Gayatri Vidya Parishad College of Engineering for Women
Roll No. 20JG1A0529 has successfully completed the Internship Program
entitled/in the area of Data Science Using Python
under
our guidance. It is a bonafide work carried out by her/him from 15/05/2023 to 14/06/2023
He/She has completed the assigned module as per the requirements within the time frame
During the above period, the trainee's conduct was found Good

Project Coordinator

Technical Services Centre NSIC Hyderabad-500062 T.S.

Centre Head

यह प्रमाणपत्र बस होलोग्राम लगा होने के साथ ही मान्य होगा / This Certificate shall be valid only with affixed hologram