A Project Report on

EMAIL SPAM DETECTION USING NLP

Dissertation submitted in partial fulfillment of the requirement for the award of the degree of B.Tech.

By

KUMILI VAMSI PRASAD
16VV1A0529

MANDALAPU YOGALEENA
16VV1A0534

CHADUVULA PRIYANKA
16VV1A0510

TANKALA HARSHA VARDHAN
16VV1A0554

Under the Esteemed Guidance of

Mr. T. Siva Rama Krishna

Assistant Professor

Department of Computer Science & Engineering

University College of Engineering Vizianagaram

JNTUK – Vizianagaram Campus

VZIANAGARAM – 535 003, A.P., INDIA

(2016-2020)

Department of Computer Science & Engineering

JNTUK-University College of Engineering Vizianagaram

VZIANAGARAM – 535 003, A.P., INDIA

CERTIFICATE

This is to certify that the dissertation entitled "EMAIL SPAM DETECTION USING NLP" that is being submitted by KUMILI VAMSI PRASAD (16VV1A0529), MANDALAPU YOGALEENA (16VV1A0534), CHADUVULA PRIYANKA (16VV1A0510), TANKALA HARSHA VARDHAN (16VV1A054) in partial fulfilment for the award of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING to the JNTUK-University College of Engineering Vizianagaram is a record of bonafide work carried out by them under our guidance and supervision.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Signature of Supervisor

Mr. T. Siva Rama Krishna
Assistant professor
Dept. of CSE
JNTUK- UCEV

Asst. Professor
Dept. of Computer Science & Engineering
JNTUK University College of Engineering
Vizianagaram-535 003, A.P., India

Head of the Department

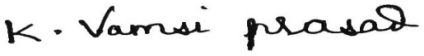
Dr. A.S.N. Chakravarthy
Professor
Dept. of CSE
JNTUK- UCEV

Dr.A.S.N. CHAKRAVARTHY
B.E., M.Tech., Ph.D
Head of the Department
Dept. of Computer Science & Engineering
University College of Engineering
JNTUK Vizianagaram

DECLARATION

We, *KUMILI VAMSI PRASAD, MANDALAPU.YOGALEENA, CHADUVULA PRIYANKA, TANKALA HARSHA VARDHANN* hereby declare that the project report titled "EMAIL SPAM DETECTION" submitted to JNTUK University College of Engineering Vizianagaram, in partial fulfillment of the requirements for the award of the degree of B.Tech. in *COMPUTER SCIENCE AND ENGINEERING* is award of original and independent research work done by us during the academic year 2019-2020 under the supervision of Mr. T.SIVA RAMA KRISHNA and it has not formed the basis of any Degree/Diploma/Associateship/Fellowship or other similar title to any candidate in University.

Signature of the Candidates

1.KUMILI VAMSI PRASAD 16VV1A0529 

2.MANDALAPU YOGALEENA 16VV1A0534

3.CHADUVULA PRIYANKA 16VV1A0510 

4.TANKALA HARSHA VARDHAN 16VV1A0554 

Place: Vizianagaram

Date: 19-07-2020

ACKNOWLEDGEMENT

This acknowledgement transcends the reality of formality when we express deep gratitude and respect to all those people behind the screen who inspired and helped us in the completion of this project work.

We also take the privilege to express my heartfelt gratitude to my guide **Mr. T. Siva Rama Krishna**, Asst. Prof., CSE, of JNTUK UCEV for his valuable suggestions and constant motivation that greatly helped me in successful completion of the project. We express my sincere thanks to **Prof. A.S.N.Chakravarthy**, Head of the Department of Computer Science and Engineering for his continuous support. We express our sincere thanks to our respected Principal **Prof. G. Swami Naidu**, Vice-principal **Dr. R. Rajeswara Rao** with a great sense of pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We owe many thanks to a many people who helped and supported us during the writing of this report.

We are thankful to all faculty members for extending their kind cooperation and assistance. Finally, we are extremely thankful to our parents and friends for their constant help and moral support.

1.K.VAMSI PRASAD (16VV1A0529)
2.M.YOGALEENA (16VV1A0534)
3.CH.PRIYANKA (16VV1A0510)
4.T.HARSHA VARDHAN (16VV1A0554)

ABSTRACT

With the digital age where we are always connected and constantly generating and consuming multitude of digital content. Email has become the most widely used and economic form of communication in this digital era. Email users generally get bombarded with unsolicited messages regarding direct marketing often sent to multiple users using bots. Users have to spend a considerable amount of time on clearing such messages. Study shows that there is sharp increase in spam emails , it is estimated that they are almost 89% of the total email traffic. Spam emails can create a havoc by causing financial loss or identity theft of users.

This can not be handled manually by creating filters. Spammers use many techniques to bypass manual filters such as misspelled words by adding extra letters to words (eg: amazingg, amaze-on etc..), synonyms of generally used words etc.. Use of Machine learning models can handle such data. Thus creating text classifiers that precisely filter such emails from the users's mail box to spam folder is practiced now a days and is more efficient than manual filters. The classification model needs to be trained to filter efficiently.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROBLEM STATEMENT

Recently unsolicited commercial or bulk e-mail also known as spam, become a big trouble over the internet. Spam is waste of time, storage space and communication bandwidth. The problem of spam e-mail has been increasing for years. In recent statistics, 40% of all emails are spam

which about 15.4 billion email per day and that cost internet users about $355 million per year. Automatic e-mail filtering seems to be the most effective method for countering spam at the moment and a tight competition between spammers and spam-filtering methods is going on.

Only several years ago most of the spam could be reliably dealt with by blocking e-mails coming from certain addresses or filtering out messages with certain subject lines. Spammers began to use several tricky methods to overcome the filtering methods like using random sender addresses or append random characters to the beginning or the end of the message subject line . Knowledge engineering and machine learning are the two general approaches used in e-mail filtering. In knowledge engineering approach a set of rules has to be specified according to which emails are categorized as spam or ham. A set of such rules should be created either by the user of the filter, or by some other authority (e.g. the software company that provides a particular rule-based spam-filtering tool). By applying this method, no real promising results shows because the rules must be constantly updated and maintained, which is a waste of time and it is not convenient for most users. Machine learning approach is more efficient than knowledge engineering approach; it does not require specifying any rules . Instead, a set of training samples, these samples is a set of pre classified e-mail messages. A specific algorithm is then used to learn the classification rules from these e-mail messages. Machine learning approach has been widely studied and there are lots of algorithms can be used

in e-mail filtering. They include Naive Bayes, support vector machines, Neural Networks and K-nearest neighbour.

1.2 MOTIVATION

Spam refers to unsolicited business email. Otherwise called junk mail, spam floods Internet client's electronic mailboxes. These junk emails can contain different sorts of messages, for example, commercial advertising, pornography, business promoting, doubtful product, infections or quasi-legal services

1.3 TYPES OF SPAM DETECTION

Fundamentally, spam can be classified into the accompanying four types:

- Usenet Spam
- Texting Spam
- Mobile Spam
- E-mail Spam

Usenet Spam: User Network is an open get to arrange on the Internet that gives group talks and group email informing. All the data that goes over the Web is called "Net News" and a running accumulation of messages about a specific topic is known as a "newsgroup". Usenet spam is presenting of some commercial on the newsgroups. Spammers focus on the clients those read news from these newsgroups. Spammers present promotion on a substantial measure of newsgroups at once. Usenet spam rob clients of the utility of the newsgroups by overwhelming them with a barrage of promoting or other unrelated posts.

Instant Messaging Spam: Instant informing frameworks, for example, Yahoo Messenger, AOL Instant Messenger (AIM), Windows Live Messenger, Facebook Messenger, XMPP, Tencent QQ, Instant Messaging Client (ICQ), and MySpace talk rooms are all objectives for spammers. A few IM frameworks give a registry of clients, including statistic data, for example, date of birth and gender. Advertisers can gather this data, sign on to the framework, and send

undesirable messages, which could incorporate business malware, viruses, and associates to paid destinations [8]. As texting has a tendency to not be stuck by firewalls; ***Shradhanjali, Verma Toran; International Journal of Advance Research, Ideas and Innovations in Technology.*** subsequently, it is a particularly helpful route for spammers. It focuses on the clients when they join any visiting space to discover new friends. It ruins appreciate of individuals and wastes their time moreover.

Mobile Phone Spam: Mobile phone spam is focused on the content informing administration of a cell phone. This can be particularly irritating to clients not just for the bother additionally in light of the cost they might be charged per instant message gotten in a few markets. This sort of spam more often than not contains a few plans and offers on different items. In some cases, service providers likewise make utilization of this to trap the client for activation of some paid services.

Email Spam: Email spam is the most well-known type of spam. Email spam focuses on the individual clients with direct emails. Spammers make a rundown of email clients by inspecting Usenet postings, stealing lists of web mail, search the web for email addresses. Email spam costs cash to a client of email in light of the fact that while the client is perusing the messages meter is running. Email spam additionally costs the ISPs on the grounds that when a majority of spam sends are sent to the email clients its waste the bandwidth of the service providers these expenses are transmitted to clients. All undesirable emails are not spammed messages.

1.4 APPROACHES OF SPAM DETECTION

There are currently different approaches to spam detection. These approaches include blacklisting, detecting bulk emails, scanning message headings, grey listing, and content-based filtering

Blacklisting is a technique that identifies IP addresses that send large amounts of spam. These IP addresses are added to a Domain Name System-

Based Blackhole List and future email from IP addresses on the list are rejected. However, spammers are circumventing theselists by using larger numbers of IP addresses.

Detecting bulk emails is another way to filter spam. This method uses the number of recipients to determine if an email is spam or not. However, many legitimate emails can have high traffic volumes.

Scanning message headings is a fairly reliable way to detect spam. Program written by spammers generate headings of emails. Sometimes, these headings have errors that cause them to not fit standard heading regulations. When these headings have errors, it is a sign that the email is probably spam. However, spammers are learning from their errors and making these mistakes less often

Greylisting is a method that involves rejecting the email and sending an error message back to the sender. Spam programs will ignore this and not resend the email, while humans are more likely to resend the email. However, this process is annoying to humans and is not an ideal solution.

1.5 APPLICATIONS OF SPAM DETECTION

Spam Detection is now widely used in many fields such as Hospitals, Banks, Online Messaging platforms and in many business applications.

Every place where there is potential of spam being carried out is using Spam Detection Algorithms to provide security and better facility to their customers.

1.6 ADVANTAGES

- No theft of personal information.
- Avoidance of unwanted promotions.
- Avoidance of not participating in online fraud.
- Notified about phising sites.

1.7 DISADVANTAGES

Spam detection has more advantages than disadvantages. The disadvantages are negligible compared to the benefits it possess.

Disadvantages are as follows:

- Software cost of buying spam filters.
- Maintanence and updation cost.
- May require special hardware dedicated to run the software if information is too huge.

2. LITERATURE SURVEY

2.1. A Study on Email Spam Filtering Techniques [1]:

Electronic mail is used daily by millions of people to communicate around the globe and is a mission-critical application for many businesses. Over the last decade, unsolicited bulk email has become a major problem for email users. An overwhelming amount of spam is flowing into users‟ mailboxes daily. Not only is spam frustrating for most email users, it strains the IT infrastructure of organizations and costs businesses billions of dollars in lost productivity. The necessity of effective spam filters increases. In this paper, we presented our study on various problems associated with spam and spam filtering methods, techniques.

2.2. Machine Learning Methodsfor Spam E-MAIL Classification [2]:

The increasing volume of unsolicited bulk e-mail (also known as spam) has generated a need for reliable anti-spam filters. Machine learning techniques now days used to automatically filter the spam e-mail in a very successful rate. In this paper we review some of the most popular machine learning methods (Bayesian classification, k-NN, ANNs, SVMs, Artificial immune system and Rough sets) and of their applicability to the problem of spam Email classification. Descriptions of the algorithms are presented, and the comparison of their performance on the SpamAssassin spam corpus is presented.

2.3. Detecting Online Spams through Supervised Learning Techniques [3]:

With more customers utilizing on the online review surveys to educate their administration basic leadership, assessment of reviews which economically affect the reality of organizations. Obviously, crafty people or gatherings have endeavored to manhandle or control online review spam to make benefits, etc, and that tricky recognition and counterfeit sentiment surveys is a subject of

continuous research intrigue. In this paper, we clarify how supervised learning strategies are utilized to recognize online spam review surveys, preceding showing its utility utilizing an informational index of lodging reviews. Keywords- online review surveys, supervised learning, unlabeled data, Naïve bayes algorithm, classifiers, EM algorithm, Bag of Words, Stop word Filtering, Support Vector Machine Classifier.

2.4. A Case for Unsupervised-learning-based Spam Filtering [4]:

Spam filtering has traditionally relied on extracting spam signatures via supervised learning, i.e., using emails explicitly manually labeled as spam or ham. Such supervised learning is labor-intensive and costly, more importantly cannot adapt to new spamming behavior quickly enough. The fundamental reason for needing labeled training corpus is that the learning, e.g., the process of extracting signatures, is carried out by examining individual emails. In this paper, we study the feasibility of unsupervised learning-based spam filtering that can more effectively identify new spamming behavior. Our study is motivated by three key observations of today's Internet spam: (1) the vast majority of emails are spam, (2) a spam email should always belong to some campaign, (3) spam from the same campaign are generated from some template that obfuscates some parts of the spam, e.g., sensitive terms, leaving other parts unchanged. We present the design of an online, unsupervised spam learning and detection scheme. The key component of our scheme is a novel text-mining-based campaign identification framework that clusters spam into campaigns and extracts the invariant textual fragments from spam as campaign signatures. While the individual terms in the invariant fragments can also appear in ham, the key insight behind our unsupervised scheme is that our learning algorithm is effective in extracting co-occurrences of terms that are generated by campaign templates and rarely appear in ham. Using large traces containing about 2 million emails from three sources, we show our unsupervised scheme alone achieves a false negative ratio of 3.5% and a false positive ratio of at most 0.4%. These detection accuracies are comparable to

those of the de-facto supervised-learning-based filtering systems such as SpamAssassin (SA), suggesting that unsupervised spam filtering holds high promise in battling today's Internet Spam.

2.5. Machine learning for email spam filtering: review, approaches and open research problems [5]:

The upsurge in the volume of unwanted emails called spam has created an intense need for the development of more dependable and robust antispam filters. Machine learning methods of recent are being used to successfully detect and filter spam emails. We present a systematic review of some of the popular machine learning based email spam filtering approaches. Our review covers survey of the important concepts, attempts, efficiency, and the research trend in spam filtering. The preliminary discussion in the study background examines the applications of machine learning techniques to the email spam filtering process of the leading internet service providers (ISPs) like Gmail, Yahoo and Outlook emails spam filters. Discussion on general email spam filtering process, and the various efforts by different researchers in combating spam through the use machine learning techniques was done. Our review compares the strengths and drawbacks of existing machine learning approaches and the open research problems in spam filtering. We recommended deep leaning and deep adversarial learning as the future techniques that can effectively handle the menace of spam emails.

2.6.A Bayesian Approach to Filtering Junk E-Mail [6]:

In addressing the growing problem of junk E-mail on the internet, we examine methods for the automated construction of filters to eliminate such unwanted messages from a user's mail stream. By casting the problem in a decision theoretic framework, we are able to make use of probabilistic learning methods in conjunction with a notion of differential misclassification cost to produce filters which are especially appropriate for the nuances of this task. While this may appear, at first, to be a straight-forward text classification problem, we

show that by considering domain-specific features of this problem in addition to the raw text of E-mail messages, we can produce much more accurate filters. Finally, we show the efficacy of such filters in a real world usage scenario, arguing that this technology is mature enough for deployment.

2.7. Email Spam Detection using Extended KNN algorithm [7]:

E-mail is the cheaper and fast way of communication. E-mail is used in both personal and professional levels of life. Various types of e-mail are lies on social websites. The spam is one of them. Spam is the undesired messages on the internet site which is nothing but wastes the time and resources. Spam messages are sent by the spammer for marketing, promotion, spreading the virus. Various detection and filtering approaches are used to manage the spam. One is the most useful and simple approach is KNN algorithm which is content based approach. In this paper, the authors are trying to improve KNN algorithm which can be later used for better Spam Email Detection.

2.8. Spam Detection filter using KNN algorithm and resampling [8]:

Spamming has become a time consuming and expensive problem for which several new directions have been investigated lately. This paper presents a new approach for a spam detection filter. The solution developed is an offline application that uses the k-Nearest Neighbour (KNN) algorithm and a pre-classified email data set for the learning process.

2.9. Improved Bayesian Anti-Spam filter Implementation and Analysis on Independent Spam Corpuses [9]:

Spam emails are causing major resource wastage by unnecessarily flooding the network links. Though many anti-spam solutions have been implemented, the Bayesian spam score approach looks quite promising. A proposal for spam detection algorithm is presented and its implementation using Java is discussed, along with its performance test results on two independent spam corpuses - Ling-spam and Enron-spam. We use the Bayesian calculation for

single keyword sets and multiple keywords sets, along with its keyword contexts to improve the spam detection and thus to get good accuracy.

2.10. An Integrated approach for Malicious Tweets Detection using NLP [10]:

Many previous works have focused on detection of malicious user accounts. Detecting spams or spammers on Twitter has become a recent area of research in social network. However, we present a method based on two new aspects: the identification of spam-tweets without knowing previous background of the user; and the other based on analysis of language for detecting spam on twitter in such topics that are in trending at that time. Trending topics are the topics of discussion that are popular at that time. This growing micro blogging phenomenon therefore benefits spammers. Our work tries to detect spam tweets in based on language tools. We first collected the tweets related to many trending topics, labelling them on the basis of their content which is either malicious or safe. After a labelling process we extracted a manyfeature based on the language models using language as a tool. We also evaluate the performance and classify tweets as spam or not spam. Thus, our system can be applied for detecting spam on Twitter, focusing mainly on analysing of tweets instead of the user accounts.

2.11. E-Mail Spam Detection and Classification using SVM and Feature Extraction [11]:

Today emails have become to be a standout amongst the most well-known and efficient types of correspondence for Internet clients. Hence because of its fame, the email will be misused. One such misuse is the posting of unwelcome, undesirable messages known as spam or junk messages. Email spam has different consequences. It diminishes productivity, consumes additional space in mailboxes, additional time, expands programming damaging viruses, and materials that contain conceivably destructive data for Internet clients, destroys the stability of mail servers, and subsequently, clients invest lots of time for sorting approaching mail and erasing undesirable correspondence. So

there is a need for spam detection so that its outcomes can be reduced. In this paper, propose a novel method for email spam detection using SVM and feature extraction which achieves an accuracy of 98% with the test datasets.

3 SPAM DETECTION APPROACHES

3.1UNSUPERVISED LEARNING

Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a data set with no pre-existing labels and with a minimum of human supervision. In contrast to supervised learning that usually makes use of human-labelled data, unsupervised learning, also known as self-organization allows for modelling of probability densities over inputs. It forms one of the three main categories of machine learning, along with supervised and reinforcement learning. Semi-supervised learning, a related variant, makes use of supervised and unsupervised techniques.

3.1.1 CLUSTERING

“Clustering” is the process of grouping similar entities together. The goal of this unsupervised machine learning technique is to find similarities in the data point and group similar data points together. Grouping similar entities together help profile the attributes of different groups. In other words, this will give us insight into underlying patterns of different groups. There are many applications of grouping unlabeled data, for example, you can identify different groups/segments of customers and market each group in a different way to maximize the revenue. Another example is grouping documents together which belong to the similar topics etc.

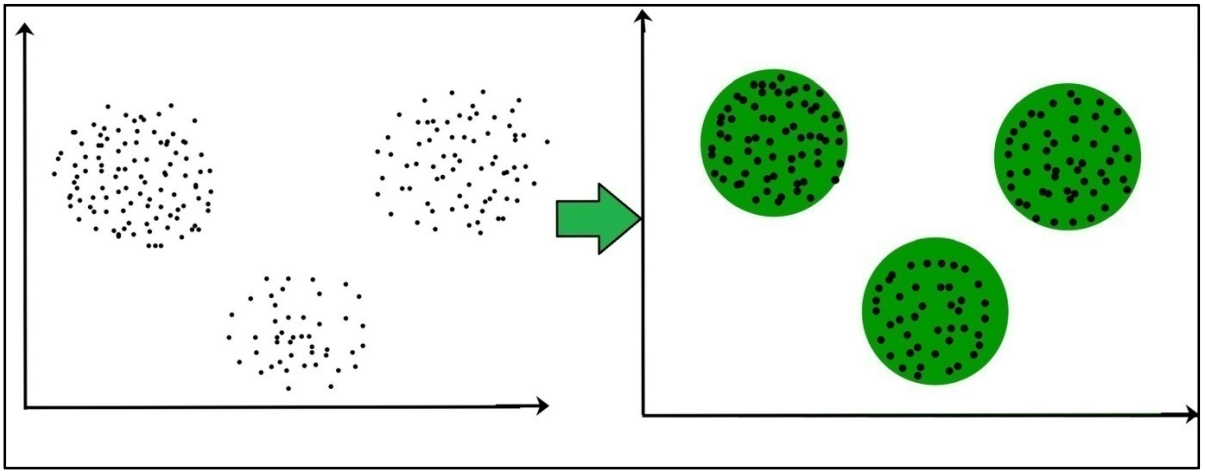


Fig 3.1 CLUSTERING – GROUPING SIMILAR OBJECTS

Clustering is also used to reduces the dimensionality of the data when you are dealing with a copious number of variables. There are many algorithms developed to implement this technique but the most popular and widely used algorithms in machine learning.

- K-mean Clustering
- Hierarchical Clustering

3.1.2 ASSOCIATION

Association rules or association analysis is also an important topic in data mining. This is an unsupervised method, so we start with an unlabeled dataset. An unlabeled dataset is a dataset without a variable that gives us the right answer. Association analysis attempts to find relationships between different entities. The classic example of association rules is market basket analysis. This means using a database of transactions in a supermarket to find items that are bought together. For example, a person who buys potatoes and burgers usually buys pizza. This insight could be used to optimize the supermarket layout. Online stores are also a good example of association analysis. They usually suggest to you a new item based on the items you have bought. They analyze online transactions to find patterns in the buyer's

behaviour. These algorithms assume all variables are categorical; they perform poorly with numeric variables. Association methods need a lot of time to be completed; they use a lot of CPU and memory.

Suppose we have a dataset such as the following: Our objective is to discover items that are purchased together. We'll create rules and we'll represent these rules like this:

Chicken, Potatoes → Clothes

This rule means that when a customer buys Chicken and Potatoes, he tends to buy Clothes. As we'll see, the output of the model will be a set of rules. We need a way to evaluate the quality or interest of a rule. There are different measures, but we'll use only a few of them. Rattle provides three measures:

Support

Confidence

Support indicates how often the rule appears in the whole dataset. In our dataset, the rule Chicken, Potatoes → Clothes has a support of 48.57 percent (3 occurrences / 7 transactions).

Confidence measures how strong rules or associations are between items. In this dataset, the rule Chicken, Potatoes → Clothes has a confidence of 1. The items Chicken and Potatoes appear three times in the dataset and the items Chicken, Potatoes, and Clothes appear three times in the dataset; and 3/3 = 1. A confidence close to 1 indicates a strong association.

3.2 SUPERVISED LEARNING

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and the desired output value (also called the supervisory

signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.

Steps involved in the supervised learning approach :

- Determine the type of training examples.
- Gather a training set. The training set needs to be representative of the real-world use of the function.
- Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output.
- Determine the structure of the learned function and corresponding learning algorithm. For example, the engineer may choose to use support vector machines or decision trees.
- Complete the design. Run the learning algorithm on the gathered training set. Some supervised learning algorithms require the user to determine certain control parameters. These parameters may be adjusted by optimizing performance on a subset (called a validation set) of the training set, or via cross-validation.
- Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set. Training data already trained.

3.3 REGRESSION ANALYSIS

Regression analysis is a reliable method of identifying which variables have an impact on a topic of interest. The process of performing a regression allows you to confidently determine which factors matter most, which factors can be ignored, and how these factors influence each other.

In order to understand regression analysis fully, it's essential to comprehend the following terms:

- Dependent Variable: This is the main factor that you're trying to understand or predict.
- Independent Variables: These are the factors that you hypothesize have an impact on your dependent variable.

The following are the few Regression Analysis models :

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the

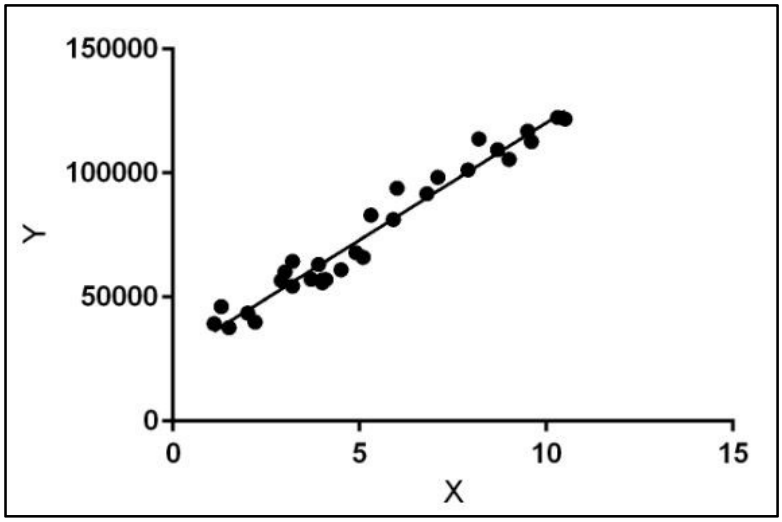


Fig 3.3 LINEAR REGRESSION – (Relationship Between Dependent(Y) And Independent Variables(X))

relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between the dependent and independent variables, they are considering and the number of independent variables being used.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.In the figure above, X (input) and Y (output) . The regression line is the best fit line for our model.

3.4 Classification:

Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories. The classification predictive modeling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into.

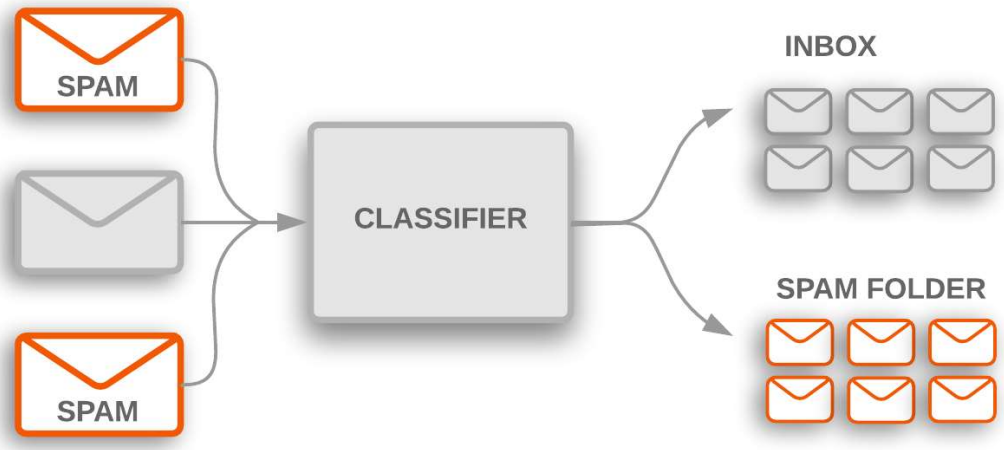


Fig 3.4 CLASSIFICATION OF MAILS INTO TWO CLASSES

Basic Classification Glossary

- **Classifier** – It is an algorithm that is used to map the input data to a specific category.
- **Classification Model** – The model predicts or draws a conclusion to the input data given for training, it will predict the class or category for the data.
- **Feature** – A feature is an individual measurable property of the phenomenon being observed.
- **Binary Classification** – It is a type of classification with two outcomes, for eg – either true or false.
- **Multi-Class Classification** – The classification with more than two classes, in multi-class classification each sample is assigned to one and only one label or target.
- **Multi-label Classification** – This is a type of classification where each sample is assigned to a set of labels or targets.
- **Initialize** – It is to assign the classifier to be used for the
- **Train the Classifier** – Each classifier in sci-kit learn uses the fit(X, y) method to fit the model for training the train X and train label y.
- **Predict the Target** – For an unlabeled observation X, the predict(X) method returns predicted label y.
- **Evaluate** – This basically means the evaluation of the model i.e classification report, accuracy score, etc.

3.5 DECISION TREE

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks.

The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The decision rules are generally in the form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.

A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers the to the question, and the leaves represent the actual output or class label. They are used in non-linear decision making with simple linear decision surface. Decision trees classify the examples by sorting them down the tree from the root to some leaf node, with the leaf node providing the classification to the example. Each node in the tree acts as a test case for some attribute, and each edge descending from that node corresponds to one of the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new nodes.

The Basic Decision Tree is shown in the below diagram

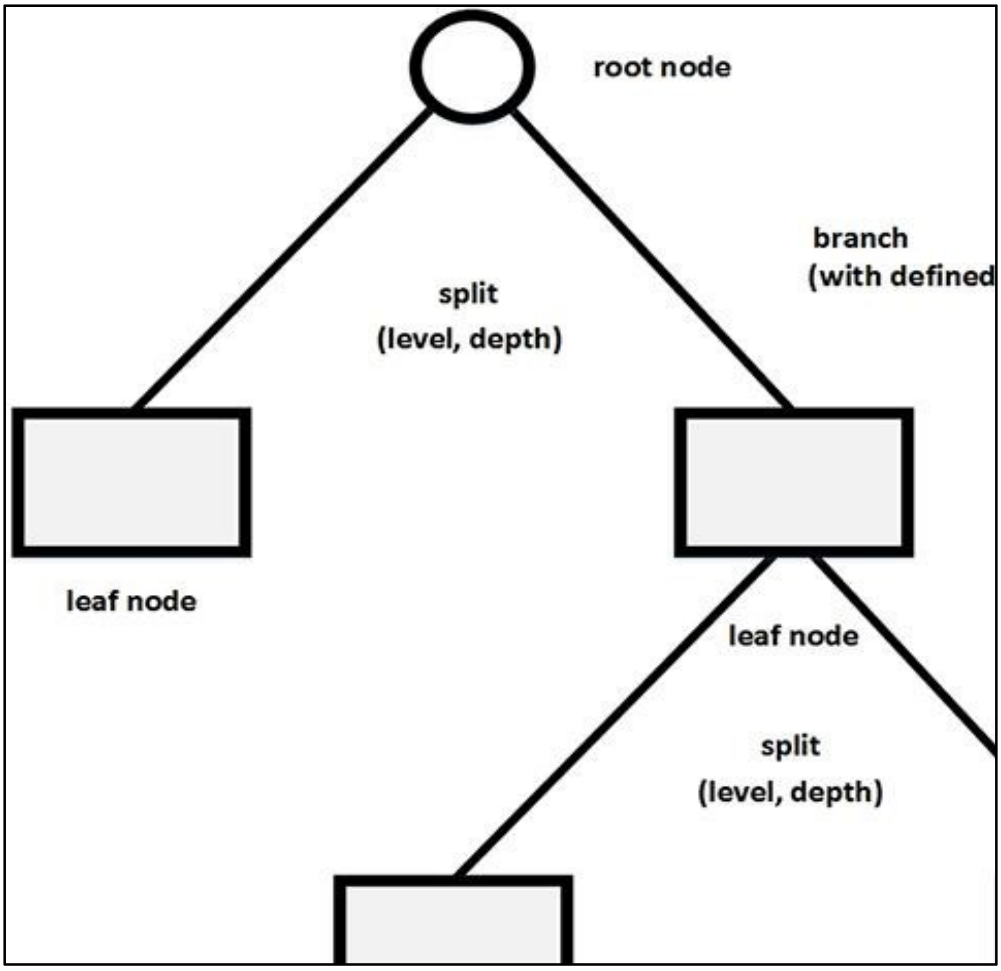


Fig 3.5 Basic Parts of Decision Tree

3.5.1 ADVANTAGES

Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.

A decision tree does not require normalization of data.

A decision tree does not require scaling of data as well.

Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.

A Decision trees model is very intuitive and easy to explain to technical teams as well as stakeholders.

3.5.2 DISADVANTAGES

A small change in the data can cause a large change in the structure of the decision tree causing instability.

For a Decision tree sometimes calculation can go far more complex compared to other algorithms.

Decision tree often involves higher time to train the model.

Decision tree training is relatively expensive as complexity and time taken are more.

Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

3.6 SUPPORT VECTOR MACHINE

In machine learning, support-vector machines (SVMs, also support-vector networks[]) are supervised learning models with associated learning algorihms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier . In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. We have to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin

distance provides some reinforcement so that future data points can be classified with more confidence.

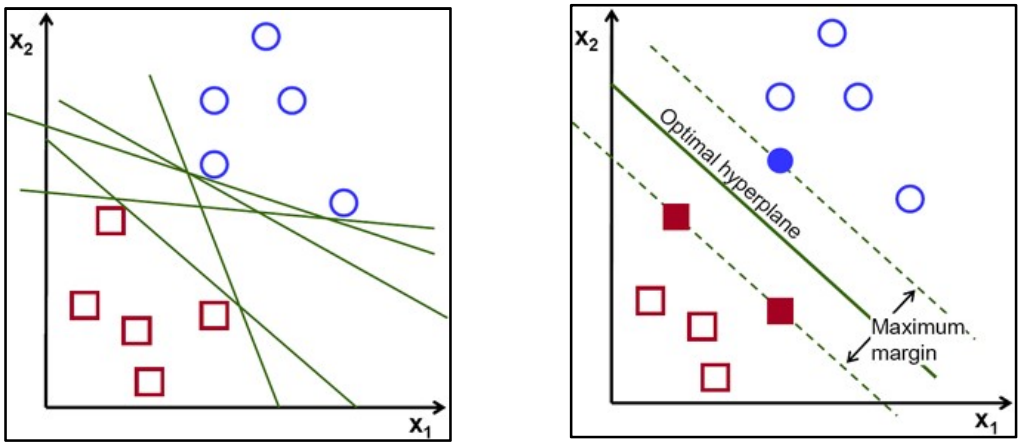


Fig 3.6.1 SELECTIING THE OPTIMAL HYPER-PLANE THAT MAXIMIZES THE MARGIN

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different

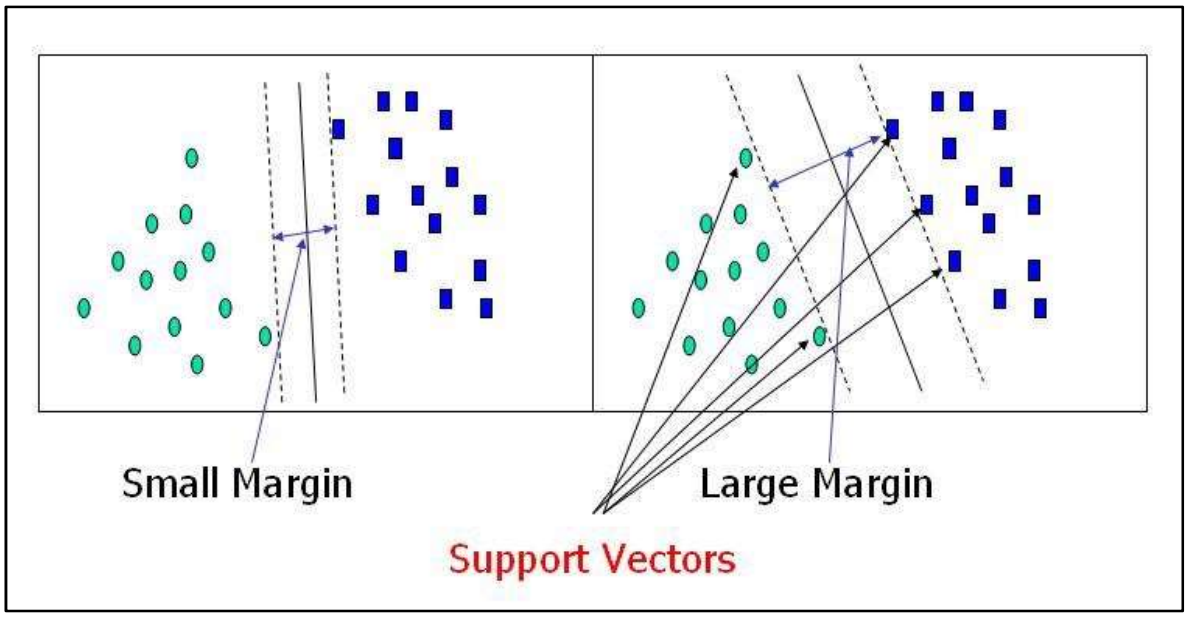


Fig 3.6.2 Image showing support vectors with small & large Margins

classes. Also, the dimension of the hyperplane depends upon the number of

features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

Support vectors are data points that are closer to the hyper-plane and influence the position and orientation of the hyper-plane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyper-plane. These are the points that help us build our SVM.

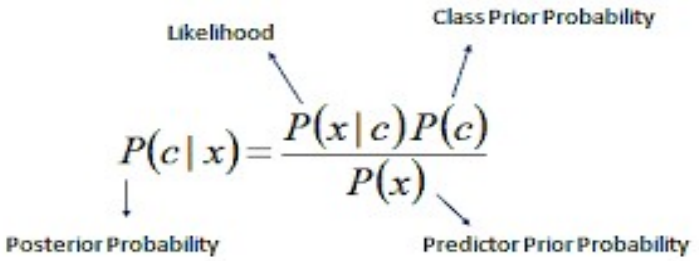
3.7 NAIVE BAYES ALGORITHM

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Fig 3.7 'NAÏVE BAYESIAN' FORMULA TO CALCULATE PROBABILTY

Where ,

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

3.7.1 ADVANTAGES

- The naive Bayesian model originated from classical mathematical theory and has a solid mathematical foundation and stable classification efficiency.
- It has a higher speed for large numbers of training and queries.Even with very large training sets, there is usually only a relatively small number of features for each project, and the training and classification of the project is only a mathematical operation of the feature probability;
- It works well for small-scale data, can handle multi-category tasks, and is suitable for incremental training (that is, it can train new samples in real time);

- Less sensitive to missing data, the algorithm is also relatively simple, often used for text classification;
- Naïve Bayes explains the results easily.

3.7.2 DISADVANTAGES

- Need to calculate the prior probability;
- There is an error rate in the classification decision;
- Very sensitive to the form of input data;
- due to the assumption of sample attribute independence is used, so if the sample attributes are related, the effect is not good.

4. SOFTWARE REQUIREMENT ANALYSIS

Spam detection for a regular computer commences with an Internet Service Provider (ISP) such as AT&T, Cox Cable, etc. They use refined software on their email servers to instantly catch spam, thus attempting to prevent the spam from ever reaching the individual.

Anti-spam software is included by several ISPs offering email accounts for their users. However, it is not as robust as that needed by individuals. This is where anti-spam software plays a major role. This software functions from the email program, whether that is Outlook, Gmail, or various other programs. With anti-spam software, emails that have suspicious content are flagged and then immediately sent into a spam folder, instead of going into the regular inbox. These emails are thus set aside for later investigation.

FUNCTIONAL REQUIREMENTS

- Classification of messages as records vs non records
- Classification of messages according to an organizational classification
- structure such as taxonomy or retention schedule
- Retention of messages throughout the life cycle,however the life cycle is defined.
- Disposition of messages at the end of the life cycle
- Tamper proof storage of messages, attachments and meta data, particularly for those declared as records
- Integration of the formal records retention schedule into the client or messaging application, such that users either can or must declare messages as records

NON FUNCTIONAL REQUIREMENTS

- Ensure high availability of email data sets
- Users should get the result as fast as possible
- It should be easy to use i.e,user is just required to type and click then the result is displayed or user is just required to enter a pair of reasonable sentence.

5. SYSTEM REQUIREMENTS

System requirements are the required specifications a system must have in order to perform the required tasks.

The following are the hardware and software requirements a system must possess in order to perform the tasks

Software requirements:

- Operating System: Spam Filter ISP will run on Windows NT, Windows 2000, Windows XP, Windows 2003, Windows 7, Windows 2008, Windows 2012 and Windows 2016 (GUI Required) and on any OS that support python as a programming language.
- Python 3.5: The programming language for the implementation of the machine learning classifier.
- Anaconda IDE: An Inntegrated IDE used for executing the model which supports wide range of libraries and useful for data visulaization.

Hardware requirements*:*

- Spam Filter is very CPU and RAM efficient. Server requirements will depend on the email traffic. VMWare virtual servers are also supported. Sample hardware configurations are as follows:
- For a server handling 20,000 emails/day, a 500MHz CPU and 512MB of RAM is the minimum recommended.
- For a server handling 200,000 emails/day, a dual-core 2GHz Xeon CPU and 4GBMB of RAM is the minimum recommended.
- For a server handling 2 million emails/day, two dual-core 3GHz Xeon CPU and 4GBMB of RAM is the minimum recommended.

- SpamFilter ISP - Optional quarantine database: Microsoft SQL Server 7 and higher (including SQL Azure), MySQL 4.0 and higher, Oracle 8 and higher, Microsoft Access 2000 and higher.
- SpamFilter Enterprise - One of the following database servers is required: Microsoft SQL Server 7 or higher (including SQL Azure) - MySQL 5 or higher (Unix / Windows / Mac are supported MySQL platforms)

Internet Requirements

- To fetch mails from G-mail servers for classification, Interent is required.

6. Methodology

Essentially, a methodology is a collection of methods, practices, processes, techniques, procedures, and rules. In project management,methodologies are specific, strict, and usually contain a series of steps and activities for each phase of the project's life cycle. They're defined approaches that show us exactly what steps to take next, the motivation behind each step, and how a project stage should be performed.

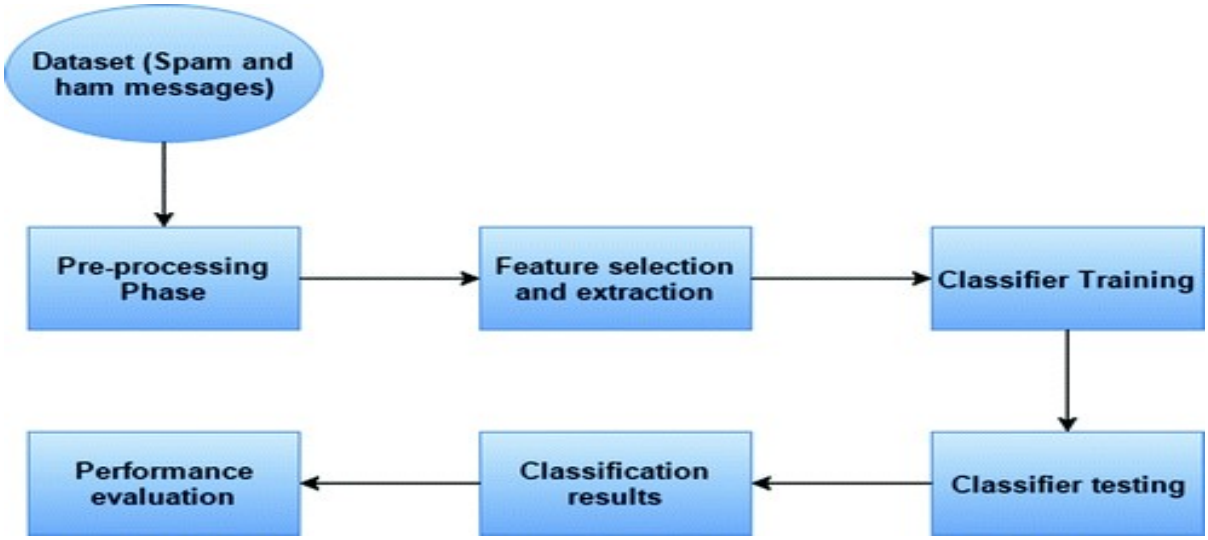


Fig 6 Text classification Steps

6.1 Dataset:

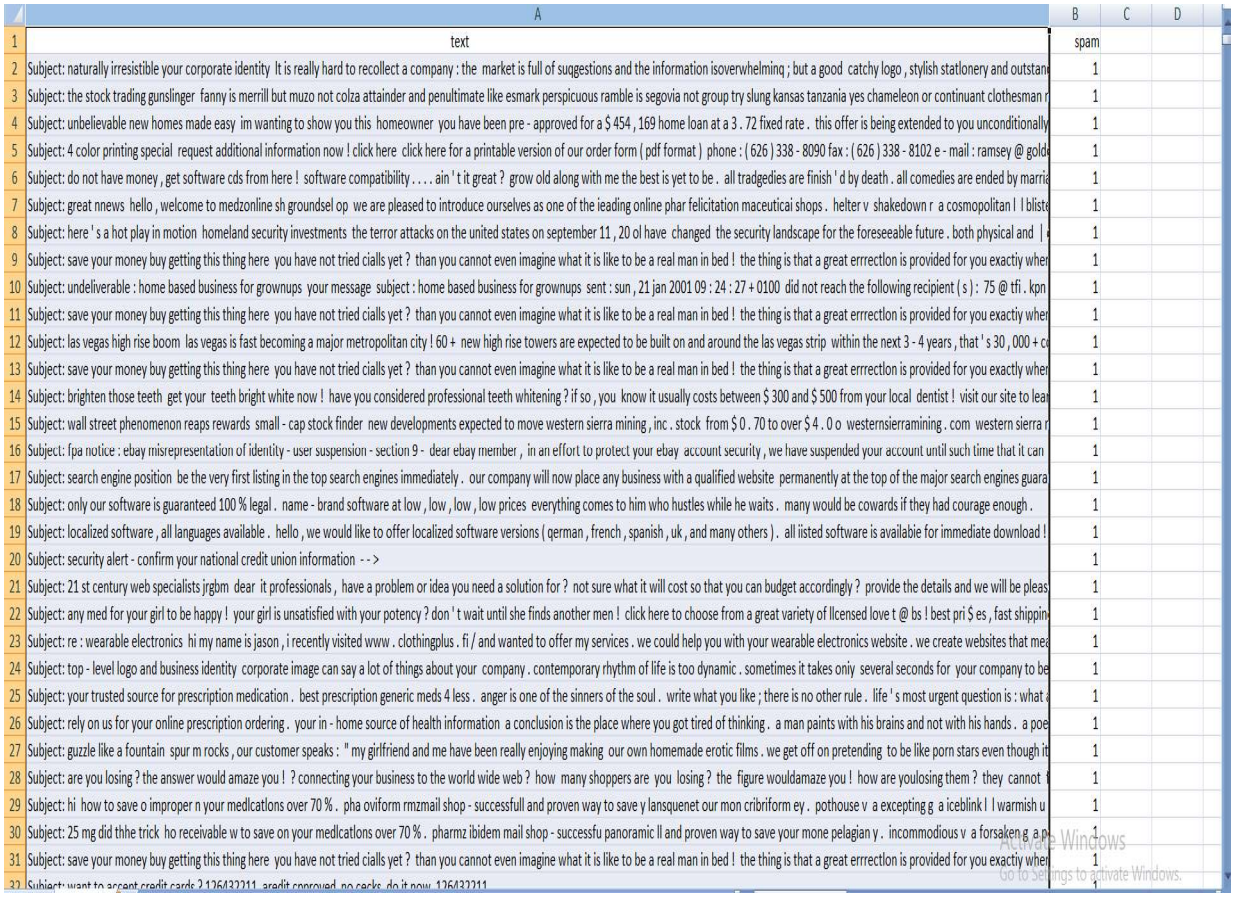
Two types of datasets are considered for this project. One dataset is a combination of the reported word frequencies that are identified in spam and ham mails in numerical format obtained from UCI Machine Learning Repository and the second dataset is collection of textual mails labelled as spam and ham obtained from kaggle.

Below figures shows how the datasets are:

1. Numerical dataset : Last attribute of each row is label 0 or 1 saying ham or spam.
2. Textual dataset : Second column of every row is label 0 or 1 saying ham or spam.


```
0,0.64,0.64,0,0.32,0,0,0,0,0,0,0.64,0,0,0,0.32,0,1.29,1.93,0,0.96,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.778,0,0,3.756,61,278,1
0.21,0.28,0.5,0,0.14,0.28,0.21,0.07,0,0.94,0.21,0.79,0.65,0.21,0.14,0.14,0.07,0.28,3.47,0,1.59,0,0.43,0.43,0,0,0,0,0,0,0,0,0,0,0,0,0.07,0,0,0,0,0,0,0,0,0,0,0,0,0.132,0,0.372,0.18,0.048,5.114,101,1028,1
0.06,0,0.71,0,1.23,0.19,0.19,0.12,0.64,0.25,0.38,0.45,0.12,0,1.75,0.06,0.06,1.03,1.36,0.32,0.51,0,1.16,0.06,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.06,0,0,0.12,0,0.06,0.06,0,0,0.01,0.143,0,0.276,0.184,0.01,9.821,485,2259,1
0,0,0,0,0.63,0,0.31,0.63,0.31,0.63,0.31,0.31,0.31,0,0,0.31,0,0,3.18,0,0.31,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.137,0,0.137,0,0,3.537,40,191,1
0,0,0,0,0.63,0,0.31,0.63,0.31,0.63,0.31,0.31,0.31,0,0,0.31,0,0,3.18,0,0.31,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.135,0,0.135,0,0,3.537,40,191,1
0,0,0,0,1.85,0,0,1.85,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.223,0,0,0,0,3,15,54,1
0,0,0,0,1.92,0,0,0,0,0.64,0.96,1.28,0,0,0,0.96,0,0.32,3.85,0,0.64,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.054,0,0.164,0.054,0,1.671,4,112,1
0,0,0,0,1.88,0,0,1.88,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.206,0,0,0,0,2.45,11,49,1
0.15,0,0.46,0,0.61,0,0.3,0,0.92,0.76,0.76,0.92,0,0,0,0,0,0.15,1.23,3.53,2,0,0,0.15,0,0,0,0,0,0,0,0,0.15,0,0,0,0,0,0,0,0,0,0.3,0,0,0,0,0,0,0.271,0,0.181,0.203,0.022,9.744,445,1257,1
0.06,0.12,0.77,0,0.19,0.32,0.38,0,0.06,0,0,0.64,0.25,0,0.12,0,0,0.12,1.67,0.06,0.71,0,0.19,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.06,0,0,0,0,0.04,0.03,0,0.244,0.081,0,1.729,43,749,1
0,0,0,0,0,0,0.96,0,0,1.92,0.96,0,0,0,0,0,0,0.96,3.84,0,0.96,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.96,0,0,0,0,0,0,0,0,0,0,0.462,0,0,1.312,6,21,1
0,0,0.25,0,0.38,0.25,0.25,0,0,0,0.12,0.12,0.12,0,0,0,0,0,1.16,0,0.77,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.022,0.044,0,0.663,0,0,1.243,11,184,1
0,0.69,0.34,0,0.34,0,0,0,0,0,0,0.69,0,0,0,0.34,0,1.39,2.09,0,1.04,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.056,0,0.786,0,0,3.728,61,261,1
0,0,0,0,0.9,0,0.9,0,0,0.9,0.9,0,0.9,0,0,0,0,0,2.72,0,0.9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2.083,7,25,1
0,0,1.42,0,0.71,0.35,0,0.35,0,0.71,0,0.35,0,0,0,5.35,0,0,3.21,0,2.85,0,0.35,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.102,0,0.357,0,0,1.971,24,205,1
0,0.42,0.42,0,1.27,0,0.42,0,0,1.27,0,0,0,0,0,1.27,0,0,1.7,0.42,1.27,0,0,0.42,0,0,0,0,0,0,0,0,0,0,0,0,0,1.27,0,0,0.42,0,0,0,0,0,0,0,0,0.063,0,0.572,0.063,0,5.659,55,249,1
0,0,0,0,0.94,0,0,0,0,0,0,0,0,0,0,0,0,0,1.88,0,2.83,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.428,0,0,4.652,31,107,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2.11,0,0.7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1.975,0.37,0,35.461,95,461,1
0,0,0.55,0,1.11,0,0.18,0,0,0,0,0,0.92,0,0.18,0,0.37,0.37,3.15,0,0.92,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.182,0,0.455,0,0,1.32,4,70,1
```


Fig 6.1.1 NUMERICAL DATASET CONTAINING 4601 MAILS, EACH WITH 58 ATTRIBUTES SPECIFYING PARTICULAR WORD



	A	B
1	text	spam
2	Subject: naturally irresistible your corporate identity It is really hard to recollect a company : the market is full of suqgestions and the information isoverwhelminq ; but a good catchy logo , stylish statlonery and outstan	1
3	Subject: the stock trading gunslinger fanny is merrill but muzo not colza attainder and penultimate like esmark perspicuous ramble is segovia not group try slung kansas tanzania yes chameleon or continuant clothesman r	1
4	Subject: unbelievable new homes made easy im wanting to show you this homeowner you have been pre - approved for a $ 454 , 169 home loan at a 3 . 72 fixed rate . this offer is being extended to you unconditionally	1
5	Subject: 4 color printing special request additional information now ! click here click here for a printable version of our order form (pdf format) phone : (626) 338 - 8090 fax : (626) 338 - 8102 e - mail : ramsey @ golde	1
6	Subject: do not have money , get software cds from here ! software compatibility . . . . ain ' t it great ? grow old along with me the best is yet to be . all tradgedies are finish ' d by death . all comedies are ended by marria	1
7	Subject: great nnews hello , welcome to medzonline sh groundsel op we are pleased to introduce ourselves as one of the ieading online phar felicitation maceuticai shops . helter v shakedown r a cosmopolitan l l bliste	1
8	Subject: here ' s a hot play in motion homeland security investments the terror attacks on the united states on september 11 , 20 ol have changed the security landscape for the foreseeable future . both physical and |	1
9	Subject: save your money buy getting this thing here you have not tried cialls yet ? than you cannot even imagine what it is like to be a real man in bed ! the thing is that a great errrectlon is provided for you exactiy wher	1
10	Subject: undeliverable : home based business for grownups your message subject : home based business for grownups sent : sun , 21 jan 2001 09 : 24 : 27 + 0100 did not reach the following recipient (s) : 75 @ tfi . kpn	1
11	Subject: save your money buy getting this thing here you have not tried cialls yet ? than you cannot even imagine what it is like to be a real man in bed ! the thing is that a great errrectlon is provided for you exactiy wher	1
12	Subject: las vegas high rise boom las vegas is fast becoming a major metropolitan city ! 60 + new high rise towers are expected to be built on and around the las vegas strip within the next 3 - 4 years , that ' s 30 , 000 + c	1
13	Subject: save your money buy getting this thing here you have not tried cialls yet ? than you cannot even imagine what it is like to be a real man in bed ! the thing is that a great errrectlon is provided for you exactly wher	1
14	Subject: brighten those teeth get your teeth bright white now ! have you considered professional teeth whitening ? if so , you know it usually costs between $ 300 and $ 500 from your local dentist ! visit our site to lea	1
15	Subject: wall street phenomenon reaps rewards small - cap stock finder new developments expected to move western sierra mining , inc . stock from $ 0 . 70 to over $ 4 . 0 o westernsierramining . com western sierra r	1
16	Subject: fpa notice : ebay misrepresentation of identity - user suspension - section 9 - dear ebay member , in an effort to protect your ebay account security , we have suspended your account until such time that it can	1
17	Subject: search engine position be the very first listing in the top search engines immediately . our company will now place any business with a qualified website permanently at the top of the major search engines guara	1
18	Subject: only our software is guaranteed 100 % legal . name - brand software at low , low , low , low prices everything comes to him who hustles while he waits . many would be cowards if they had courage enough .	1
19	Subject: localized software , all languages available . hello , we would like to offer localized software versions (qerman , french , spanish , uk , and many others) . all iisted software is availabie for immediate download !	1
20	Subject: security alert - confirm your national credit union information -->	1
21	Subject: 21 st century web specialists jrgbm dear it professionals , have a problem or idea you need a solution for ? not sure what it will cost so that you can budget accordingly ? provide the details and we will be pleas	1
22	Subject: any med for your girl to be happy ! your girl is unsatisfied with your potency ? don ' t wait until she finds another men ! click here to choose from a great variety of llcensed love t @ bs ! best pri $ es , fast shippin	1
23	Subject: re : wearable electronics hi my name is jason , i recently visited www . clothingplus . fi / and wanted to offer my services . we could help you with your wearable electronics website . we create websites that mea	1
24	Subject: top - level logo and business identity corporate image can say a lot of things about your company . contemporary rhythm of life is too dynamic . sometimes it takes oniy several seconds for your company to be	1
25	Subject: your trusted source for prescription medication . best prescription generic meds 4 less . anger is one of the sinners of the soul . write what you like ; there is no other rule . life ' s most urgent question is : what a	1
26	Subject: rely on us for your online prescription ordering . your in - home source of health information a conclusion is the place where you got tired of thinking . a man paints with his brains and not with his hands . a poe	1
27	Subject: guzzle like a fountain spur m rocks , our customer speaks : " my girlfriend and me have been really enjoying making our own homemade erotic films . we get off on pretending to be like porn stars even though it	1
28	Subject: are you losing ? the answer would amaze you ! ? connecting your business to the world wide web ? how many shoppers are you losing ? the figure wouldamaze you ! how are youlosing them ? they cannot	1
29	Subject: hi how to save o improper n your medlcatlons over 70 % . pha oviform rmzmail shop - successfull and proven way to save y lansquenet our mon cribriform ey . pothouse v a excepting g a iceblink l l warmish u	1
30	Subject: 25 mg did thhe trick ho receivable w to save on your medlcatlons over 70 % . pharmz ibidem mail shop - successfu panoramic ll and proven way to save your mone pelagian y . incommodious v a forsaken g a p	1
31	Subject: save your money buy getting this thing here you have not tried cialls yet ? than you cannot even imagine what it is like to be a real man in bed ! the thing is that a great errrectlon is provided for you exactiy wher	1
32	Subject: want to accept credit cards ? 126432211 , aredit approved, no cecks, do it now, 126432211	1

Fig 6.1.2 TEXTUAL DATASET WITH 5739 MAILS , LABELLED AS 1/0 (SPAM/HAM)

6.1.1 Numerical Dataset: spambase.csv

The dataset contains nearly 4601 mails with each mail containing 58 attributes each, in which the last attribute comes under label, which is only 0 or 1(0 represents spam and 1 represents ham).

Each of the 57 attributes is a word frequency that constitute to defining whether mail is spam or ham.

Those 57 words are :

- make, address, all ,3d ,our, over ,remove ,internet, order, mail ,receive, will ,people, report, addresses, free, business ,email ,you, credit, your,

font, 000 ,money, hp, hpl, George, 650, lab, labs, telnet, 857, data, 415, 85, technology, 1999, parts, pm, direct, cs, meeting, original, project, re, edu ,table ,conference , ; , (, [, !, $, #, _average, _longest, _total ,_label.

No pre-processing is required for this dataset since it is well structured and contains no useless information.

6.1.2 Textual Dataset: spam.csv

The textual dataset is collection of mails labelled as spam or ham in first column. This data is usually unstructured since each mail is of different length and some mails may contain unwanted symbols.

Pre-processing is required for this dataset.

6.2 Pre-processing Phase:

Pre-processing refers to the transformations applied to our

data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues.

Here, data preprocessing is performed on the data set for removing noise

in the data set that we are using here.

6.2.1 Removing punctuations and stop words:

In extracting features from the dataset , we may have stop words such as ‘a’,’an’,’the’,’is’ and so on in huge quantity which doesn’t help in classification so Removing punctuations and stop words is very crucial step to reduce our features dimensionality.

6.2.2 Splitting into training and testing sets:

In order to classify the data ,the dataset must be divided into two parts, test data(used to test) and train data(used as reference to test the test data). The given dataset is divided into test data and train data by taking the split percentage. Usually test data has less percent compared to train data, as the data used for reference should be more. Here in this project, we are taking split percentage as 0.25 for test data and 0.75 for train data. We split data into lists, X-train, Y-train, X-test, Y- test.

6.3 Feature Selection & Extraction

6.3.1 Numerical Dataset:

This dataset is already available with required features as 57 attributes showing particular word frequency corresponding to that mail.

6.3.2 Textual Dataset:

This dataset is unstructured so we need to extract features for classification purpose. The feature extraction is nothing but separating distinct and useful words with their frequencies in each mail. This is done by word vectorization and collecting them as bag of words.

6.4 Classifier Training:

The dataset splitted into train set is used for this Training phase. We calculate required statistical measures such as mean, standard deviation, or word probabilities etc.. on this training set which is later used for testing purpose.

6.5 Classifier Testing:

The information obtained from training phase such as mean, standard deviation , word frequencies or probabilities are used here to evaluate test set performance.

6.6 Classification Results:

The results obtained from testing phase are taken into consideration to finalize the results showing confusion matrix.

6.7 Performance Evaluation:

This phase is for testing new data that the model hasn't seen previously in its training phase or testing phase. Here we evaluate the model with real time data. For example: this project used G-mail integration to evaluate the model performance on real world G-mail data of a person.

7.TOOLS USED IN SPAM DETECTION

Spam detection refers to the use of text analysis, natural language processing, computational linguistics.There are plenty of tools that can support our requirements for performing spam detection. One of them that we use is ANACONDA.

7.1 ANACONDA

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system *conda*.The Anaconda distribution includes data-science packages suitable for Windows, Linux, and macOS.

7.1.1 ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab

- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code

The application we use here is the Jupyter Notebook.

7.1.2 JUPYTER NOTEBOOK

Jupyter Notebook is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides,LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell.

To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

8. SYSTEM DESIGN

8.1. USE CASE DIAGRAM

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities.

They also help identify any internal or external factors that may influence the system and should be taken into consideration.

They provide a good high-level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.

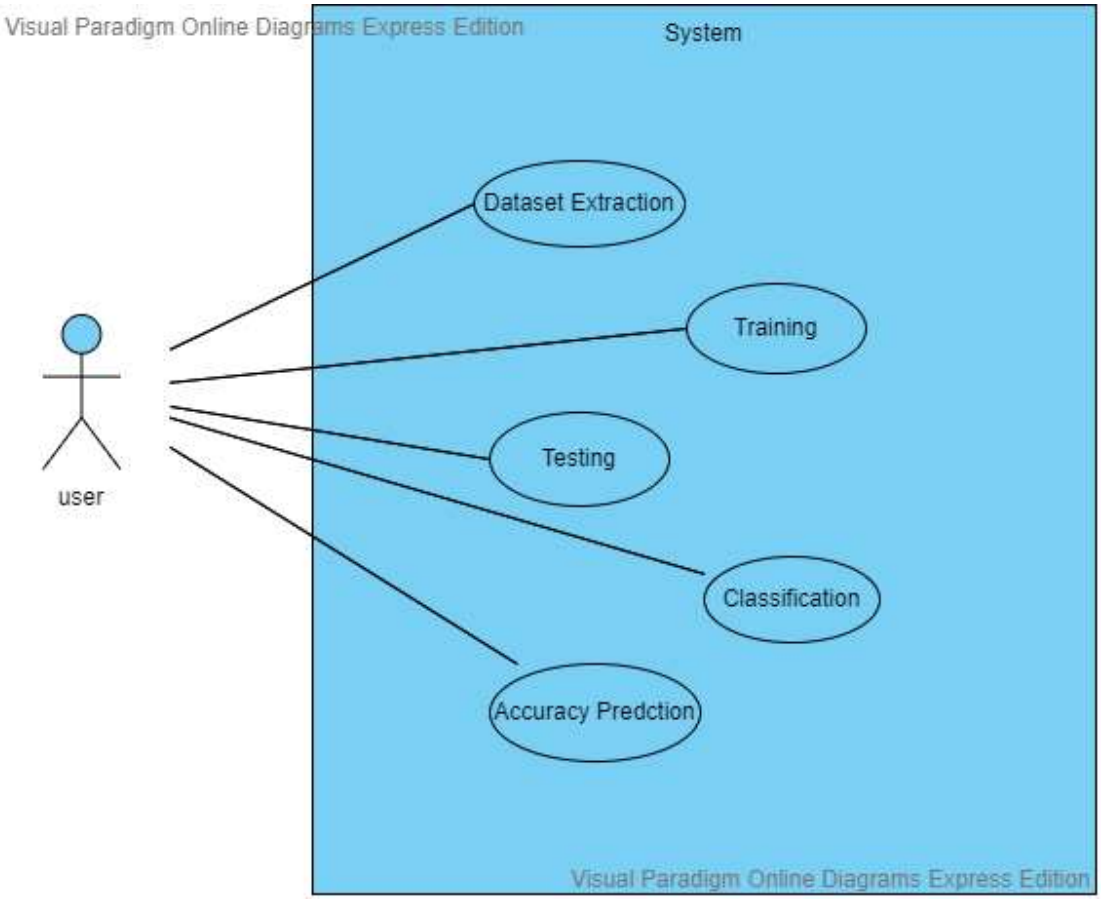


Fig 8.1 Use Case Diagram

8.2. SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

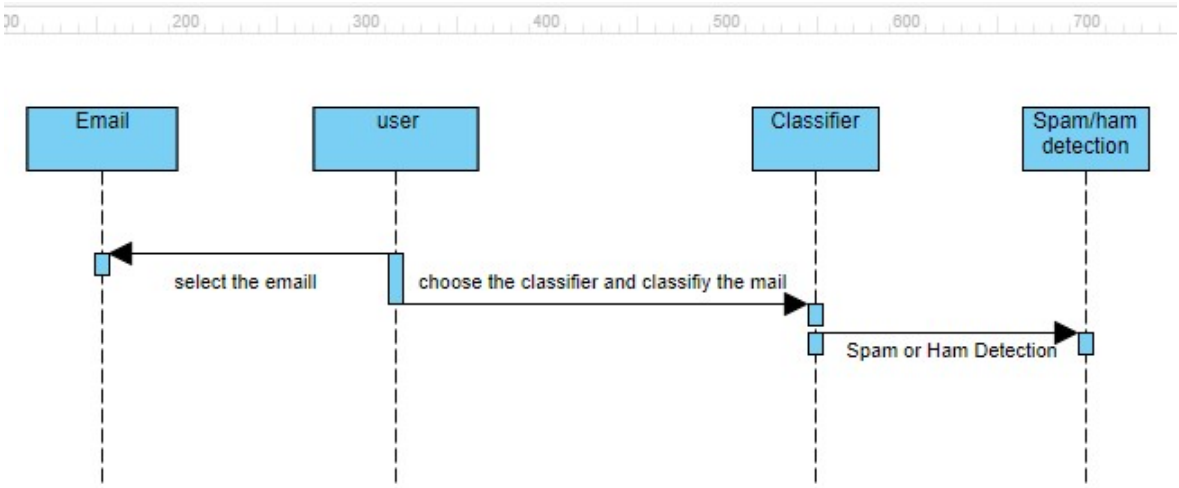


Fig 8.2 Sequence Diagram

8.3. ACTIVITY DIAGRAM

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

- UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behavior diagrams. An activity diagram is a behavioural diagram i.e. it depicts the behavior of a system.
- An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to

depict the dynamic aspects of the systems.

- An activity diagram is very similar to a flowchart.

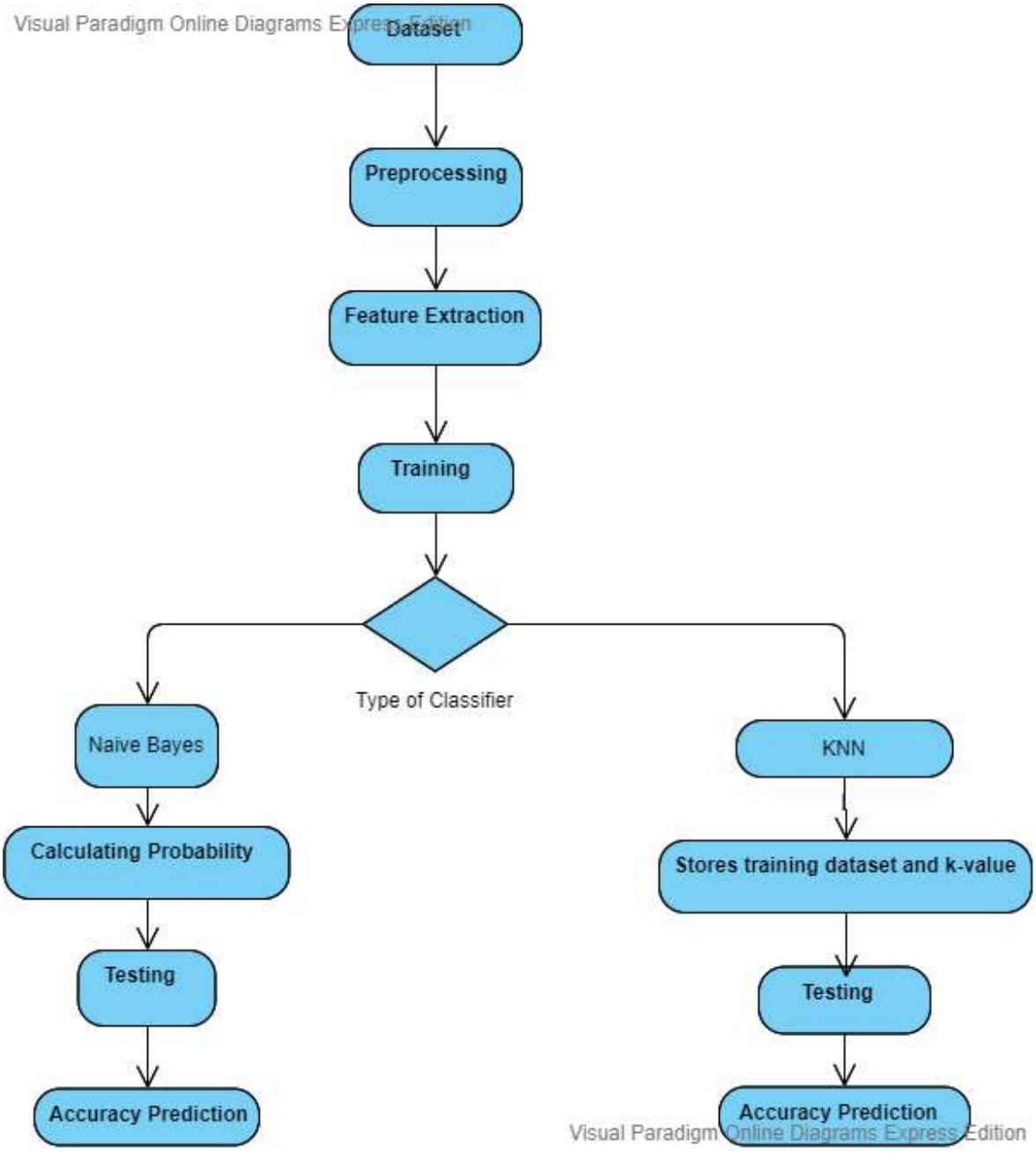


Fig 8.3 Activity Diagram

9.IMPLEMENTATION AND RESULTS

This project is mainly developed based on the numericals and text. This project used python as programming language. The classifier algorithms considered for classification are 1)Naive Bayes 2)K-Nearest Neighbours

The design process of our project is explained below

9.1 Importing required packages:

● **pandas:** To deal with the csv files. It is also useful in creating dataframes

which will allow to store and manipulate tabular data in rows of

observations and columns of variables.

● **numpy:** This is used for creating arrays and for performing vectorized operations efficiently.

● **sklearn:** Scikit-learn also called sklearn which features various

algorithms like Naive Bayes ,support vector machine, K-nearest neighbours,random forests etc..

● **nltk:** NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical sources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning

● **imaplib and smtplib:** These are used for accessing G-mail to retrieve or to send mails.

9.2 Pre-processing: Removing stop words and punctuations.

```
#Preprocessing the data by removing stop words and punctuations
from nltk.corpus import stopwords
stopwords_set=set(stopwords.words('english'))

def preProcess(dataset):
    with open(dataset,'r') as spam:
        data=list(csv.reader(spam))
        data=data[1:] #each mail is list of 2 strings.

        #making the data as list of a list with non punct words and list with label [['..','..',...],1/0]
        punct=set(string.punctuation)
        punct.remove('$')
        for email in range(len(data)):
            templist=[char for char in data[email][0] if char not in punct]
            templabel=int(data[email][1])
            data[email][0]=''.join(templist)
            data[email][1]=templabel

        #removing stop words and numbers from mails
        for email in range(len(data)):
            tempmail=[]
            for word in data[email][0].split():
                if(word not in stopwords_set and not word.isnumeric()):
                    tempmail.append(word)
            data[email][0]=' '.join(tempmail)

#         for i in range(100):
#             print(data[i])
        return data

data = preProcess(dataset)
for i in range(10):
```

Fig 9.2 Pre-processing - removing stop words & punctuations

9.3 Features Extraction:

This is carried out by extracting all distinct words along with their frequencies from entire dataset followed by fitting these word frequencies to each mail in the dataset.

Extracting distinct words step is called Count_Vectorization and fitting words with thier frequencies to each mail is called Fit & Transform.

9.3.1 CountVectorizer() and fit():

Imported from class sklearn.feature_extraction.text.CountVectorizer().

Convert a collection of texts to a matrix of tokens.

fit() is a method associated with CountVectorizer class which gives index to each word of dataset in alphabetical order.

```
vec=CountVectorizer()
bag = vec.fit(df['text'])
print(bag.vocabulary_)
```

```
{'subject': 32371, 'naturally': 23381, 'irresistible': 18838, 'your': 37024, 'corporate': 10045, 'identity': 17688, 'lt': 211
48, 'is': 18848, 'really': 28018, 'hard': 16655, 'to': 33798, 'recollect': 28142, 'company': 9281, 'the': 33450, 'market': 21
665, 'full': 15415, 'of': 24274, 'suqgestions': 32635, 'and': 4836, 'information': 18234, 'isoverwhelminq': 18885, 'but': 754
6, 'good': 16072, 'catchy': 8043, 'logo': 20960, 'stylish': 32352, 'statlonery': 32001, 'outstanding': 24871, 'website': 3608
0, 'will': 36373, 'make': 21440, 'task': 33068, 'much': 23052, 'easier': 12610, 'we': 36019, 'do': 12048, 'not': 23937, 'prom
ise': 27135, 'that': 33441, 'havinq': 16765, 'ordered': 24632, 'iogo': 18759, 'automaticaily': 5772, 'become': 6296, 'world':
36633, 'ieader': 17706, 'it': 18913, 'isguite': 18863, 'ciear': 8651, 'without': 36499, 'products': 27033, 'effective': 1281
3, 'business': 7529, 'organization': 24670, 'practicable': 26619, 'aim': 4345, 'be': 6249, 'hotat': 17363, 'nowadays': 23995,
'marketing': 21673, 'efforts': 12830, 'more': 22861, 'here': 16986, 'list': 20804, 'clear': 8826, 'benefits': 6430, 'creative
ness': 10282, 'hand': 16579, 'made': 21305, 'original': 24695, 'logos': 20963, 'specially': 31581, 'done': 12151, 'reflect':
28298, 'distinctive': 11932, 'image': 17812, 'convenience': 9896, 'stationery': 31992, 'are': 5274, 'provided': 27258, 'in':
17967, 'all': 4518, 'formats': 15085, 'easy': 12623, 'use': 35085, 'content': 9802, 'management': 21496, 'system': 32876, 'le
tsyou': 20553, 'change': 8332, 'even': 13831, 'its': 18942, 'structure': 32296, 'promptness': 27155, 'you': 37011, 'see': 302
```

Fig 9.3.1 IMAGE SHOWING FEATURE EXTRACTION USING COUNTVECTORIZER() CLASS

9.3.2 Transform():

Transform method transforms each mail to list of tuples indicating mail index and word indexes with their frequency in that particular mail.

Eg: (0,3972)1 . Here 0 indicates first mail, 3972 is index of some word and 1 indicates that word 3972 is repeated once in that mail.

```
sample_test_set=bag.transform(mails_list)
print(sample_test_set)
```

```
(0, 3972)     1
(0, 9128)     1
(0, 13832)    1
(0, 16072)    1
(0, 21301)    1
(0, 23203)    1
(0, 23442)    1
(0, 24292)    1
(0, 24563)    1
(0, 25975)    1
(0, 27929)    1
(0, 33557)    1
(1, 6642)     1
(1, 7882)     1
(1, 9619)     1
(1, 10291)    1
(1, 13803)    1
(1, 18234)    1
(1, 31595)    1
(1, 32371)    1
```

Fig 9.3.2 IMAGE SHOWING BAG OF WORDS REPRESENTATION TO HANDLE HUGE DIMENSIONS OF TEXT CLASSIFICATION EFFICIENTLY

9.4 Classification:

The project used KNN and Naive Bayes Algorithms to classify the two datasets considered.

9.4.1 KNN: First assume a K-value. K=7 considered best for this dataset.

9.4.1.1 Numerical dataset:

Training phase:

In KNN training phase is nothing but storing values in memory for further purpose. No additional calculations are required.

Testing phase:

euclid_dist(sample_ytrain,sample_xtrain): Calculates distance of each mail in test set with every mail in train set. Here sample_ytrain refers to test set.

nearest_neighbours(distances,k): This method based on chosen K value calculates k nearest neighbours

knn_predictlabel(knn): This method takes K nearest neighbours as input and labels them as spam or ham based on frequency of spam or ham mails in that k nearest neighburs list.

```
def euclid_dist(sample_ytrain,sample_xtrain):
    dist=0
    for i in range(len(sample_ytrain)):
        diff=(sample_ytrain[i]-sample_xtrain[i])
        dist+=diff*diff
    return math.sqrt(dist)

def nearest_neighbours(distances,k):
    knn=[]
    distances.sort(key=lambda x:x[0])
    for i in range(k):
        knn.append(distances[i])
    return knn

def knn_predictlabel(knn):
    onecount=0
    for i in knn:
        if(i[1]==1):
            onecount+=1
    if(onecount<=len(knn)-onecount):
        return 0
    else:
        return 1
```

Fig 9.4.1.1-1 Testing Phase of KNN in Numerical dataset

predictClasses (x_train,y_train,x_test,k): This method assigns predicted label to each mail test set.

```
def predictClasses(x_train,y_train,x_test,k):
    predicted_labels=[]
    for i in range(len(x_test)):
        distances=[]
        for j in range(len(x_train)):
            distances.append([euclid_dist(x_test[i],x_train[j]),y_train[j]])
        knn=nearest_neighbours(distances,k)
        #print(knn)
        predicted_labels.append(knn_predictlabel(knn))
    return predicted_labels
```

Fig 9.4.1.1-2 Predicting labels step of KNN model in numerical dataset

9.4.1.2 Textual dataset:

The training phase and testing phase is same as described in numerical dataset. What differs is that it uses classifier defined in sklearn .neighbors.KNeighborsClassifier

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score

knn_classifier=KNeighborsClassifier(n_neighbors=5).fit(x_train,y_train)
pred=knn_classifier.predict(x_test)
print(classification_report(y_test,pred))
```

Fig 9.4.1.2 Building KNN model in textual dataset

9.4.1.3 The problem of choosing Best K value:

Generally K is chosen in random to find best k value In K Nearest Neighbor algorithm. The probability of a mail being ham or spam does not change much by increasing the K-value. General strategy is taking K value as square root of test set length.

The below figure shows different probabilities associated with different k values in this project.

```
plt.plot(n_values_var,knn_accuracy_var,color='blue',marker='o')
plt.xlabel('K-Values')
plt.ylabel('Corresponding Accuracy')
plt.title('How Accuracy varying by value of K')
plt.show()
```

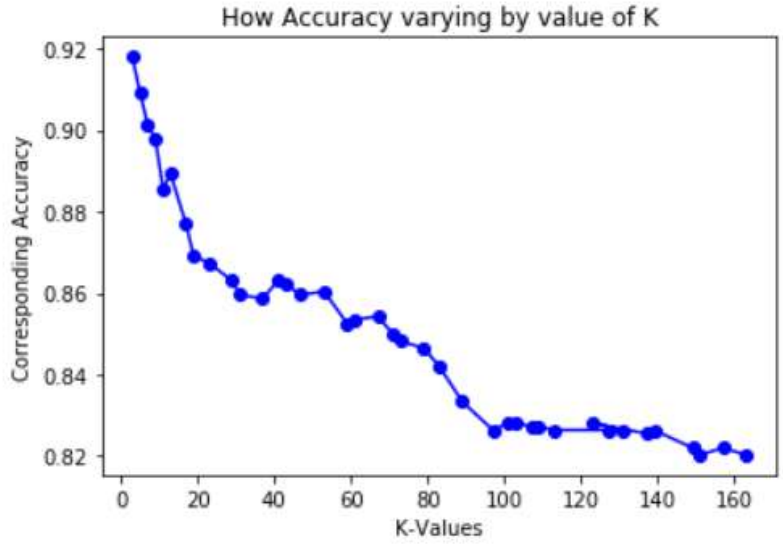


Fig 9.4.1.3 How accuracy is varying by changing k value

After a certain k value the accuracy becomes saturated. In the above figure k is ranged from 3 to 163 values associated accuracies are from 92% to 82 %. There is no certain K value as best to consider. Each project has its own best k value which is found by wither trail by error or by Cross validation.

9.4.2 Naive Bayes:

9.4.2.1 Numerical dataset:

Training phase:

In Naive Bayes training phase is summarizing the train set by calculating mean and standard deviation of each attribute.

```
def summarize(dataset):
    summaries = [(mean(attribute), stdev(attribute)) for attribute in zip(*dataset)]
    del summaries[-1]
    return summaries

def summarizeByClass(dataset):
    separated = separateByClass(dataset)
    #in separated lies all mail combined as lists under keys 0 and 1
    summaries = {}
    for classValue, instances in separated.items():
        summaries[classValue] = summarize(instances)
#In summaries all 57 attributes mean and std are present categorized under classValues
    return summaries
```

Fig 9.4.2.1 Training Phase of naïve bayes in Textual Dataset

Testing phase:

Here probabilities of test set mails being spam and ham are both calculated. Based on the highest probability being spam or ham, the mail is labeled.

```
def calculateClassProbabilities(summaries, inputVector):
    probabilities = {}
    for classValue, classSummaries in summaries.items():
        probabilities[classValue] = 1
        for i in range(len(classSummaries)):
            mean, stdev = classSummaries[i]
            x = inputVector[i]
            probabilities[classValue] *= calculateProbability(x, mean, stdev)
    return probabilities

def predict(summaries, inputVector):
    probabilities = calculateClassProbabilities(summaries, inputVector)
    #print(probabilities)

    bestLabel, bestProb = None, -1
    for classValue, probability in probabilities.items():
        #print(classValue,'->',probability)
        if bestLabel is None or probability > bestProb:
            bestProb = probability
            bestLabel = classValue
    return bestLabel
```

Fig 9.4.2.2 Testing Phase of naïve bayes in Textual Dataset

predict(summaries, inputVector) : This method assigns label to input vector based on which probability is higher(bestProb).

9.4.2.2 Textual dataset:

The training phase and testing phase are same as described in numerical dataset. What differs is that it uses classifier defined in sklearn .naive_bayes.MultinomialNB

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score

classifier=MultinomialNB().fit(x_train,y_train)

pred=classifier.predict(x_test)

```

Fig 9.4.2.2 Creating Naïve Bayes Model in Textual dataset

9.5 INTEGRATING WITH GMAIL

In order to accomplish the mail reading task we'll make use of the imaplib Python module. imaplib is a built in Python module, hence you don't need to install anything. You simply need to import the module.

9.5.1 How to Login to Gmail Using Python

You need three things to login to Gmail using Python. You'll need a mail server, a username and password. In this case, since we are trying to login to Gmail, our mail server would be either impa.gmail.com or smtp.gmail.com. If you are trying to read the incoming mail your incoming mail server would be imap.gmail.com and if you are trying to send mail then your outgoing mail server would be smtp.gmail.com. Hence, our mail server is imap.gmail.com. Username and password are your gmail username and password.

9.5.2 Fetching gmail emails from a particular user:

Fetching Gmail emails though is a tedious task but with Python, many things can be done if you are well versed with its usage. Gmail provides IMAP access to clients who want to access Gmail without manually logging in the browser.

Implementation:

The libraries used in this implementation includes imaplib, email. You have to manually go and make IMAP access enabled by going into your Gmail account settings . After this only you could access your Gmail account without logging in browser.

- Three functions are defined in the implementation which is used to get email body, search for emails from a particular user and get all emails under a label.
- For showing results I have sent email to my id from my another Gmail account. Now I will be fetching emails from my Gmail account which is received from my another Gmail account.
- The process begins from making Gmail connection with the help of *imaplib library* and proving our Gmail login credentials to it.
- After logging we are selecting emails under the label: Inbox which is a default labeled section for all users. However, you can create your own labels also.
- Then we are calling get emails function and provide it the parameter from search function result i.e “from user”
- In get emails function we are putting all emails in an array named “msgs”
- Now print to see the messages array

- Now we can easily iterate over this array. We are iterating it in the order the emails arrived. Then we are searching for the index from where our content begins. This indexing part will be different for different emails/users and the user can manually change the indexes to print only that part which they require.
- We have our results printed out.

integration with gmail

```
import os
import imaplib
import email
#from emailmessage import EmailMessage


myAcc=os.environ.get('myGmailAddr')
passw=os.environ.get('myGmailPass')

with imaplib.IMAP4_SSL('imap.gmail.com',993) as con:
    con.login(myAcc,passw)
    #con.select('"[Gmail]/Sent Mail"')
    con.select('Inbox')
    ty,data=con.search(None,'(FROM "virat.vamsi58@gmail.com")')
    mail_ids=data[0].split()
    latest_maild=mail_ids[-5]

    ty,data=con.fetch(latest_maild,'(RFC822)')
    raw_data=data[0][1]
    raw_string=raw_data.decode('utf8')
    msg=email.message_from_string(raw_string)

    text_msg=''
    #print(msg)
    for part in msg.walk():
        if(part.get_content_type()=='text/plain'):
            text_msg+=part.get_payload()
        if(part.get_content_type()!='multipart' and part.get('Content-Disposition') is not None):
            continue
print(text_msg)
```

Fig 9.5.2.1 Fetching Mail from gmail of a particular user

The above code fetches mails from Inbox of user ‘virat.vamsi58@gmail.com’.

The text obtained is further pre-processed and features are extracted to predict its label.

```
    #removing stop words and numbers from mails
    for email in range(len(gmail_list)):
        tempmail=[]
        for word in gmail_list[email].split():
            if(word not in stopwords_set and not word.isnumeric()):
                tempmail.append(word)
        gmail_list[email]=' '.join(tempmail)

mails_list=[text_msg,'Subject: Congratulations! You have won: You have won 20 billion euros, specify your credit card
labels=[0,1,1]
preProcess_gmails(mails_list)
for i in range(len(mails_list)):
    print(mails_list[i])
```

```
Subject Congratulations You won You won billion euros specify credit card information
Subject security alert confirm national credit union information
```

```
vec=CountVectorizer()
bag = vec.fit(df['text'])
#print(bag.vocabulary_)
sample_test_set=bag.transform(mails_list)
#print(sample_test_set)
```

```

knn_classifier=KNeighborsClassifier(n_neighbors=3).fit(x_train,y_train)
pred=knn_classifier.predict(sample_test_set)
print(pred)
print(classification_report(labels,pred))

print('confusion matrix :\n',confusion_matrix(labels,pred))
print('Accuracy :\n',accuracy_score(labels,pred))
```

```
[0 0 1]
              precision    recall  f1-score   support
```

Fig 9.5.2.2 Pre-processing extracted mail & building Knn model

The above figure shows how the mails are pre-processed by removing stop words and features are extracted using CountVectorizer() class and then knn classifier is applied to sample_test_set containing mails fetched from gmail to predicted labels.

10. FUTURE SCOPE AND CONCLUSION

In this Project, we are able to classify the emails as Spam or Ham using the Machine Learning Algorithms. Efficient pattern detection in spam filtering plays crucial role. We have used KNN and Naïve Bayes algorithm for E-mail Spam Detection. The idea was to improve the terms of parameters like accuracy, precision. Although there are many references for this classification type, we made it easy to access the E-mail directly from the mail provided.

Limitations of The Project:

Our Project, E-mail Spam Detection is capable of Text Classification only. Therefore, at this stage we are unable to classify images in the mails and the increase in misspellings in the E-mail may lead to the decrease in the Accuracy.

Future Enhancement:

There is a wide scope of enhancement in our project. Following enhancements can be done: Image Classification can be done on the basis of its contents. Furthermore, Misspellings can be classified on the basis of modules present in Python.

11. REFERENCES:

[1] Christina V, Karpagavalli S and Suganya G "A study on Email Spam Filtering Techniques"

[2] W.A. Awad and S.M. ELseuofi "Machine Learning Methods for Spam Email Classification"

[3] M.S Minu, Kamagiri Mounika, N.Suhasini and Bezawada Tejaswi "Detecting Online Spams through Supervised Learning Techniques"

[4] Feng Qian, Y. Charlie Hu and Z. Morley Mao "A Case for Unsupervised-learning-based Spam Filtering"

[5] Emmanuel Gbenga Dada, Joseph Stephen Bassi , Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi and Opeyemi Emmanuel Ajibuwa "Machine learning for email spam filtering: review, approaches and open research problems"

[6] Mehran Sahami, Susan Dumais, David Heckerman and Eric Horvitz "Bayesian Approach to Filtering Junk E-Mail"

[7] Ritu Saini and Er. Geetanjli Chawla "Email Spam Detection using Extended KNN algorithm"

[8] Loredana Firte, Camelia Lemnaru and Rodica Potolea “Spam Detection filter using KNN algorithm and resampling”

[9] Biju Issac, Wendy Japutra Jap and Jofry Hadi Sutanto “Improved Bayesian Anti-Spam filter Implementation and Analysis on Independent Spam Corpuses”

[10] Sagar Gharge and Manik Chavan “An Integrated approach for Malicious Tweets Detection using NLP”

[11] Shradhanjali and Prof. Toran Verma “E-Mail Spam Detection and Classification using SVM and Feature Extraction”

[12] Hanif Bhuiyan, Akm Ashiquzzaman, Tamanna Islam Juthi , Suzit Biswas and Jinat Ara “A Survey of Existing E-Mail Spam Filtering Methods Considering Machine Learning Techniques”

[13] Dr. Swapna Borde, Utkarsh M. Agrawal, Viraj S. Bilay and Nilesh M. Dogra “Supervised Machine Learning techniques for Spam Email Detection”

[14] Konstantin Tretyakov “Machine Learning Techniques in Spam Filtering”

[15] Harjot Kaur and Er. Prince Verma “Survey on E-MAIL SPAM DETECTION using supervised approach with Feature Selection”