INTRODUCATION

1.1 OVERVIEW

In today's globalized world, email is a primary source of communication. This communication can vary from personal, business, corporate to government. With the rapid increase in email usage, there has also been increase in the SPAM emails. SPAM emails, also known as junk email involves nearly identical messages sent to numerous recipients by email. Apart from being annoying, spam emails can also pose a security threat to computer system. It is estimated that spam cost businesses on the order of $100 billion in 2007. In this project, we use text mining to perform automatic spam filtering to use emails effectively. We try to identify patterns using Data-mining classification algorithms to enable us classify the emails as Not Spam or SPAM.

1.2 MOTIVATION

Email has become one of the most important forms of communication. In 2014, there are estimated to be 4.1 billion email accounts worldwide, and about 196 billion emails are sent each day worldwide. Spam is one of the major threats posed to email users. In 2013, 69.6% of all email flows were spam. Links in spam emails may lead to users to websites with malware or phishing schemes, which can access and disrupt the receiver's computer system. These sites can also gather sensitive information from. Additionally, spam costs businesses around $2000 per employee per year due to decreased productivity.Therefore, an effective spam filtering technology is a significant contribution to the sustainability of the cyberspace and to our society.

AIM AND OBJECTIVES

2.0. OBJECTIVES

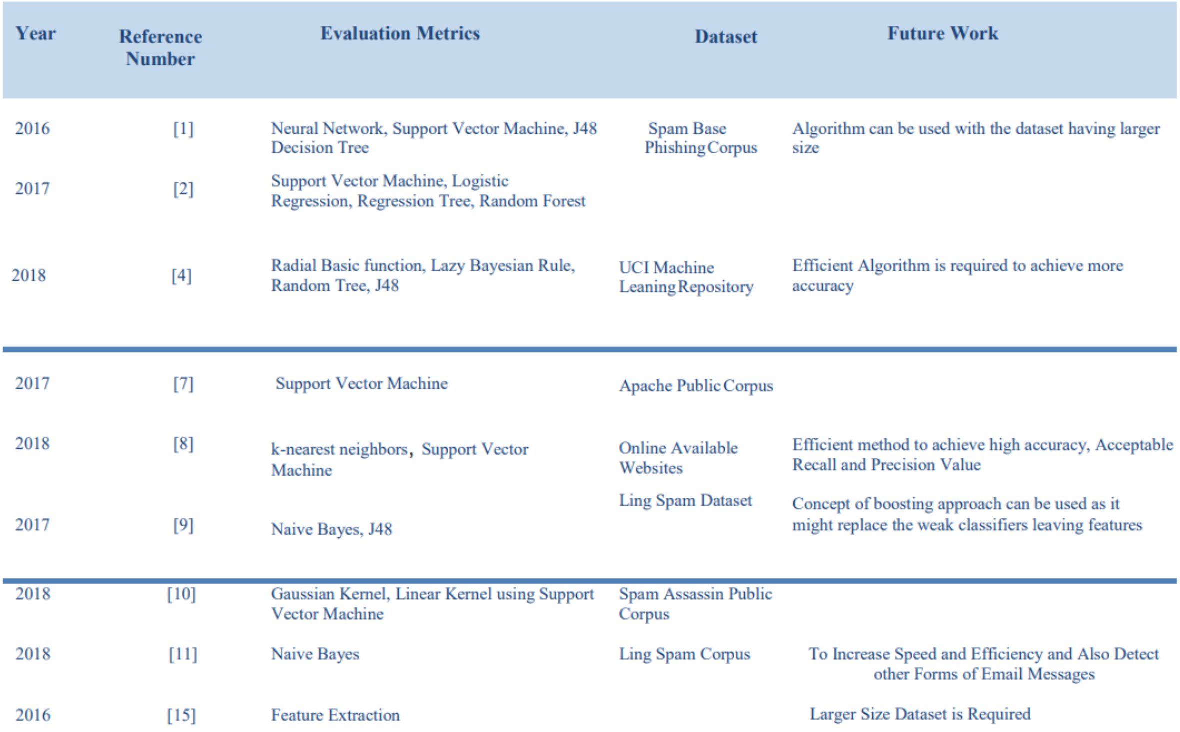
1. User can understand Which mail are getting.
2. User can Immediately know the mail type spam or not spam
3. Time Saving
4. To classify Email into Spam or Ham.

2.1. PROJECT SCOPE AND LIMITATIONS

It considers a complete message instead of single words with respect to its organization. It can be referred as the intelligent approach due to its message examining criteria. It provides sensitivity to the client and adapts well to the future spam techniques. Even if the spam word is slightly modificd, this algorithm stil succeeds and notices the spam content. Even though this frame work is accessible for the URL groups embedded in email, it lacks in action taking for IT security groups. The degree of business in the real time is one more major draw back that is faced by this frame work. It does not provide any clarity about how well it is pertormed in a real time tor the spam campaigns. Spammers can easily develop techniques to meet the preventive measures of Auto RE framework like making legitimate domains fall into the list of illegitimate emails. The results that are obtained do not provide any aggregate view of the large groups of emails. Moreover, it does not let the network administrator to have an online monitoring system across the network

LITERATURE SURVEY

3.1. LITERATURE SURVEY



Year	Reference Number	Evaluation Metrics	Dataset	Future Work
2016	[1]	Neural Network, Support Vector Machine, J48 Decision Tree	Spam Base Phishing Corpus	Algorithm can be used with the dataset having larger size
2017	[2]	Support Vector Machine, Logistic Regression, Regression Tree, Random Forest		
2018	[4]	Radial Basic function, Lazy Bayesian Rule, Random Tree, J48	UCI Machine Leaning Repository	Efficient Algorithm is required to achieve more accuracy
2017	[7]	Support Vector Machine	Apache Public Corpus	
2018	[8]	k-nearest neighbors, Support Vector Machine	Online Available Websites	Efficient method to achieve high accuracy, Acceptable Recall and Precision Value
2017	[9]	Naive Bayes, J48	Ling Spam Dataset	Concept of boosting approach can be used as it might replace the weak classifiers leaving features
2018	[10]	Gaussian Kernel, Linear Kernel using Support Vector Machine	Spam Assassin Public Corpus	
2018	[11]	Naive Bayes	Ling Spam Corpus	To Increase Speed and Efficiency and Also Detect other Forms of Email Messages
2016	[15]	Feature Extraction		Larger Size Dataset is Required

Fig 3.1: LITERATURE SURVEY

PROBLEM STATEMENT

4.1. PROBLEM DEFINITION AND OBJECTIVES

Spamming is one of the major attacks that accumulate the large number of compromised machines by sending unwanted messages, viruses and phishing through emails. We have chosen this project because now days there are lot of people trying to fool you just by sending you fake e-mails like you have won 1000 dollars, this much amount is deposited in your account once you open this link then they will track you and try to hack your in formation. Sometimes relevant e-mails are considered as spam emails

- Unwanted email irritating Intemet consumers.
- Critical email messages are missed and/or delayed.
- Consumers change ISP's all the time looking for consistent email delivery.
- Loss of Internet perfomance and bandwidth.
- Millions of compromised computers.
- Billions of dollars lost worldwide.
- Identity Theft.
- Increase in Worms and Trojan Horses.
- Spam can crash mail servers and fill up hard drives.

Existing System

4.1 Circle packing

One standard "fix" to word clouds involves creating a bubble chart with a circle packing algorithm to arrange the bubbles. This avoids the problem that different word lengths bring to word clouds. However, despite their appeal, in this case, the cure is worse than the illness. The small size of the bubbles prevents writing in the labels of all the countries. I have to put the names into tooltips which appear when you hover your mouse over the bubbles.

While I love these plots, I am not a great fan of tooltips for critical information. You can, no doubt, appreciate this point if you access this from a mobile device or the R-Bloggers website, where the tooltips cannot be seen unless you click on the visualization.

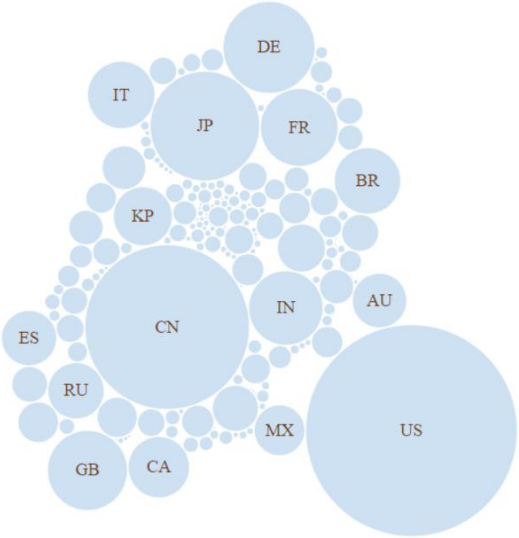


Fig 4.1 Circle Packing

4.2 Cartogram

Rather than packing the circles close together, we can spread them out on a map. I have done this in the cartogram below. The resulting visualization, in most regards, improves on the visualizations above. Problems, however, occur here too. The cartogram relies on a firm understanding of geography, and it fails completely for Europe, where overplotting causes issues. If you have a scroll wheel on your mouse you can zoom in (go to the interactive cartogram). Nevertheless, just as with including names in tooltips (as done with the circle packing), this is a salve rather than a cure. The IMF, who provided the data used in this post, have created a nicer interactive cartogram if you want to see how to do this better.

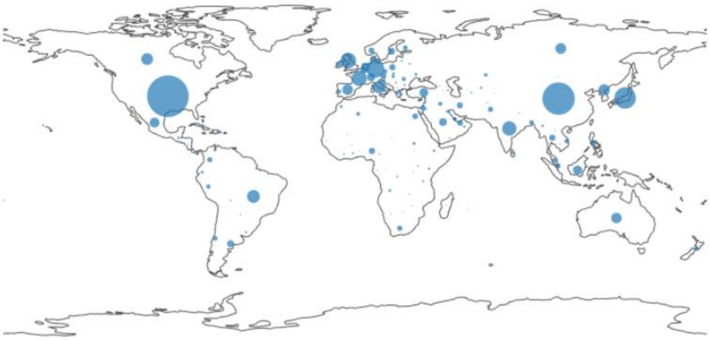


Fig 4.2 Cartogram

4.3 MLlib

This Is the alternative of scikit-learn algorithm. MLlib is Spark's machine learning (ML) library that make practical machine learning scalable and easy it provides ML Algorithms: common learning algorithms such as classification, regression, clustering, and collaborative filtering, feature extraction, transformation, dimensionality reduction, and selection, tools for constructing, evaluating, and tuning ML Pipelines, saving and load algorithms, models, and Pipelines and linear algebra, statistics, data handling, etc.

4.4 Weka

Weka is a machine learning algorithms for data mining tasks that can either be applied directly to a dataset or called from own Java code, it contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization and well-suited for developing new machine learning schemes.

4.5 Tree-Based Methods

In machine-learning, perhaps the best known tree-based methods are AQ11 and ID3, which automatically generate trees from data. Classification And Regression Tree (CART) is perhaps the best well known in the statistics community. All of these tree-based methods work by recursively partitioning the sample space, which–put simply–creates a space that resembles a tree with branches and leaves.

4.6 Traditional Statistical Methods

Traditional statistical methods are time tested and shouldn’t be overlooked in favor of ML algorithms or Neural networks just for the sake of appearing “up to date”. In some cases, traditional methods outperform even the most tried and trusted modern algorithms. For example, in Comparing Classification Methods for Campaign Management, Bichler et al. concluded that "…stepwise logistic regression performed best and dominated all other methods."

Discriminant analysis is a very popular longstanding tool for classification. In a practical sense, there are very minor differences between discriminant analysis and logistic regression (Michie et al. 1994, as cited in Bichler et al., 2004). In fact, LR and linear discrimination are identical for normally distributed data that have equal covariances and for independent binary attributes (Bichler, 2004).

4.7 Working Flow

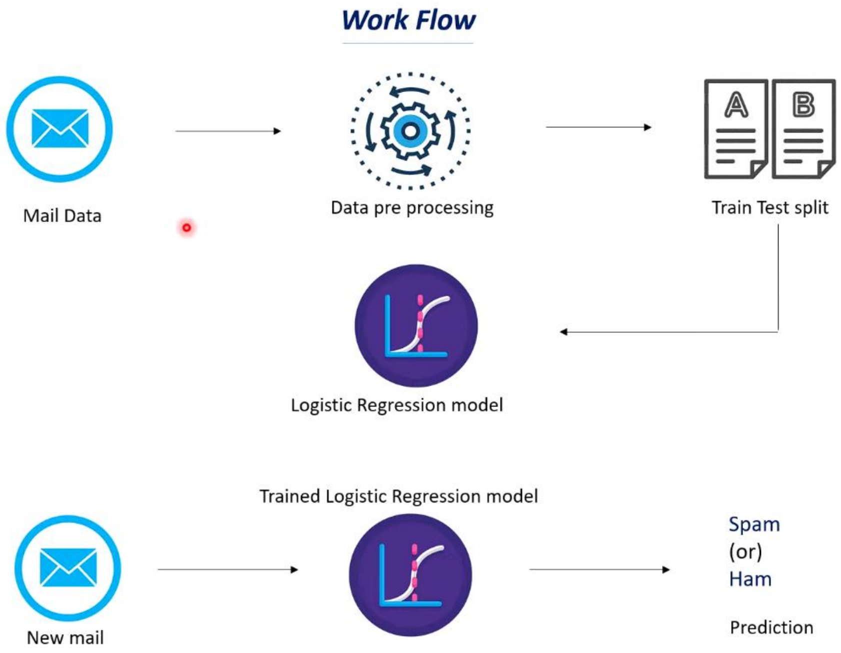


Fig 4.3 Working Flow Diagram

PROPOSED SYSYTEM

5.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis is a very important process of data science. It helps the data scientist to understand the data at hand and relates it with the business context.

The open source tools that I will be using in visualizing and analyzing my data is Word Cloud. Word Cloud is a data visualization tool used for representing text data. The size of the texts in the image represent the frequency or importance of the words in the training data.

Steps to take in this section:

1. Get the email data
2. Explore and analyze the data
3. Visualize the training data with Word Cloud & Bar Chart

5.1.1 Get the spam data

Data is the essential ingredients before we can develop any meaningful algorithm. Knowing where to get your data can be a very handy tool especially when you are just a beginner.

Below are a few of the famous repositories where you can easily get thousand kind of data set for free :

UC Irvine Machine Learning Repository

Kaggle datasets

AWS datasets

For this email spamming data set, it is distributed by Spam Assassin, you can click this link to go to the data set. There are a few categories of the data, you can read the readme.html to get more background information on the data.

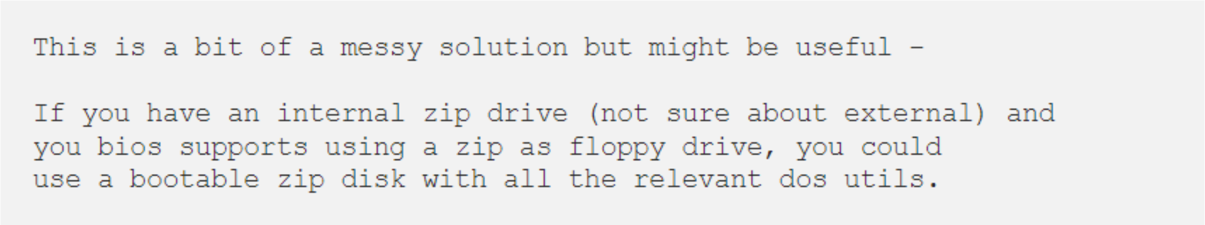
In short, there is two types of data present in this repository, which is ham (non-spam) and spam data. Furthermore, in the ham data, there are easy and hard, which mean there is some non-spam data that has a very high similarity with spam data. This might pose a difficulty for our system to make a decision.

5.1.2 Explore and Analyze the Data

Let’s take a look at the email message content and have a basic understanding of the data

Ham

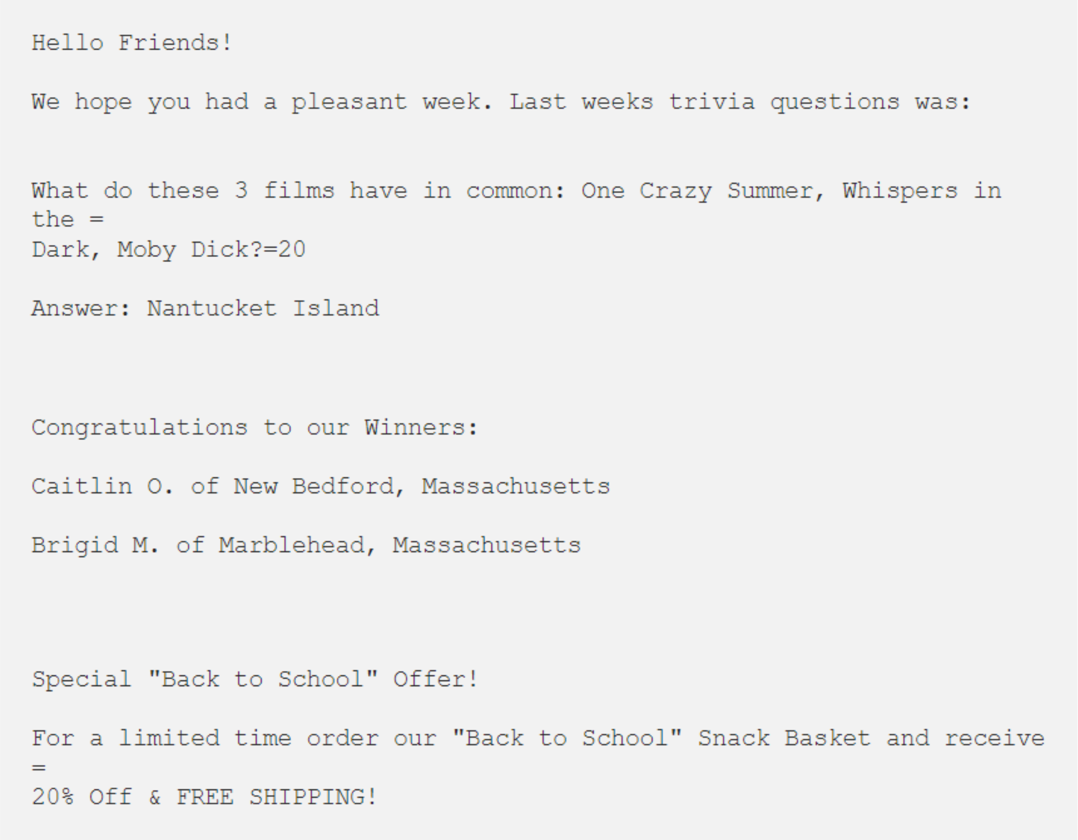
This looks like a normal email reply to another person, which is not difficult to classified as a ham:



```
This is a bit of a messy solution but might be useful -

If you have an internal zip drive (not sure about external) and
you bios supports using a zip as floppy drive, you could
use a bootable zip disk with all the relevant dos utils.
```

5.1.3 Hard Ham (Ham email that is trickier)

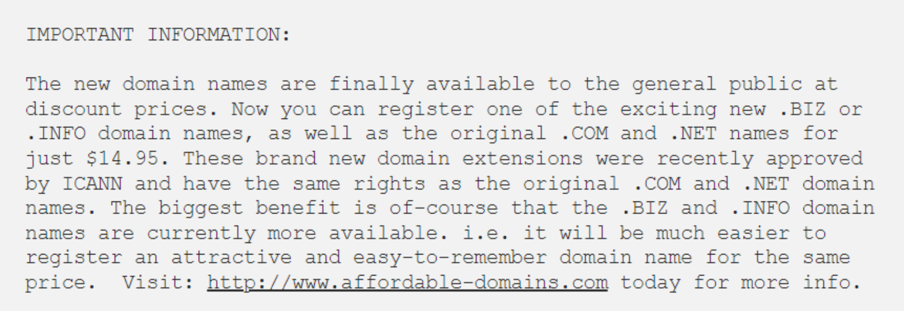
Hard Ham is indeed more difficult to differentiate from the spam data, as they contain some key words such as limited time order, Special “Back to School” Offer, this make it very suspicious !



```
Hello Friends!

We hope you had a pleasant week. Last weeks trivia questions was:


What do these 3 films have in common: One Crazy Summer, Whispers in
the =
Dark, Moby Dick?=20

Answer: Nantucket Island



Congratulations to our Winners:

Caitlin O. of New Bedford, Massachusetts

Brigid M. of Marblehead, Massachusetts



Special "Back to School" Offer!

For a limited time order our "Back to School" Snack Basket and receive
=
20% Off & FREE SHIPPING!
```

5.1.4 Spam

One of the spam training data does look like one of those spam advertisement email in our junk folder:

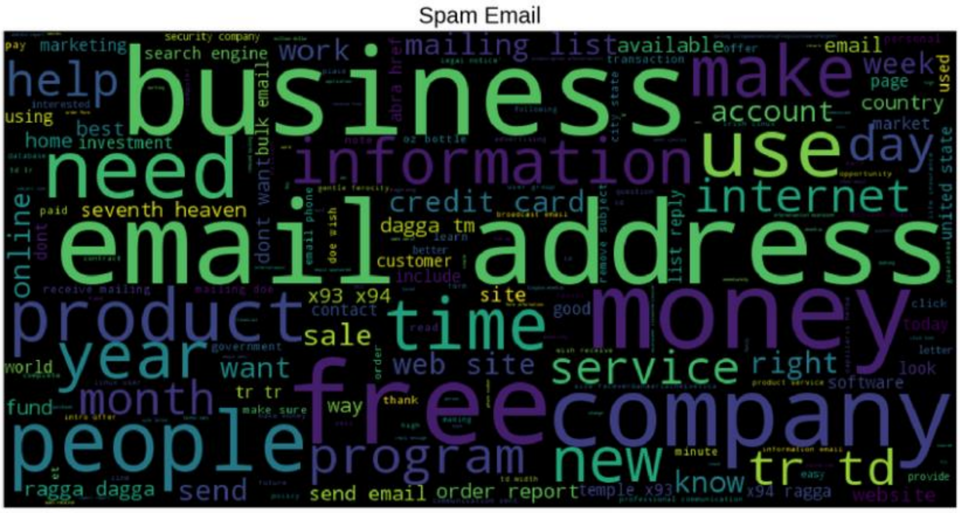


```
IMPORTANT INFORMATION:

The new domain names are finally available to the general public at
discount prices. Now you can register one of the exciting new .BIZ or
.INFO domain names, as well as the original .COM and .NET names for
just $14.95. These brand new domain extensions were recently approved
by ICANN and have the same rights as the original .COM and .NET domain
names. The biggest benefit is of-course that the .BIZ and .INFO domain
names are currently more available. i.e. it will be much easier to
register an attractive and easy-to-remember domain name for the same
price.  Visit: http://www.affordable-domains.com today for more info.
```

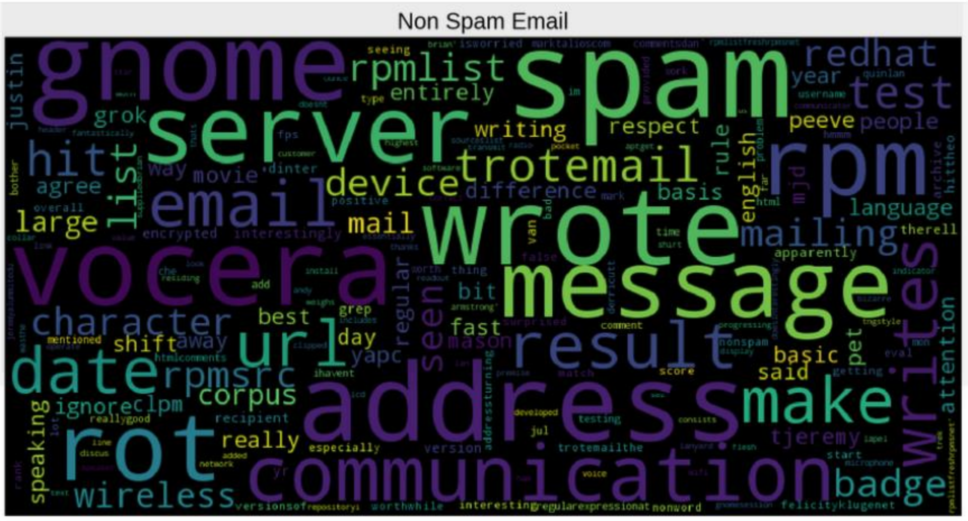
5.2 Visualization

5..2.1 Wordcloud

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



Visualization for spam email



Visualization for non spam email

From this visualization, you can notice something interesting about the spam email. A lot of them are having high number of “spammy” words such as: free, money, product etc. Having this awareness might help us to make better decision when it comes to designing the spam detection system.

One important thing to note is that word cloud only displays the frequency of the words, not necessarily the importance of the words. Hence it is necessary to do some data cleaning such as removing stopwords, punctuation and so on from the data before visualizing it.

5.2.2 Train Test Split

It is important to split your data set to training set and test set, so that you can evaluate the performance of your model using the test set before deploying it in a production environment. One important thing to note when doing the train test split is to make sure the distribution of the data between the training set and testing set are similar.

What it means in this context is that the percentage of spam email in the training set and test set should be similar.

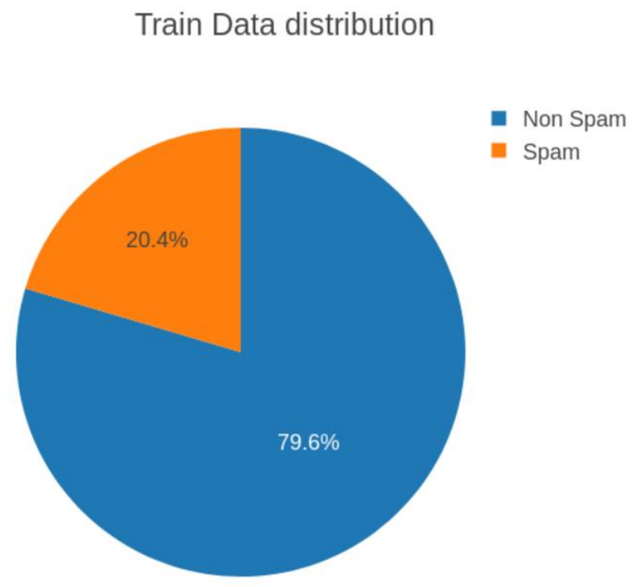


Fig 5.1 Train Data Distribution

The distribution between train data and test data are quite similar which is around 20–21%, so we are good to go and start to process our data !

5.2.3 Data Preprocessing

Text Cleaning

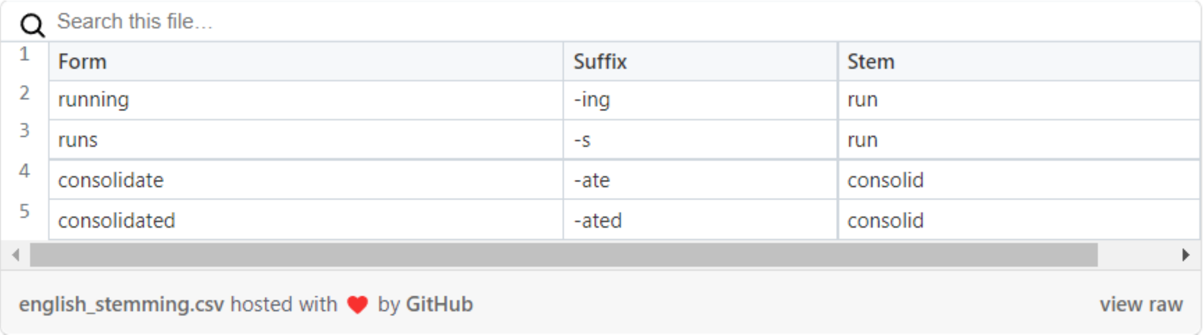
Text Cleaning is a very important step in machine learning because your data may contains a lot of noise and unwanted character such as punctuation, white space, numbers, hyperlink and etc.

Some standard procedures that people generally use are:

- convert all letters to lower/upper case
- removing numbers
- removing punctuation
- removing white spaces
- removing hyperlink
- removing stop words such as a, about, above, down, doing and the list goes on…
- **Word Stemming**
- **Word lemmatization**

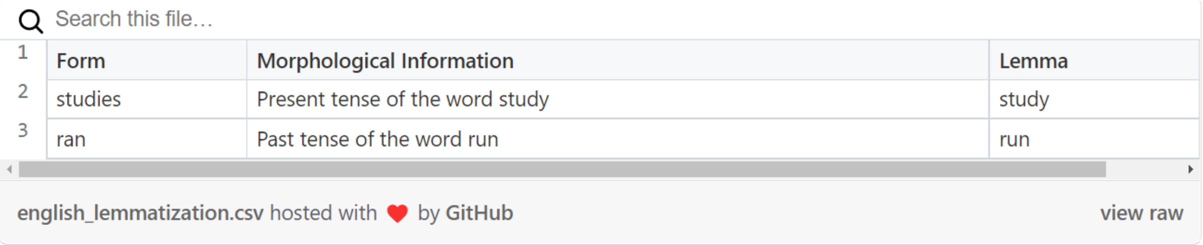
The two techniques that might seem foreign to most people are word stemming and word lemmatization. Both these techniques are trying to reduce the words to its most basic form, but doing this with different approaches.

- Word stemming — Stemming algorithms work by removing the end or the beginning of the words, using a list of common prefixes and suffixes that can be found in that language. Examples of Word Stemming for English words are as below:



Form	Suffix	Stem
running	-ing	run
runs	-s	run
consolidate	-ate	consolid
consolidated	-ated	consolid

- Word Lemmatization — Lemmatization is utilizing the dictionary of a particular language and tried to convert the words back to its base form. It will try to take into account of the meaning of the verbs and convert it back to the most suitable base form.



Form	Morphological Information	Lemma
studies	Present tense of the word study	study
ran	Past tense of the word run	run

Implementing these two algorithms might be tricky and requires a lot of thinking and design to deal with different edge cases.

Luckily NLTK library has provided the implementation of these two algorithms, so we can use it out of the box from the library!

5.2.4 Feature Extraction

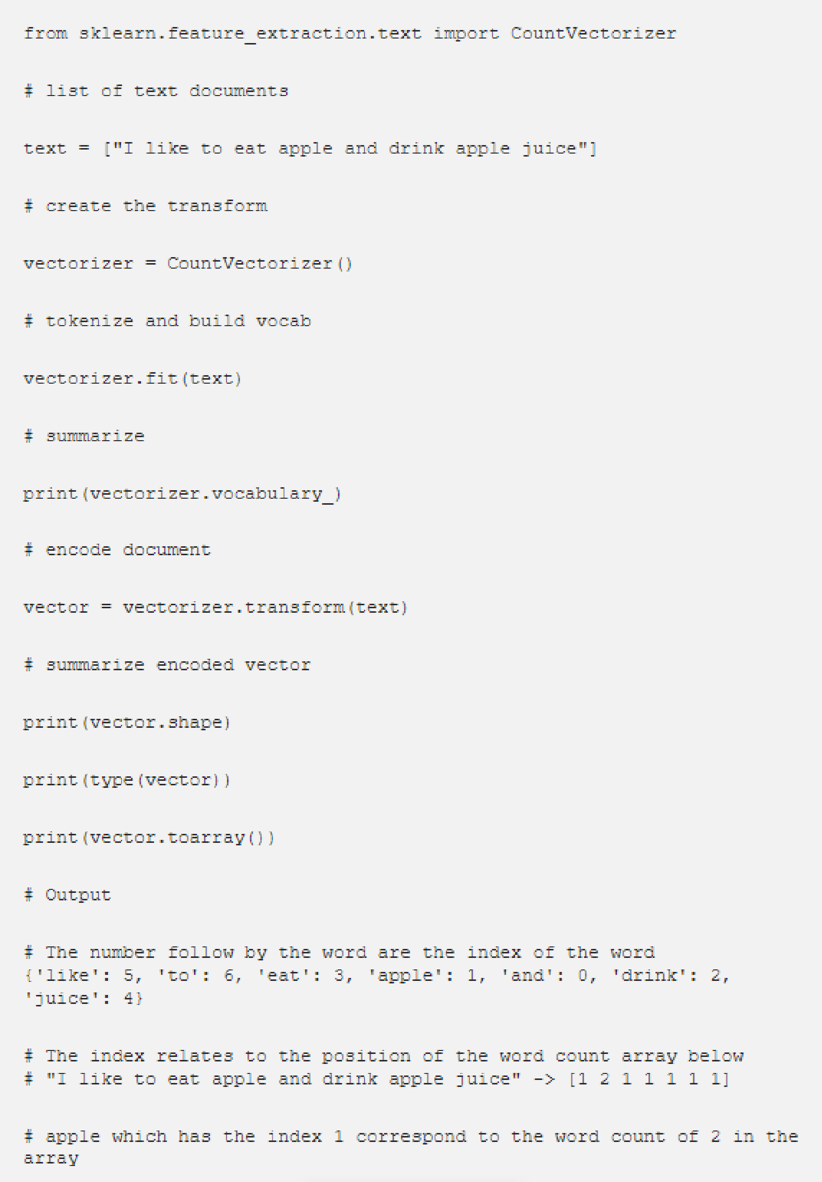
Our algorithm always expect the input to be integers/floats, so we need to have some feature extraction layer in the middle to convert the words to integers/floats

1. CountVectorizer
2. TfidfVectorizer

5.2.5 CountVectorizer

First we need to input all the training data into CountVectorizer and the CountVectorizer will keep a dictionary of every word and its respective id and this id will relate to the word count of this word inside this whole training set.

For example, a sentence like ‘I like to eat apple and drink apple juice’



```
from sklearn.feature_extraction.text import CountVectorizer

# list of text documents

text = ["I like to eat apple and drink apple juice"]

# create the transform

vectorizer = CountVectorizer()

# tokenize and build vocab

vectorizer.fit(text)

# summarize

print(vectorizer.vocabulary_)

# encode document

vector = vectorizer.transform(text)

# summarize encoded vector

print(vector.shape)

print(type(vector))

print(vector.toarray())

# Output

# The number follow by the word are the index of the word
{'like': 5, 'to': 6, 'eat': 3, 'apple': 1, 'and': 0, 'drink': 2,
'juice': 4}

# The index relates to the position of the word count array below
# "I like to eat apple and drink apple juice" -> [1 2 1 1 1 1 1]

# apple which has the index 1 correspond to the word count of 2 in the
array
```

5.2.6 TfidfVectorizer

Word counts are good but can we do better? One issue with simple word count is that some words like ‘the’, ‘and’ will appear many times and they don’t really add too much meaningful information.

Another popular alternative is TfidfVectorizer. Besides of taking the word count of every words, words that often appears across multiple documents or sentences, the vectorizer will try to downscale them.

For more info about CountVectorizer and TfidfVectorizer, please read from this great piece of article, which is also where I gain most of my understanding.

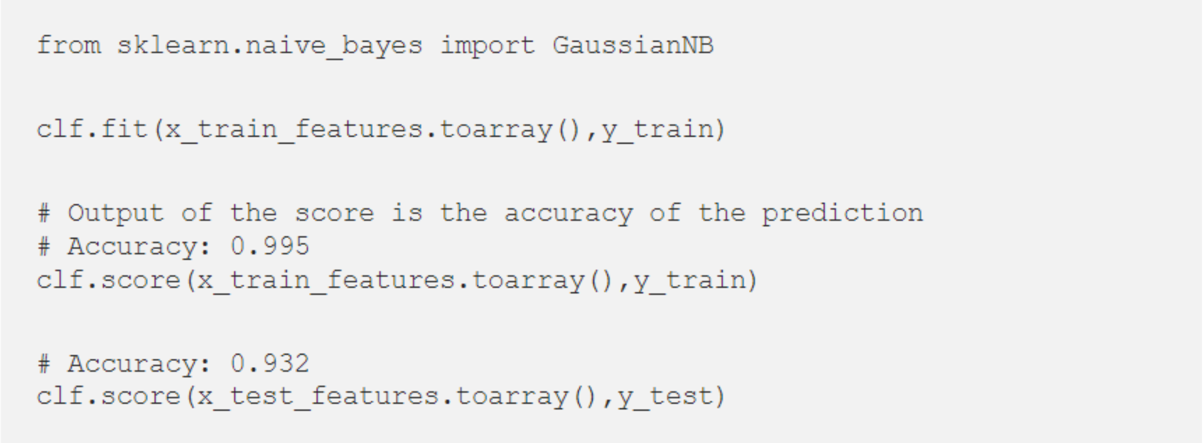
5.2.7 Algorithm Implementation

TfidfVectorizer + Naive Bayes Algorithm

The first approach that I take was to use the TfidfVectorizer as a feature extraction tools and Naive Bayes algorithm to do the prediction. Naive Bayes is a simple and a probabilistic traditional machine learning algorithm.

It is very popular even in the past in solving problems like spam detection. The details of Naive Bayes can be checkout at this article by Devi Soni which is a concise and clear explanation of the theory of Naive Bayes algorithm.

Using Naive Bayes library provided by sklearn library save us a lot of hassle in implementing this algorithm ourselves. We can easily get this done in a few lines of codes



```
from sklearn.naive_bayes import GaussianNB

clf.fit(x_train_features.toarray(),y_train)

# Output of the score is the accuracy of the prediction
# Accuracy: 0.995
clf.score(x_train_features.toarray(),y_train)

# Accuracy: 0.932
clf.score(x_test_features.toarray(),y_test)
```

We achieve an accuracy of 93.2%. But accuracy is not solely the metrics to evaluate the performance of an algorithm. Lets try out other scoring metrics and that may help us to understand thoroughly how well this model is doing.

5.3 Adaptive Spam Filtering Technique

Algorithms classify the incoming mails in various groups and, based on the comparison scores of every group with the defined set of groups, spam and non-spam emails got segregated.

This article will give an idea for implementing content-based filtering using one of the most famous algorithms

for spam detection, which is K-Nearest Neighbour (KNN).

k-NN based algorithms are widely used for clustering tasks. Let's quickly know the entire architecture of this implementation first and then explore every step. Executing these 5 steps, one after the other, will help us implement our spam classifier smoothly.

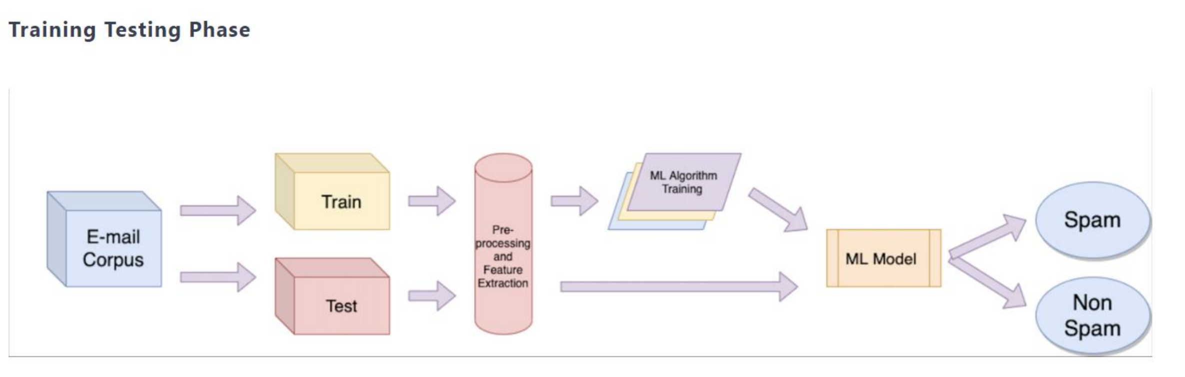


Fig: 5.2 Training Testing Phase

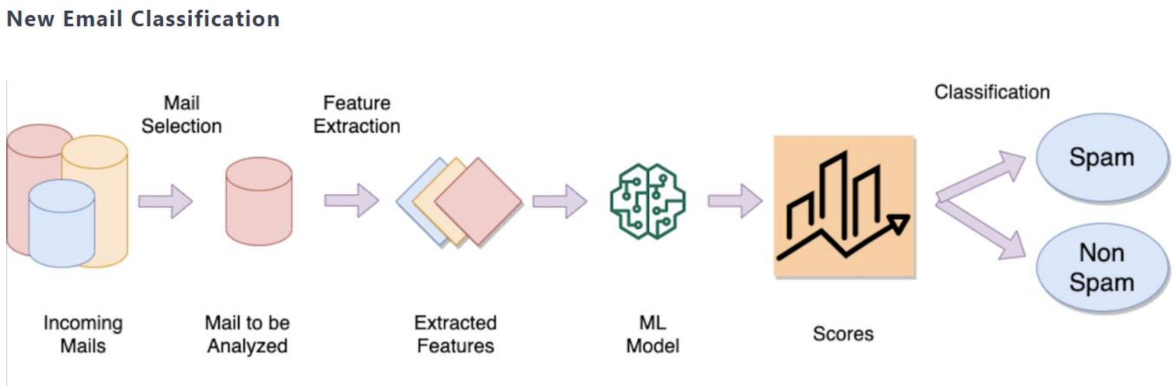


Fig : 5.3 New Email Classification

Methodology

6.1. PROJECT SCOPE AND LIMITATIONS

It considers a complete message instead of single words with respect to its organization. It can be referred as the intelligent approach due to its message examin ing criteria. It provides sensitivity to the client and adapts well to the future spam techniques. Even if the spam word is slightly modified, this algorithm still succeeds and notices the spam content. Even though this frame work is accessible for the URL groups embedded in email, it lacks in action taking for IT security groups. The degree of business in the real time is one more major draw back that is faced by this frame work. It does not provide any clarity about how well it is pertormed in a real time for the spam campaigns. Spammers can easily develop techniques to meet the preventive measures of Auto RE framework like making legitimate domains fall into the list of illegitimate emails. The results that are obtained do not provide any aggregate view of the large groups of emails. Moreover, it does not let the network administrator to have an online monitoring system across the network

6.2. METHODOLOGIES OF PROBLEM SOLVING

6.2.1 Spam mail detection system

This section represents the workflow of the Spam Mail Detection (SMD) System for the classification of emails into ham and spam emails. The SMD system consists of strong classification abilities introduced with the concept hybrid bagged approach. The feature selection method is performed with correlation based feature selection and performed classification with novel hybrid bagging approach. The bagging approach is a hybrid approach where decision tree based J48 algorithm and Naïve Bayes Multinomial classifier is the classification purpose. The flow chart of the SMD system for email classification is presented in fig. 3. The SMD system classifies the email into spam and ham emails. The text based email dataset considered is initially pre-processed for efficient feature extraction. The classification approach

considered is a hybrid bagged approach. The SMD system consists of four modules of Email Dataset Preparation, pre-processing of data, feature selection and hybrid bagged approach.

6.2.2 Outline

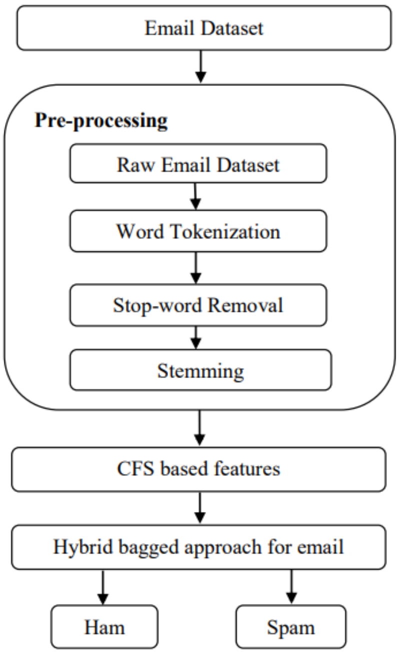
The main goal of these two parts of article is to show how you could design a spam filtering system from scratch.

Outlines of this article are summarized as below:

1. EDA (Exploratory data analysis)
2. Data Pre-processing
3. Feature Extraction
4. Scoring & Metrics
5. Improvement by using Embedding + Neural Network
6. Comparison of ML algorithm & Deep Learning

6.2.3 Email Dataset

An email dataset is prepared for the Spam mail detection system. Different emails are randomly collected from Ling spam dataset. The dataset consists of total number of 1000 emails consisting of



ig 5.1 : Spam mail detection system for email classification

6.2.4 Pre-processing of dataset

The email dataset considered is raw in nature. So it needs to be pre-processed before further consideration. The pre-processing phase consists of three steps. Initially the tokenization of the text data is done. The sentence is split into words known as tokens. From the tokenized words, stop words are removed. Stop words are unwanted words having no linguistic meaning. A text file of approximately 670 stop words is manually prepared and words are removed from the text at the time pre-processing. The third step in the pre-processing module is the stemming. The process of stemming reduces the word to its base word. Stop word removal and stemming are important steps in the pre-processing phase as they help to reduce the search space for efficient feature extraction and selection.

6.2.5 Feature selection

Features play an important role in any of the classification system. SMD system works with assumption that spam mail differs than the ham mail in terms of its content. The feature set contains different features like alphanumeric words, language, grammatical or spelling errors, inappropriate words (words related to advertisement of products/services, dating, adult words etc), frequency count, document length etc. In SMD system correlation feature selection (CFS) method is used. CFS only identifies the best features among the pool of features which are helpful in improving the performance of the system. Correlation based feature selection method works on the assumption that, "Good feature subsets contains features highly correlated with the classification, yet uncorrelated to each other"

Initially text data with feature set is considered as bag of words. The term frequency method is considered to show the number of words per document. The frequency of all words is calculated and words with frequency below a threshold value are eliminated. This method indicates the usefulness of the words and also reduces the search space. The obtained feature set is further reduced using correlation based feature selection method.

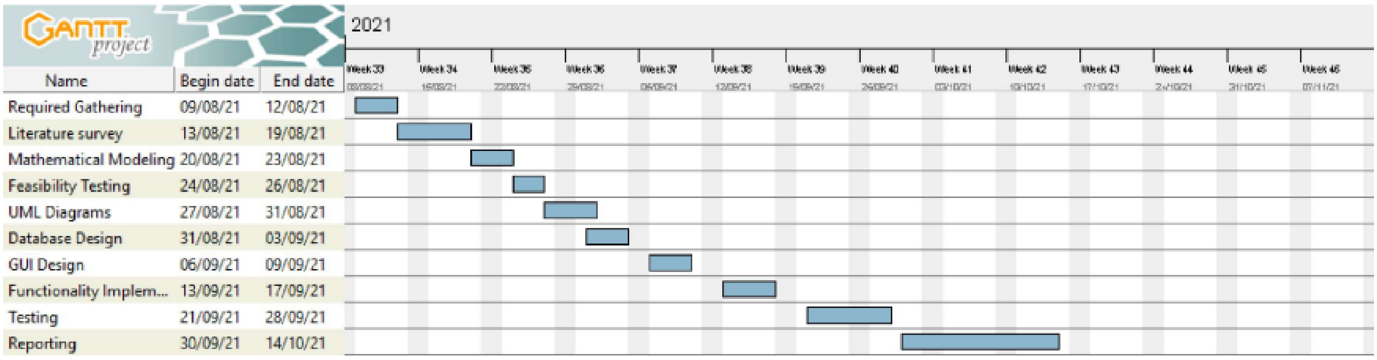
PLANNING

PLANNING

Planning of a system involves the modularization of the project in a definite number of stages and creating sequence of activities to be performed.

PURPOSE

The purpose of this project is to Detect the Email Message Spam Or Not Spam



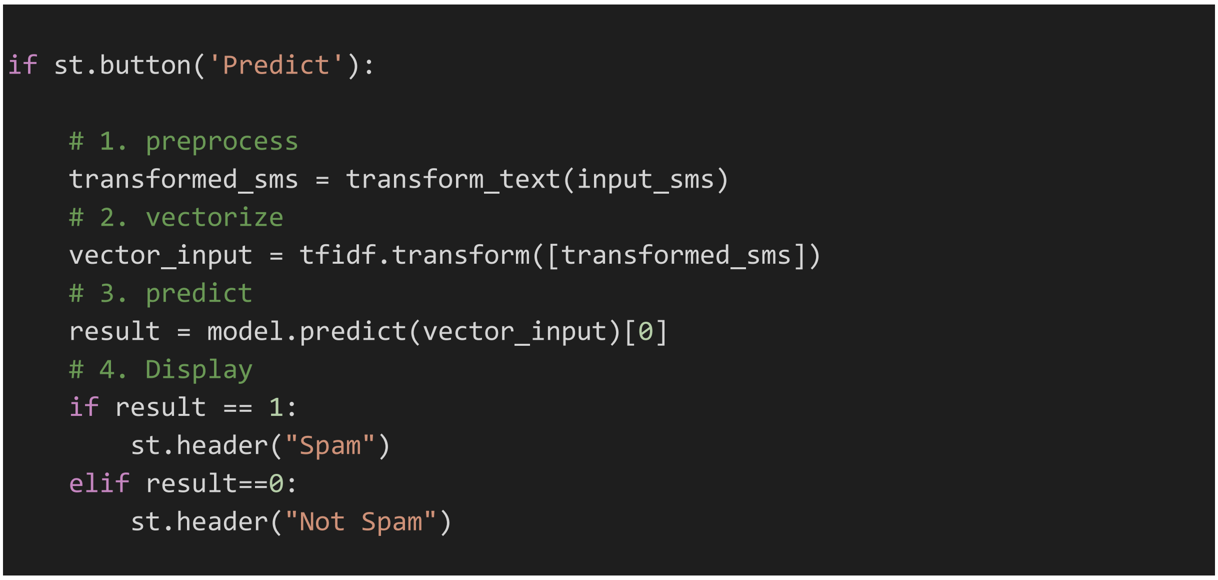
Name	Begin date	End date
Required Gathering	09/08/21	12/08/21
Literature survey	13/08/21	19/08/21
Mathematical Modeling	20/08/21	23/08/21
Feasibility Testing	24/08/21	26/08/21
UML Diagrams	27/08/21	31/08/21
Database Design	31/08/21	03/09/21
GUI Design	06/09/21	09/09/21
Functionality Implem...	13/09/21	17/09/21
Testing	21/09/21	28/09/21
Reporting	30/09/21	14/10/21

DESIGN DETAILS

8.1 Coding

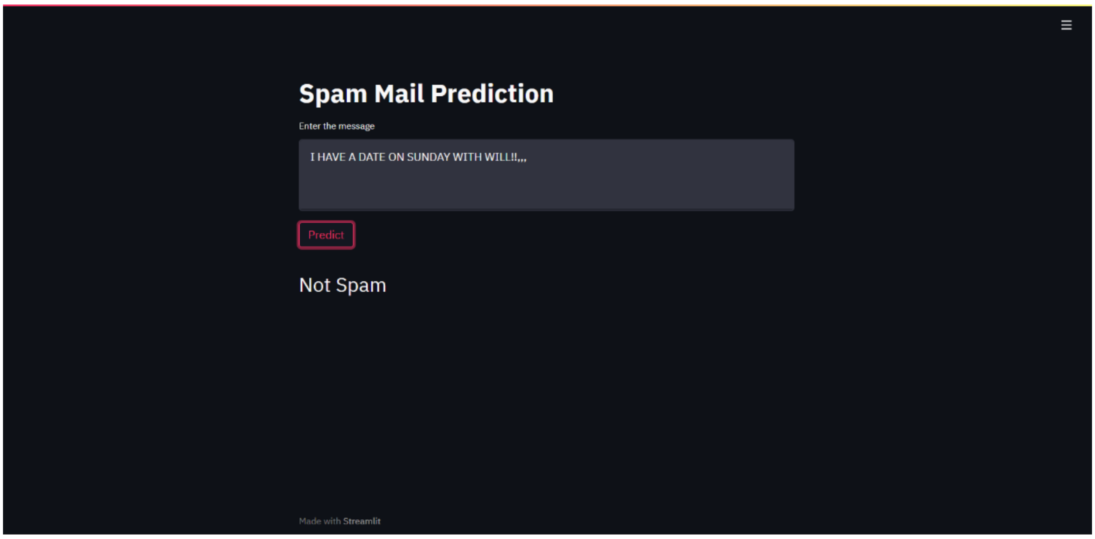
Front end Coding

```
import streamlit as st
import pickle
import string
from nltk.corpus import stopwords
import nltk
import os
os.chdir(r"D:\project\spam 2.1\SMS_EMAIL_SPAM_Detector_web_app-
main_2\SMS_EMAIL_SPAM_Detector_web_app-main\App")

from nltk.stem.porter import PorterStemmer

ps = PorterStemmer()


def transform_text(text):
    text = text.lower()
    text = nltk.word_tokenize(text)

    y = []
    for i in text:
        if i.isalnum():
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        if i not in stopwords.words('english') and i not in
string.punctuation:
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))

    return " ".join(y)


tfidf = pickle.load(open('vectorizer2.pkl','rb'))
model = pickle.load(open('model2.pkl','rb'))

st.title("Spam Mail Prediction")

input_sms = st.text_area("Enter the message")
```



```
if st.button('Predict'):

    # 1. preprocess
    transformed_sms = transform_text(input_sms)
    # 2. vectorize
    vector_input = tfidf.transform([transformed_sms])
    # 3. predict
    result = model.predict(vector_input)[0]
    # 4. Display
    if result == 1:
        st.header("Spam")
    elif result==0:
        st.header("Not Spam")
```

8.2 SCREENSHORT

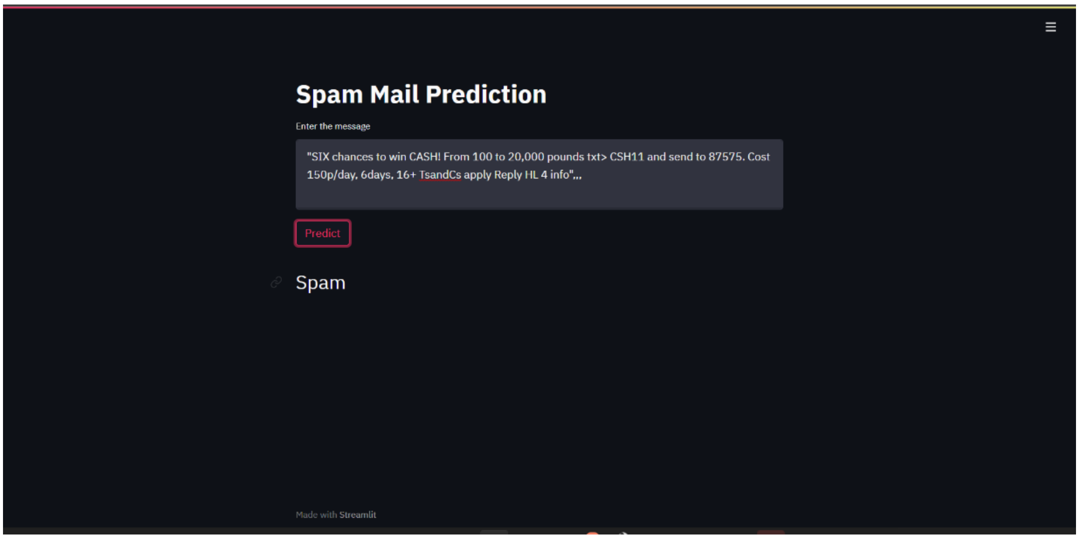


Spam Mail Prediction

Enter the message

I HAVE A DATE ON SUNDAY WITH WILL!!,,,

Not Spam



Spam Mail Prediction

Enter the message

"SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info",,,

Spam

DETAILS OF HARDWARE AND SOFTWARE

9.1. HARDWARE REQUIREMENTS

System	:	Pentium IV 2.4 GHz.
Hard Disk	:	40 GB.
Floppy Drive	:	1.44 Mb.
Monitor	:	14’ Color Monitor.
Mouse	:	Optical Mouse.
Ram	:	4 GB.
Keyboard	:	101 Keyboard Keys.
Components	:	Camera

9.2. SOFTWARE REQUIREMENTS

Operating system	:	Windows 10.
Coding Language	:	Python
Software’s used	:	Python IDE

ADVANTAGES

ADVANTAGES

1) User Can Easily Understand Mail Getting Spam or Not spam.
2) Time Saving Not Mediatory Read Full mail
3) Protection against Viruses
4) Keeping your Reputation Intact

CONCLUSION

CONCLUSION

In this study, we reviewed machine learning approaches and their application to the field of spam filtering. A review of the state of the art algorithms been applied for classification of messages as either spam or ham is provided. The attempts made by different researchers to solving the problem of spam through the use of machine learning classifiers was discussed. The evolution of spam messages over the years to evade filters was examined. The basic architecture of email spam filter and the processes involved in filtering spam emails were looked into. The paper surveyed some of the publicly available datasets and performance metrics that can be used to measure the effectiveness of any spam filter. The challenges of the machine learning algorithms in efficiently handling the menace of spam was pointed out and comparative studies of the machine learning technics available in literature was done. We also revealed some open research problems associated with spam filters. In general, the figure and volume of literature we reviewed shows that significant progress have been made and will still be made in this field. Having discussed the open problems in spam filtering, further research to enhance the effectiveness of spam filters need to be done. This will make the development of spam filters to continue to be an active research field for academician and industry practitioners researching machine learning techniques for effective spam filtering. Our hope is that research students will use this paper as a spring board for doing qualitative research in spam filtering using machine learning, deep leaning and deep adversarial learning algorithms

REFERENCES

https://www.slideshare.net/nabinsjamkatel/spam-email-identification

https://www.slideshare.net/nabinsjamkatel/final-spamemaildetection?from_action=save

Introduction (Project Report PDF)
2.1. PROJECT SCOPE AND LIMITATIONS (Slideshare)
LITERATURE SURVEY (IRJET PDF)

https://www.g2.com/products/scikit-learn/competitors/alternatives (Extraction Ststem)
https://www.displayr.com/alternatives-word-cloud/ (Extraction System)
https://www.datasciencecentral.com/alternatives-to-logistic-regression/(=/=)

https://www.enjoyalgorithms.com/blog/email-spam-and-non-spam-filtering-using-machine-learning

https://towardsdatascience.com/email-spam-detection-1-2-b0e06a5c0472 (Proposed Sysytem)
https://www.enjoyalgorithms.com/blog/email-spam-and-non-spam-filtering-using-machine-learning(Proposed sys)

file:///D:/Project%20Report/Report%202.0/Material/1-s2.0-S2405844018353404-main.pdf (Conclusion)