**OPTIMIZING SPAM FILTERING WITH**

**MACHINE LEARNING**

Submitted in partial fulfillment of requirement for the award of the Degree

**Bachelor of Computer Science**

In the faculty of Computer Science of Bharathiar University, Coimbatore

Submitted by

**TEAM ID: NM2023TMID17953**

**VAISHNAVI S**

**AKILANDESWARI S**

**ARCHANA R**

**VEDASRUTI N**

Under the guidance of

**Dr.V.PUNITHAVATHI MCA,M.Phil,DOA.,Ph.D.**

Lecturer , Department of Computer Science



**DEPARTMENT OF COMPUTER SCIENCE**

**L.R.G GOVERNMENT ARTS COLLEGE FOR WOMEN**

**(Affiliated To Bharathiar University)**

**TIRUPUR-4**

**APRIL-2023**

**LRG GOVT ARTS COLLEGE**

**FOR WOMEN**

**NAAN MUDHALVAN PROJECT WORK**

**(AFFILIATED TO BHARATHIAR UNIVERSITY)**

**TIRUPUR-641604**

**TITLE : OPTIMIZING SPAM FILTERING WITH MACHINE LEARNING**

This is to certify that this is a bonafide record of work done by the above students of III B.sc Computer Science Degree of **NAAN MUDHALVAN PROJECT** during the year 2022-23.

Submitted for the Naan Mudhalvan Project work held on................2023

**CLASS MENTOR HEAD OF THE DEPARTMENT**

[Dr.V.PUNITHAVATHI ] [Dr.R.PARIMALA]

**CONTENT**

|  |  |
| --- | --- |
| **S NO** | **TITLE** |
| 1 | INTRODUCTION  Overview |
| 2 | Problem Definition & Design Thinking  2.1 Empathy Map  2.2 Ideation & Brainstorming map screenshot |
| 3 | RESULTS  3.1 Results and Findings |
| 4 | ADVANTAGES & DISADVANTAGES  4.1 List of advantages and disadvantages of the proposed  Solution |
| 5 | CONCLUSION |
| 6 | FUTURE SCOPE AND ENHANCEMENT |
| 7 | APPENDICES  Source code |

**1.INTRODUCTION**

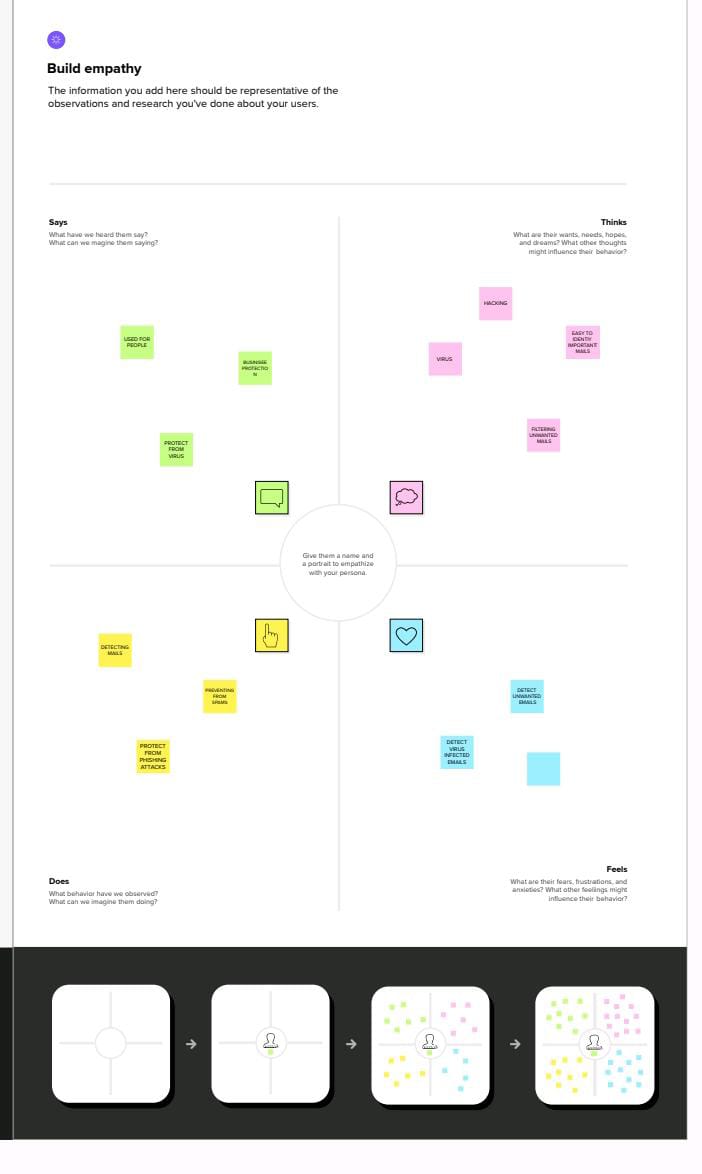
**OVERVIEW**

Over recent years, as the popularity of mobile phone devices has increased, Short Message Service (SMS) has grown into a multi-billion dollar industry. At the same time, reduction in the cost of messaging services has resulted in growth in unsolicited commercial advertisements (spams) being sent to mobile phones. Due to Spam SMS, Mobile service providers suffer from some sort of financial problems as well as it reduces calling time for users. Unfortunately, if the user accesses such Spam SMS they may face the problem of virus or malware. When SMS arrives at mobile it will disturb mobile user privacy and concentration. It may lead to frustration for the user. So Spam SMS is one of the major issues in the wireless communication world and it grows day by day.

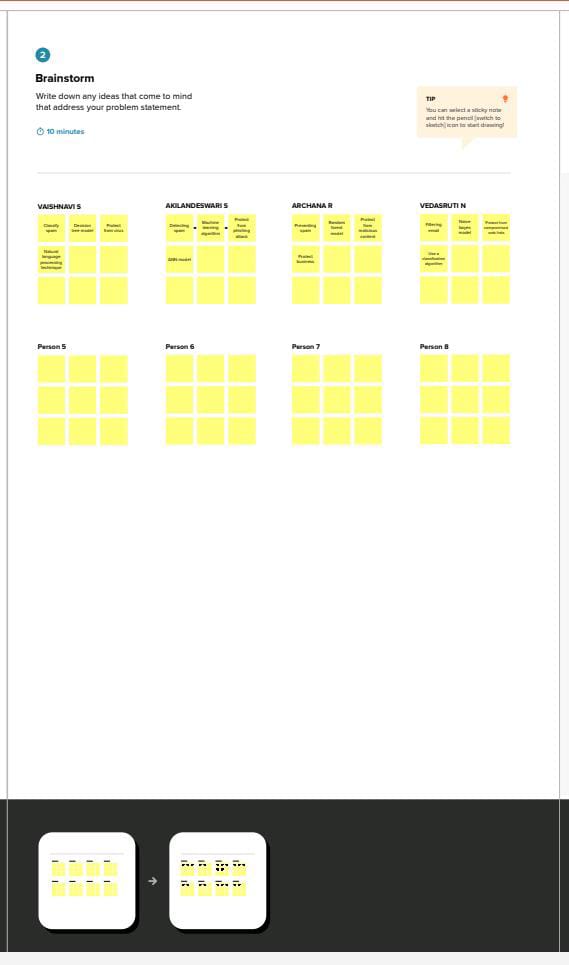
To avoid such Spam SMS people use white and black list of numbers. But this technique is not adequate to completely avoid Spam SMS. To tackle this problem it is needful to use a smarter technique which correctly identifies Spam SMS. Natural language processing technique is useful for Spam SMS identification. It analyses text content and finds patterns which are used to identify Spam and Non-Spam SMS.

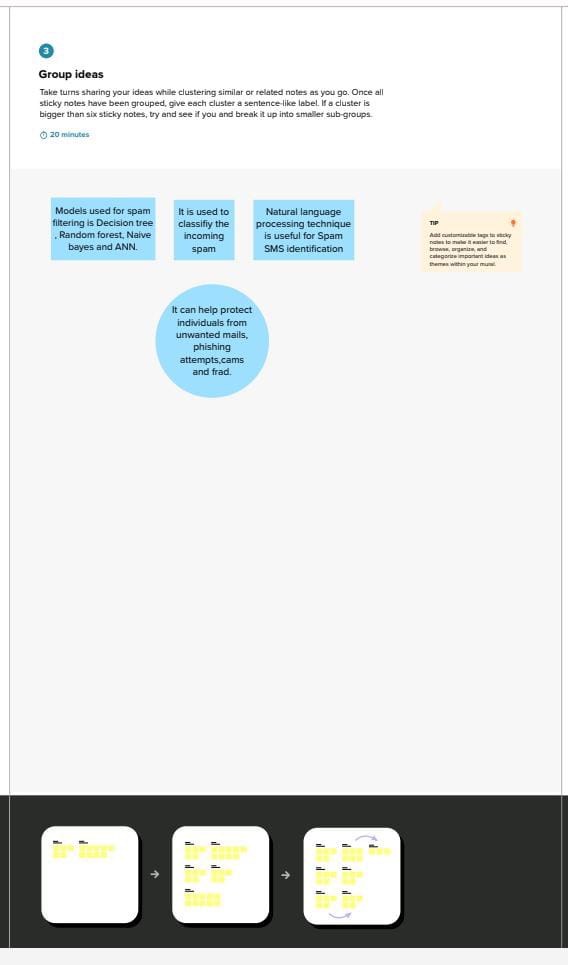
**PROBLEM DEFINITION & DESIGN THINKING**

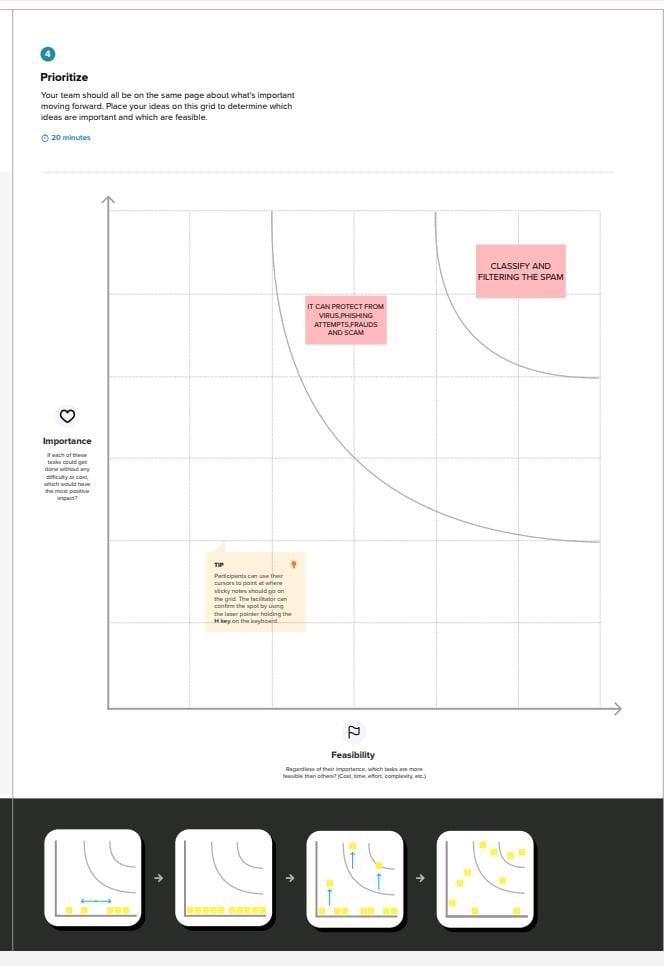
**EMPATHY MAP**

****

**BRAINSTORM**

****

****

****

**PROBLEM SELECTION**

**1.**False positives: This is when legitimate emails are incorrectly classified as spam and filtered out. This problem can be addressed by improving the accuracy of the spam filter and minimizing the number of false positives.

**2**.False negatives: This is when spam emails are not identified and end up in the inbox. This problem can be addressed by improving the sensitivity of the spam filter and minimizing the number of false negatives.

**3.**Evading spam filters: Spammers are constantly evolving their techniques to evade spam filters. This problem can be addressed by staying up-to-date with the latest spamming techniques and updating the spam filter accordingly.

**4.**User feedback: Users may sometimes mark legitimate emails as spam, or vice versa. This can lead to incorrect training of the spam filter and impact its performance. This problem can be addressed by developing a system for collecting and analyzing user feedback to improve the accuracy of the spam

filter.

**5.**User feedback: Users may sometimes mark legitimate emails as spam, or vice versa. This can lead to incorrect training of the spam filter and impact its performance. This problem can be addressed by developing a system for collecting and analyzing user feedback to improve the accuracy of the spam

filter.

**IDEATION**

**1.**Machine learning: Machine learning algorithms can be used to analyze large amounts of data to identify patterns and improve the accuracy of spam filtering.

**2.**Natural Language Processing (NLP): NLP techniques can be used to analyze the text and language used in emails to identify spam. This can include techniques such as sentiment analysis, entity recognition, and part-of-speech tagging.

**3.**Sender reputation: A sender's reputation can be used to determine the likelihood that an email is spam. This can include factors such as the sender's IP address, domain, and history of spamming activity.

**4.**Blacklist and whitelists: Lists of known spam senders and legitimate senders can be used to filter emails. This can be updated regularly to ensure that the lists are accurate.

**5.**Content filtering: The content of emails can be analyzed to identify spam. This can include techniques such as checking for keywords commonly used in spam emails, analyzing the structure of the email, and checking for attachments and links.

**6.**User behavior: User behavior can be used to help identify spam. This can include techniques such as analyzing the frequency and timing of emails, identifying patterns in user behavior, and analyzing user feedback.

**7.**Hybrid approaches: Combining multiple techniques and approaches can improve the accuracy of spam filtering. For example, combining machine learning with NLP and content filtering can provide a more comprehensive approach to spam filtering.

## REQUIREMENT ANANLYSIS

**1.** Accuracy: The spam filter should have a high accuracy rate in detecting spam and avoiding false positives. This can be measured by metrics such as precision, recall, and F1 score.

**2.** Scalability: The spam filter should be able to handle large volumes of email and be scalable to handle increased volumes in the future.

**3.** Performance: The spam filter should operate quickly and efficiently to minimize any delay in email delivery.

**4.** User feedback: The spam filter should have a mechanism for collecting and analyzing user feedback to improve its accuracy and performance.

**5.** Flexibility: The spam filter should be flexible enough to adapt to changing spamming techniques and new types of spam.

**6.** Integration: The spam filter should be easy to integrate with existing email systems and tools, such as email clients and servers.

**PROJECT DESIGN**

**1**. Problem definition: Define the specific problem(s) to be addressed in spam filtering, such as reducing false positives, minimizing false negatives, or staying up-to-date with evolving spamming techniques.

**2.** Requirements gathering: Gather requirements for the spam filter, taking into account the considerations outlined in the requirement analysis section.

**3.** Data collection and preparation: Collect and prepare a dataset of emails, including both legitimate and spam emails. The dataset should be representative of the types of emails that the spam filter will encounter in the real world.

**4.** Algorithm selection and development: Select and develop algorithms for spam filtering, such as machine learning, NLP, and content filtering. Evaluate the performance of each algorithm using appropriate metrics and select the most effective approach.

**5**. Integration: Integrate the spam filter with existing email systems and tools, such as email clients and servers.

**6**. Deployment and monitoring: Deploy the spam filter in a production environment and monitor its performance and effectiveness over time. Continuously update the spam filter to keep up with evolving spamming techniques and address any issues that arise.

## PROJECT PALNNING PHASE

**1.** Define the project scope: Clearly define the goals and objectives of the project, as well as the specific features and functionalities that will be included in the spam filter.

**2.** Identify the project team: Determine who will be involved in the project, including project managers, developers, data scientists, and other stakeholders.

**3.** Develop a project timeline: Create a timeline for the project, including key milestones and deadlines.

**4.** Allocate resources: Determine the resources that will be required for the project, including personnel, hardware, and software.

**5.** Identify potential risks: Identify potential risks that could impact the project, such as delays in data collection, algorithm development, or integration with existing email systems.

**PROJECT DEVELOPMENT PHASE**

1. Data collection and preparation: Collect a representative dataset of emails, including both legitimate and spam emails. Prepare the dataset for analysis, including cleaning and preprocessing.

2. Algorithm development: Develop and test algorithms for spam filtering, such as machine learning, NLP, and content filtering. Evaluate the performance of each algorithm using appropriate metrics and select the most effective approach.

3. Integration: Integrate the spam filter with existing email systems and tools, such as email clients and servers. Ensure that the spam filter is compatible with different email systems and can be easily integrated.

4**.** Testing and validation: Test and validate the spam filter using a separate dataset of emails, measuring the accuracy, performance, and scalability of the spam filter. Identify any issues or bugs and make necessary improvements.

5. Deployment: Deploy the spam filter in a production environment, ensuring that it is compatible with different email systems and can handle large volumes of emails.

6. User feedback and improvement: Collect and analyze user feedback to improve the accuracy and performance of the spam filter. Continuously evaluate the spam filter's performance and effectiveness and make adjustments as necessary.

**ADVANTAGES AND DISADVANTAGES**

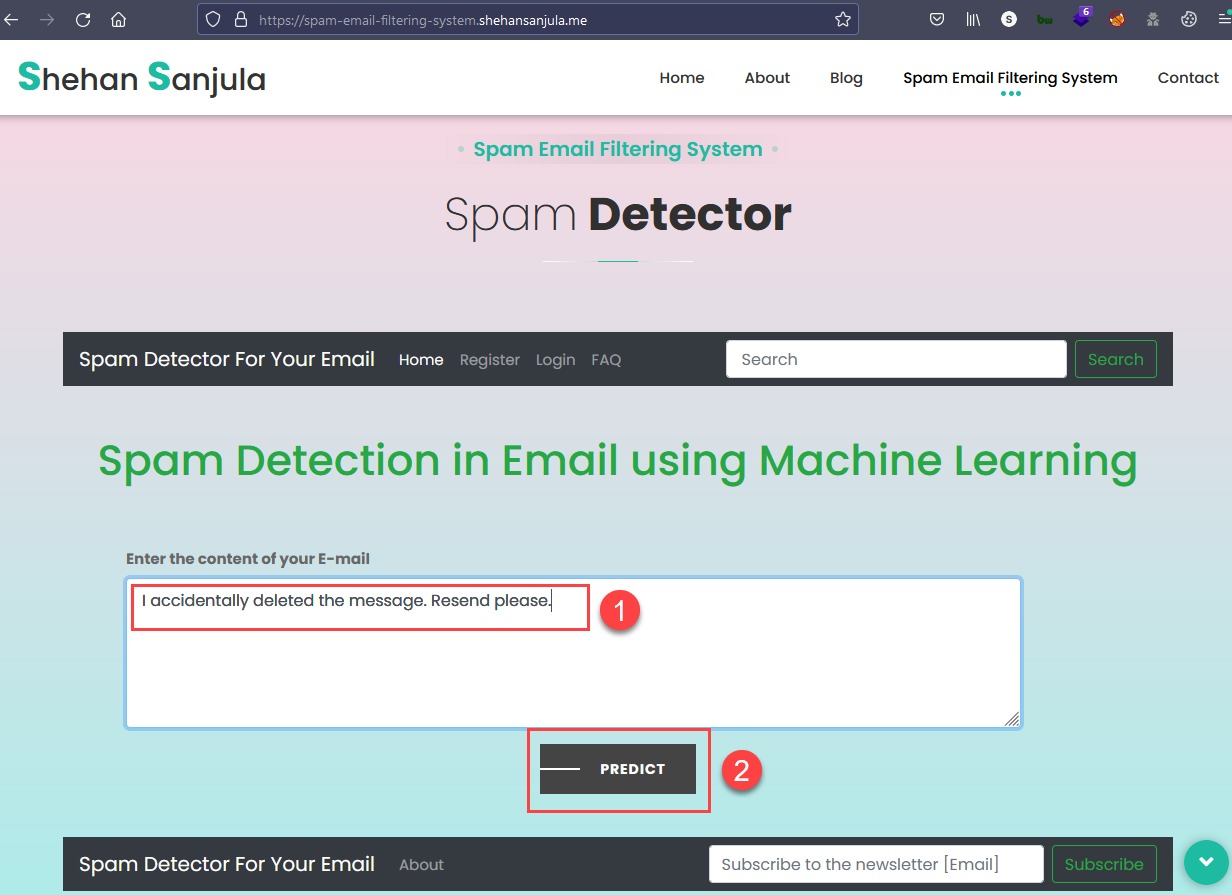
**ADVANTAGES**

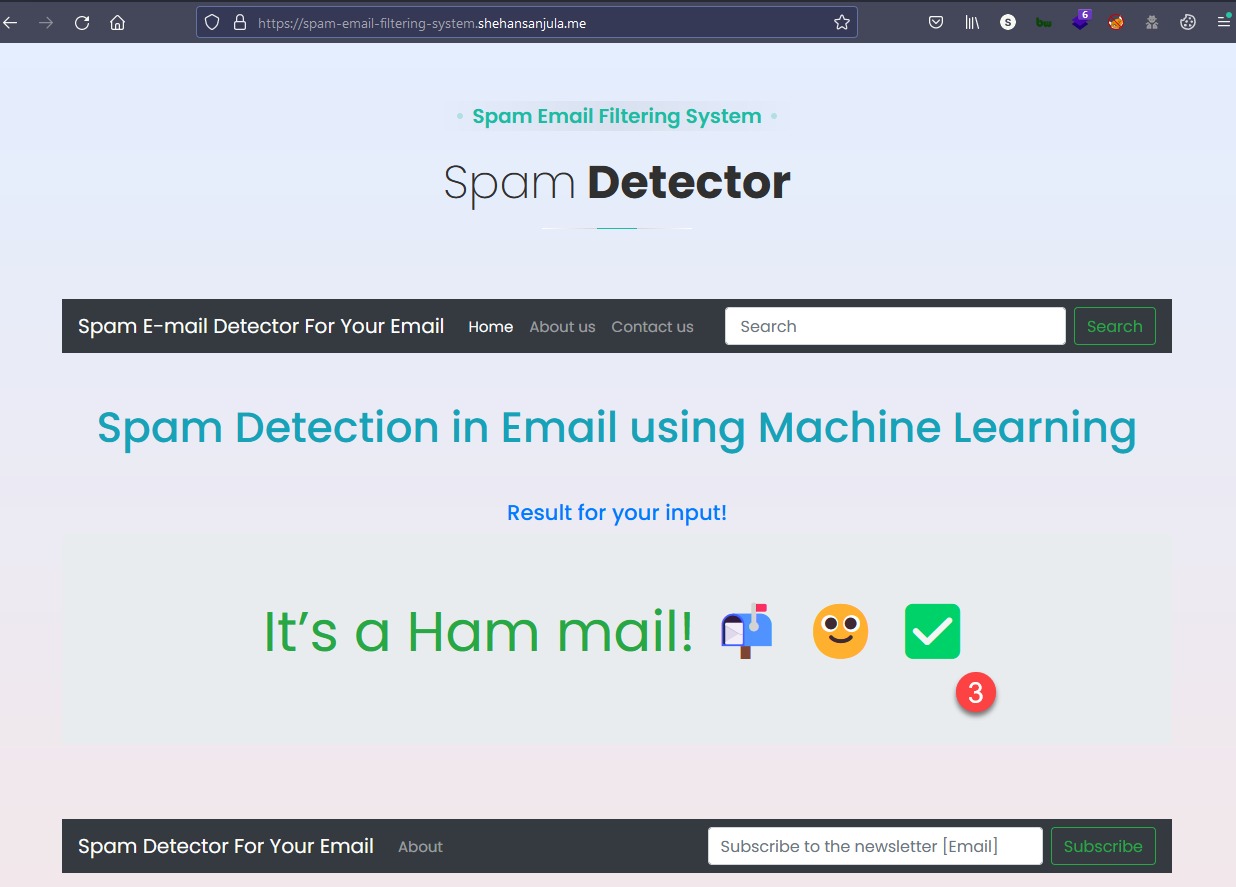
* Filtration based on a sender IP-address analysis is Simplicity of the method implementation
* Filtration based on spam messages repeats detection Suitable for operation with a large flow of letters
* The Bayesian filtering according the Possibility of filter adjusting, low rate of false activation
* Provide clear idea about the effective level of each classifier on phishing email.
* Clustering of phishing email is Fast in classification
* Multi-layered system High level of accuracy by taking advantage of many Classifier.

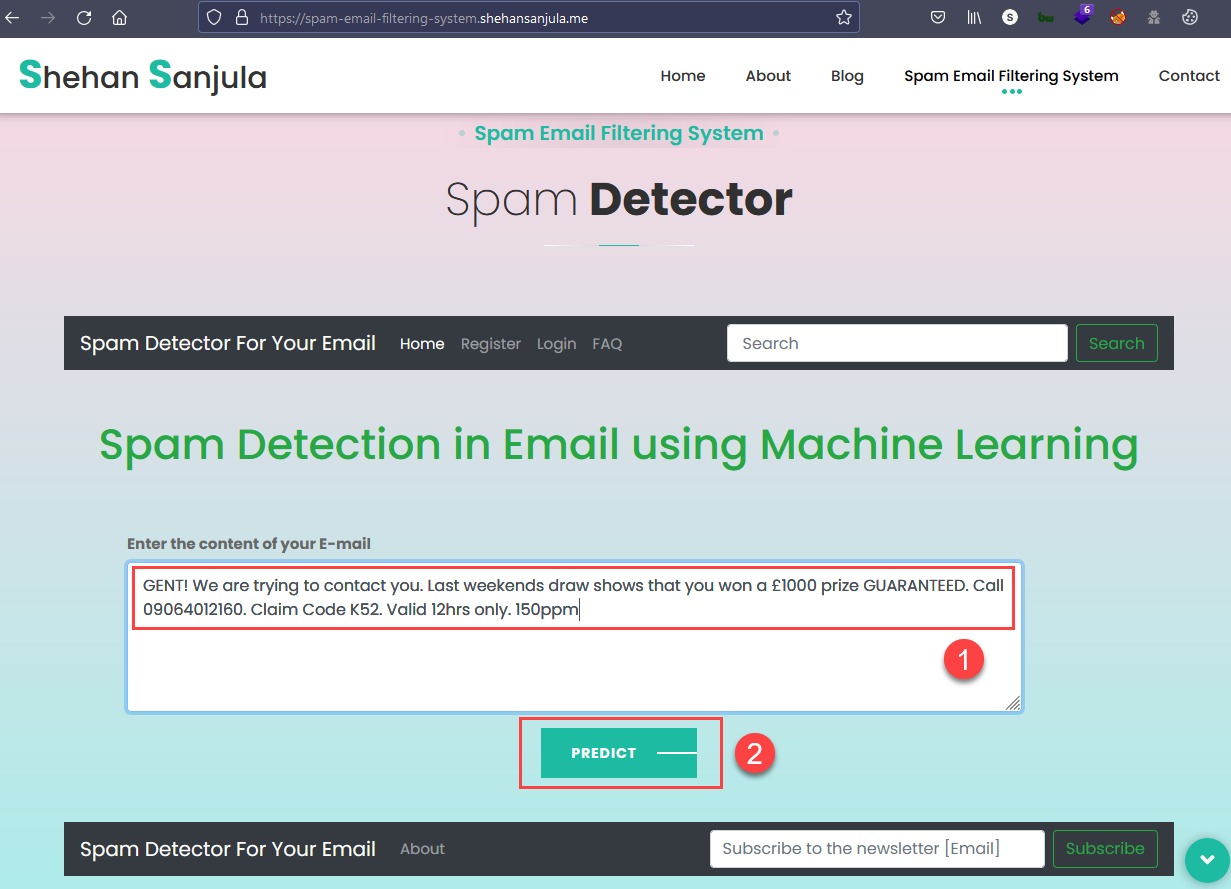
**DISADVANTAGES**

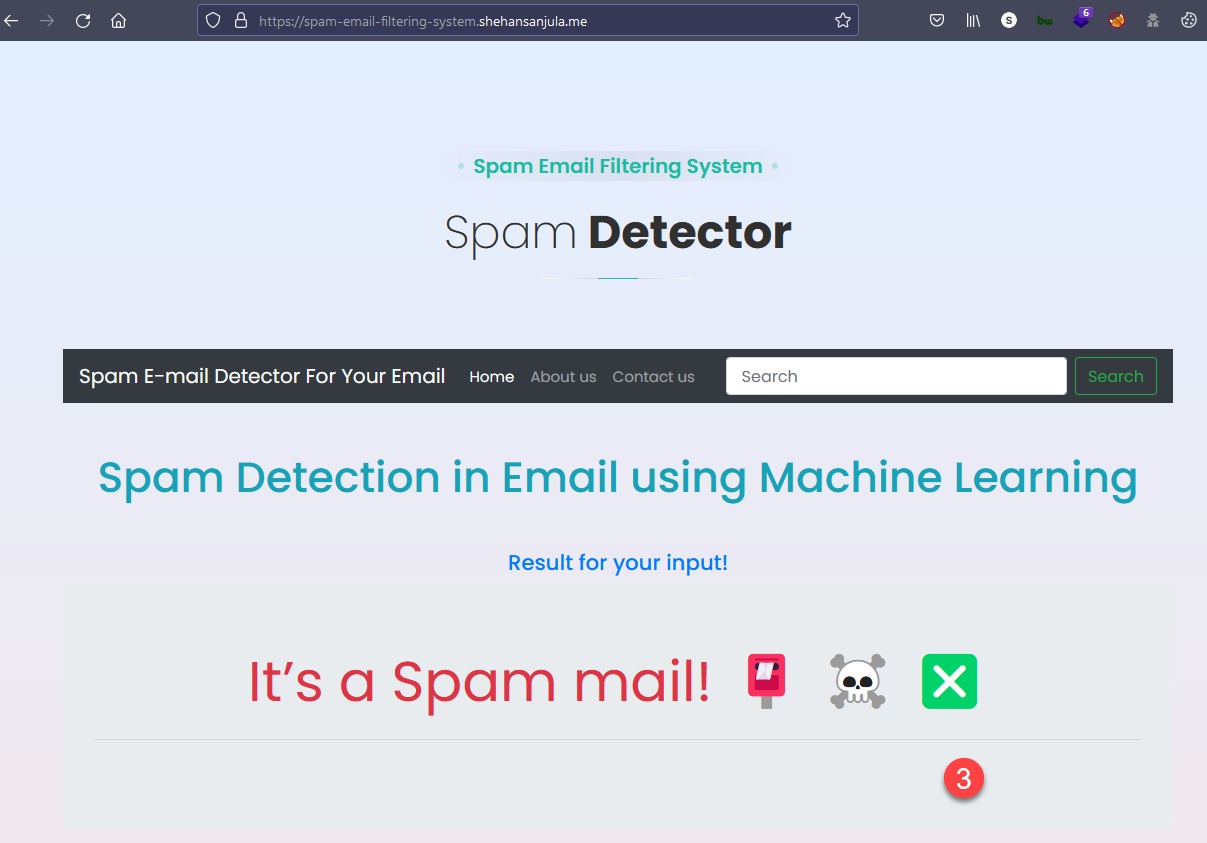
* Filtration based on a sender IP-address analysis is the variability of the feature space, possibility of circumvention.
* Filtration based on spam messages repeats a detection that allows to identify only previously identify spam e-mails, needs to maintain a constant exchange of repeats statistics information.
* The Bayesian filtering according to words requires constant human presence to implement a feedback when the statistical characteristics of letters, only with the text letters, only for individual mailboxes.
* Clustering of phishing email time consuming the technique has many layers working sequentially to produce final result.

**RESULT**

****

****

****



**CONCLUSION**

In the spam detection and filtration gained the attention of a sizeable research community. The reason for a lot of research in this area is its costly and massive effect in many situations like consumer behaviour and fake reviews. The survey covers various machine learning techniques and models that the various researchers have proposed to detect and filter spam in emails and IOT platforms. The study categorized them as supervised, unsupervised, reinforcement learning, etc. The study compares these approaches and provides a summary of learned lessons from each category. This study concludes that most of the proposed email and IOT spam detection methods are based on supervised machine learning techniques. A labelled dataset for the supervised model training is a crucial and time-consuming task. Supervised learning algorithms are SVM and Naive Bayes outperform other models in spam detection. SVM is Support Vector Machine which is one of the machine learning algorithm used for classification process. The study provides comprehensive insights of these algorithms and some future research directions for email spam detection and filtering.

**FURTURE SCOPE AND ENHANCEMENT**

The scope of Machine Learning is not limited to the investment sector. Rather, it is expanding across all fields such as banking and finance, information technology, media and entertainment, gaming, and the automotive industry. As the Machine Learning scope is very high, there are some areas where researchers are working toward revolutionizing the world for the future. Let us discuss them in detail.

**APPENDICES**

**SOURCE CODE**

from flask import Flask, render\_template, url\_for, request

import pandas as pd

import pickle

fromsklearn.model\_selection import train\_test\_split

fromsklearn.feature\_extraction.text import TfidfVectorizer

fromsklearn.linear\_model import LogisticRegression

fromsklearn.metrics import accuracy\_score

app=Flask(\_\_name\_\_)

@app.route('/')

def home():

returnrender\_template('home.html')

@app.route('/predict', methods=['POST'])

def predict():

#Loading the data from csv file to a pandas Dataframe

raw\_mail\_data = pd.read\_csv('mail\_data.csv')

#Replace the null values with a null string

mail\_data = raw\_mail\_data.where((pd.notnull(raw\_mail\_data)),'')

#Label spam mail as 0; ham mail as 1;

mail\_data.loc[mail\_data['Category'] == 'spam', 'Category',] = 0

mail\_data.loc[mail\_data['Category'] == 'ham', 'Category',] = 1

#Separating the data as texts and label

#X-input

#Y-Output/target

X = mail\_data['Message']

Y = mail\_data['Category']

#Splitting the data into training data & test data

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=3)

#Transform the text data to feature vectors that can be used as input to the Logistic regression

feature\_extraction = TfidfVectorizer(min\_df = 1, stop\_words='english', lowercase='True')

#Splited X has string values, those need to be fit & converted to integer

X\_train\_features = feature\_extraction.fit\_transform(X\_train)

X\_test\_features = feature\_extraction.transform(X\_test)

#Convert Y\_train and Y\_test values as integers [convert object type to int]

Y\_train = Y\_train.astype('int')

Y\_test = Y\_test.astype('int')

#Training the Model

model = LogisticRegression()

#Training the Logistic Regression model with the training data

model.fit(X\_train\_features, Y\_train)

#Prediction on training data

prediction\_on\_training\_data = model.predict(X\_train\_features)

accuracy\_on\_training\_data = accuracy\_score(Y\_train, prediction\_on\_training\_data)

#Prediction on test data

prediction\_on\_test\_data = model.predict(X\_test\_features)

accuracy\_on\_test\_data = accuracy\_score(Y\_test, prediction\_on\_test\_data)

ifrequest.method=='POST':

comment=request.form['comment']

data=[comment]

#Convert text to feature vectors

input\_data\_features = feature\_extraction.transform(data).toarray()

#Making prediction

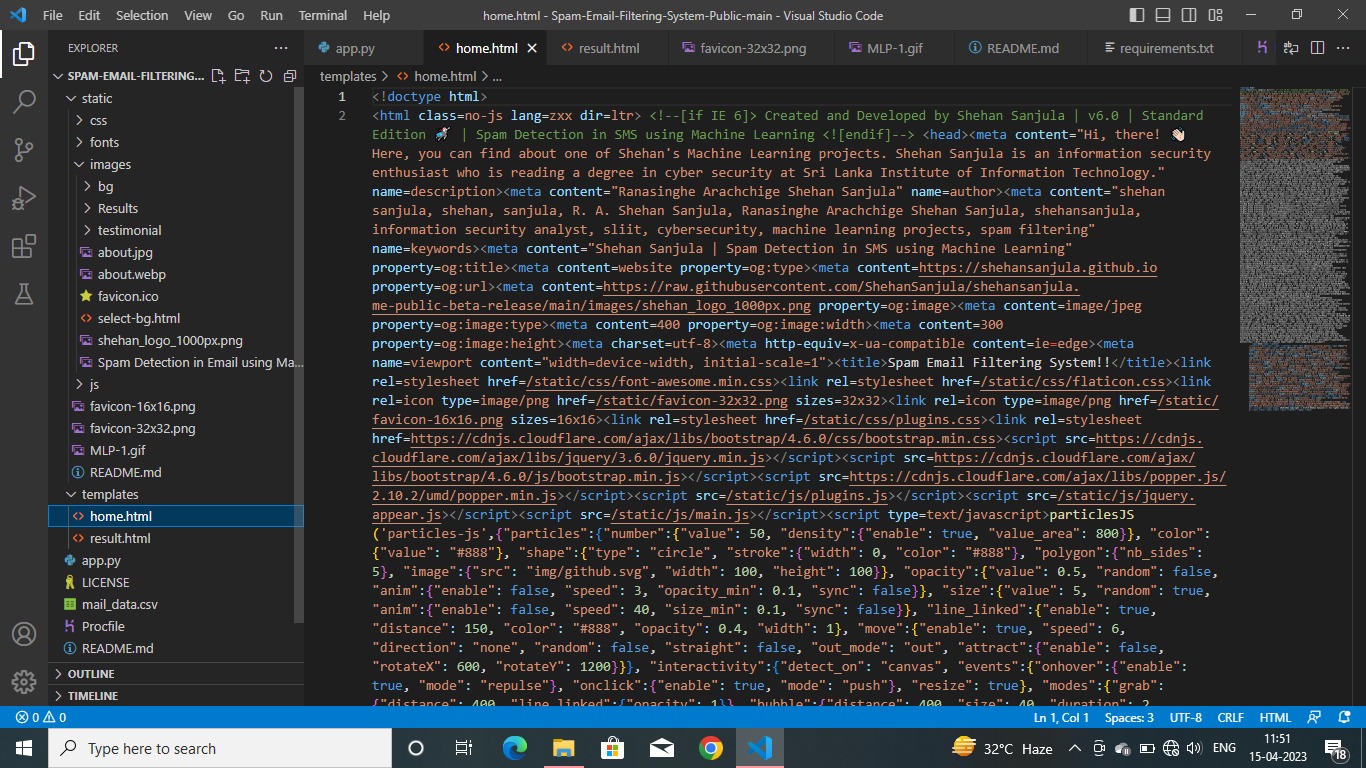
my\_prediction = model.predict(input\_data\_features)

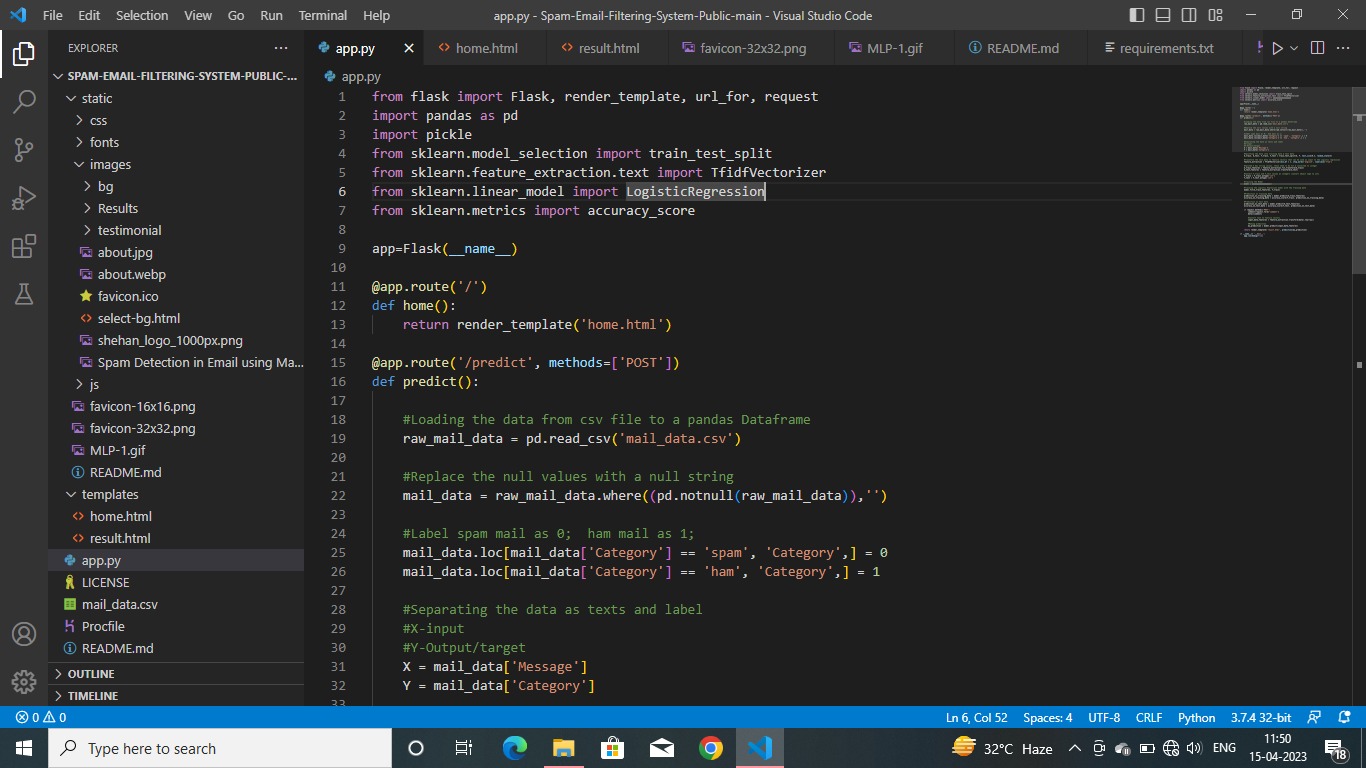
returnrender\_template('result.html', prediction=my\_prediction)

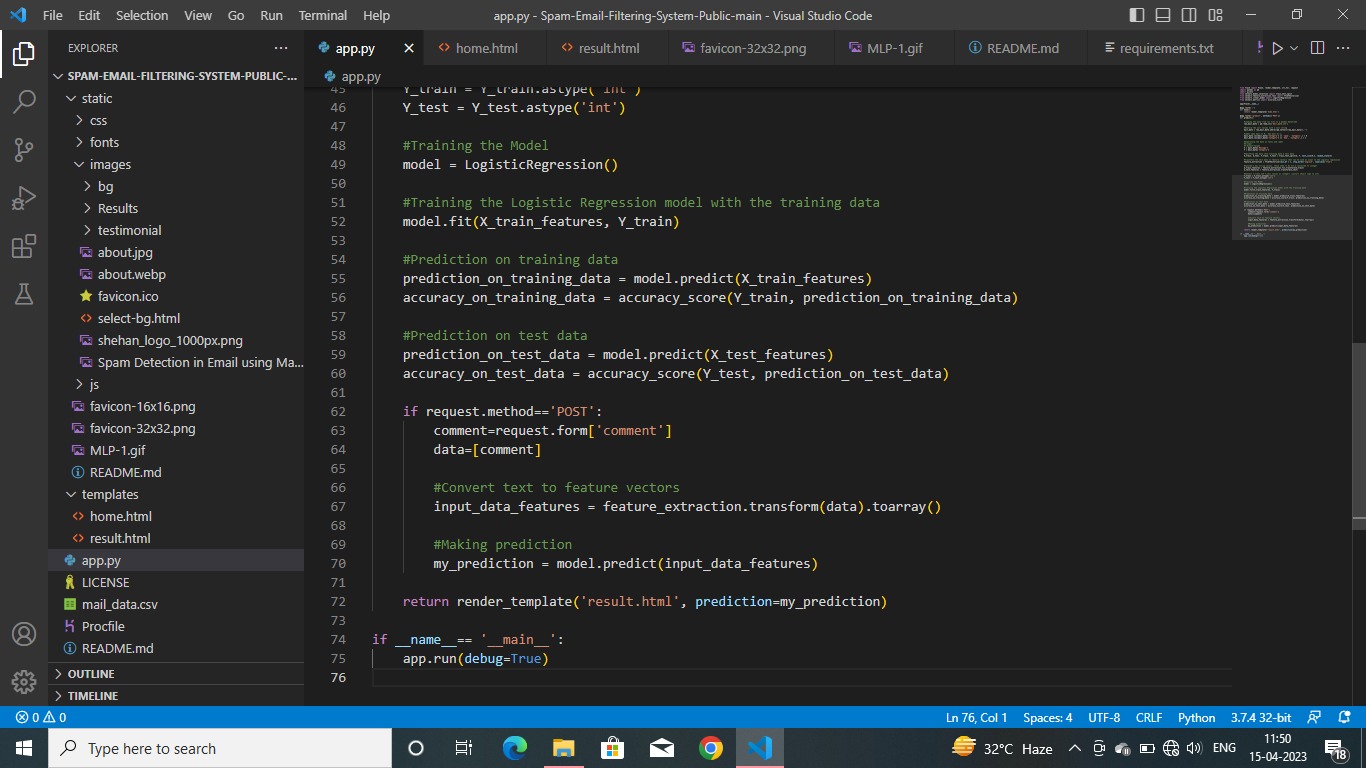
if \_\_name\_\_== '\_\_main\_\_':

app.run(debug=True)

SAMPLE CODING SCREEN



****

****