"Spam Email/SMS Classifier(S.E.C)"

A Project Report Submitted to
Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of
Bachelor of Technology
(*Computer Science and Engineering)*

Submitted By:
Atharva Pagare (0827CS201048)
Atharva Puranik (0827CS201049)
Bhumika Patidar (0827CS201060)

Guided By:
Prof. Preeti Shukla
Department of Computer Science And Engineering, AITR, Indore

Acropolis Institute of Technology & Research, Indore
July - December 2022

EXAMINER APPROVAL

The Project entitled ***"Spam Email/SMS Classifier(S.E.C)"*** submitted by **Atharva_Pagare (0827CS201048),Atharva_Puranik(0827CS201049),Bhumika_Patidar(0827CS201060)** has been examined and is hereby approved towards partial fulfillment for the award of ***Bachelor of Technology degree in Computer Science*** discipline, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve anystatement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner) **(External Examiner)**

Date: **Date:**

GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled ***"Spam Email/SMS Classifier(S.E.C)"*** submitted by Atharva_Pagare(0827CS201048), Atharva_Puranik(0827CS201049), Bhumika_Patidar(0827CS201060) is a satisfactory account of the bonafide work done under the supervision of ***Dr. Kamal Kumar Sethi***, is recommended towards partial fulfillment for the award of the Bachelor of Engineering (Computer Science) degree by Rajiv GandhiProudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide) **(Project Coordinator)**

STUDENTS UNDERTAKING

This is to certify that project entitled "***Spam Email/SMS Classifier(S.E.C)"***has developed by us under the supervision of ***Dr. Kamal Kumar Sethi***. The whole responsibilityof work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

Atharva Pagare (0827CS201048)

Atharva Puranik (0827CS201049)

Bhumika Patidar(0827CS201060)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Dr. Kamal Kumar Sethi,** Professor, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr Kamal Kumar Sethi**, HOD CSE, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from **Dr S C Sharma**, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to **our parent** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say “Thank you”, for being the best family that one could ever have and without whom none of this would have been possible.

Atharva_Pagare (0827CS201048)

Atharva_Puranik (0827CS201049)

Bhumika_Patidar(0827CS201060)

Executive Summary

Spam Email/SMS Classifier(S.E.C)

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal(MP), India for partial fulfillment of Bachelor of Engineering in Information Technology branch under the sagacious guidance and vigilant supervision of Dr. Kamal Kumar Sethi.

The project is based on the spam classifier website, which is used to classify whether an email is spam or not. In the project, Flask is used, which is a framework for Python and a powerful tool for making dynamic and interactive webpages.

Keywords: Python , Flask, dynamic

“An Error Doesn't Become A Mistake Until You Refuse to Correct It.”

- John F.

List of Figures

Table of Contents

Chapter 1 . Introduction

Introduction

The internet has progressively assimilated into daily life. The number of people using email is growing daily as a result of increased internet usage. Spam, or unsolicited mass email, is an issue that has arisen as a result of the growing usage of email. Due to email's current status as one of the greatest mediums for advertising, spam emails are produced. Emails that the recipient does not want to receive are referred to as spam. Multiple email receivers receive a lot of copies of the same message. When we disclose our email address on an unofficial or dishonest website, spam frequently results .Spam has several negative impacts. fills our Inbox with a large amount of absurd emails.significantly reduces our Internet speed. stealing important data from your contacts list, such our contact information.any computer programme that modifies the search results you receive.Spam is a major time waster for everyone and, if you get a lot of it, it can get downright annoying.It takes time to locate these spammers and their offensive information. These emails could include links to phishing or malware-hosting websites known to steal sensitive data. Utilising various spam filtering techniques, this issue has been resolved. The spam filtering methods are used to keep our mailboxes free of unwanted emails.

1.1 Overview

Today, a sizable portion of individuals rely on freely accessible email or communications provided by strangers. Because anybody may send an email or leave a note, spammers have an excellent chance to compose spam messages regarding our various interests.Spam overflows email inboxes with absurd emails. severely reduces the speed of our internet.stealing vital information, such as our contact information, from us. Finding these spammers and the spam content may be difficult work and a popular research area. Spam email is the practise of sending many communications using postal mail.Spam is basically postage due advertising because the recipient bears the majority of the cost. Spam email is a form of commercial advertising that is commercially feasible due to email's potential as a very costeffective medium for senders. Using Bayes' theorem and Naive Bayes' Classifier, the suggested model allows the supplied message to be classified as spam or not.

1.2 Background and Motivation

The motivation behind developing an email spam classifier is to address the problem of spam emails, which can be a significant issue for individuals and organizations alike. Spam emails can waste valuable time, reduce productivity, and even pose security risks if they contain malware or phishing scams.By developing an email spam classifier, we can automatically identify and filter out unwanted spam emails, saving time and improving the efficiency of email communication. This is particularly important for businesses and organizations that receive a large volume of emails every day, as spam emails can significantly impact their operations.

Furthermore, email spam classifiers can improve email security by detecting and blocking phishing emails that attempt to steal sensitive information such as usernames, passwords, and financial information. This can help protect individuals and organizations from cyberattacks and data breaches.

1.3 Problem Statement and Objectives

Unwanted spam emails can be a significant issue for individuals and organizations, leading to wasted time and reduced productivity, as well as potential security risks. The objective of a spam email classifier is to automatically identify and filter out unwanted spam emails, thereby improving the efficiency of email communication and enhancing email security.

1.4 Scope of the Project

As the amount of spam emails increases and spammers' techniques advance, the future potential of spam email detection models seems pretty bright. Here are some probable directions for advancement in spam email detection algorithms going forward:Deep Learning-based models: Deep Learning techniques like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated promising results in natural language processing tasks, and they may also be helpful for spam email identification.Hybrid Models: Combining rule-based and machine learning-based techniques may produce superior outcomes to each technique used alone. For instance, a model that combines keyword-based criteria with a machine learning algorithm could be better at spotting spam emails

1.5 Team Organization

- **Atharva Pagare:** I helped with the documentation work as well as the frontend work of the project, creating the html templates and designing the website.
- **Atharva Puranik :** I contributed to the project's backend, helped with code debugging, and helped with writing documentation.
- **Bhumika Patidar :** By developing the html templates and designing the website, I contributed to both the frontend and documentation work of the project.

1.6 Report Structure

The project ***Spam Email/SMS Classifier*** is primarily concerned with the **Spam classification** and whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the

requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the differentdeployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. Thechapter also includes the different user interface designed in project along with their functionality. Further it discuss the experiment results along with testing ofthe project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2 . Review of Literature

Review of Literature

The first attempts to tackle the problem of spam emails involved creating rules-based filters that could identify spam emails based on specific keywords and patterns. These filters were relatively effective in the early days of spam, but spammers quickly adapted by using more sophisticated techniques, such as randomizing the text and using images to bypass the filters.In the early 2000s, machine learning techniques started to be applied to the problem of email spam classification. The first machine learning-based spam filters used Bayesian algorithms, which were able to learn from the patterns of spam emails and make predictions based on probabilities.As spammers continued to evolve their techniques, more advanced machine learning algorithms were developed, such as support vector machines (SVMs) and decision trees. These algorithms were able to identify more complex patterns in spam emails and improve the accuracy of email spam classification.Today, most email providers use a combination of rules-based filters and machine learning algorithms to classify spam emails. These classifiers are continually updated and improved to keep up with the evolving tactics of spammers.Overall, the history of email spam classification shows how the problem has evolved over time and how technology has been used to develop increasingly sophisticated methods for identifying and filtering out unwanted spam emails..

2.1 Preliminary Investigation

2.1.1 Current System and Its Limitations

1. Hybrid Classifier :

Hybrid classifiers combine multiple techniques, such as rule-based filters and machine learning algorithms, to improve the accuracy of email spam classification. However, these classifiers can be complex and difficult to maintain.

Disadvantages: False positives: Spam classifiers may incorrectly identify legitimate emails as spam, leading to important emails being missed.

2. Active Learning:

Active learning is a method that uses human feedback to improve the accuracy of machine learning models. However, it requires human input and can be time-consuming.

Disadvantages: Adversarial attacks: Spammers may use adversarial techniques to bypass spam filters, such as obfuscating text or images to make them harder to classify.

3. Ensemble Methods:

Ensemble methods combine multiple classifiers to improve the accuracy of email spam classification. However, these methods can be complex and difficult to implement.

Disadvantage : Evolving tactics: Spammers are constantly adapting their tactics, making it difficult for spam classifiers to keep up.

4. Transfer learning:

Transfer learning is a method that uses pre-trained models to improve the accuracy of machine learning models. It can be used in email spam classification by using pre-trained models from other domains to improve the accuracy of spam classification. However, transfer learning may not always be effective if the pre-trained models are not well-suited to the email spam classification task.

Disadvantages: Spammers are constantly adapting their tactics, making it difficult for spam classifiers to keep up.

5. Graph-based semi-supervised learning:

Graph-based semi-supervised learning is a method that uses a graph to represent the relationships between labeled and unlabeled data, and then uses this graph to propagate label information from the labeled data to the unlabeled data. It can be used in email spam classification by constructing a graph that represents the similarities between emails and propagating label information from a small set of labeled emails to a large set of unlabeled emails.
Disadvantages: graph-based semi-supervised learning can be computationally expensive and may require careful tuning of the graph construction and label propagation methods.

2.2 Requirement Identification and Analysis for Project

Requirement identification and analysis is an important step in developing an effective email spam classifier. This involves identifying the key features and requirements that the classifier must have in order to accurately and efficiently classify spam emails. Some key requirements for an email spam classifier include:

- Accuracy: The classifier must have a high accuracy rate in order to effectively identify spam emails and reduce false positives and false negatives.
- Speed: The classifier must be able to process emails quickly, especially in high-volume email environments.
- Scalability: The classifier must be able to handle large volumes of emails and be able to scale up or down depending on the needs of the organization.
- Adaptability: The classifier must be able to adapt to new types of spam emails and changing spam tactics.
- User-friendliness: The classifier must be easy to use and understand for both technical and non-technical users.
- Integration: The classifier must be able to integrate with existing email systems and workflows.
- Data privacy: The classifier must adhere to data privacy regulations and ensure the privacy of users' email data.
- Maintenance: The classifier must be easy to maintain and update to ensure continued accuracy and effectiveness.

2.2.1 Conclusion

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. After surveying the existing systems, finding out the advantages and disadvantages , we have decided to make the travel and tourism management system which overcomes disadvantages of the existing systems to some extent.

Chapter 3. Proposed System

Proposed System

3.1 The Proposal

The proposal is to deploy a system which is designed to be more efficient than other system. The Proposal system is used to detect the mail is spam or ham .The proposed System is completely computer-based application.

3.2 Benefits of the Proposed System

- Improved productivity: By reducing the number of spam emails that users have to manually sift through, email spam classifiers can improve productivity by saving time and allowing users to focus on important emails.
- Enhanced security: Spam emails can contain malicious links or attachments that can compromise the security of an organization's network or systems. By accurately identifying and filtering out these emails, email spam classifiers can enhance the security of an organization's email system.
- Reduced risk of fraud: Many spam emails are designed to trick users into revealing personal or sensitive information, such as login credentials or financial information. By filtering out these emails, email spam classifiers can reduce the risk of users falling victim to phishing or other types of fraud.

3.3 Block Diagram

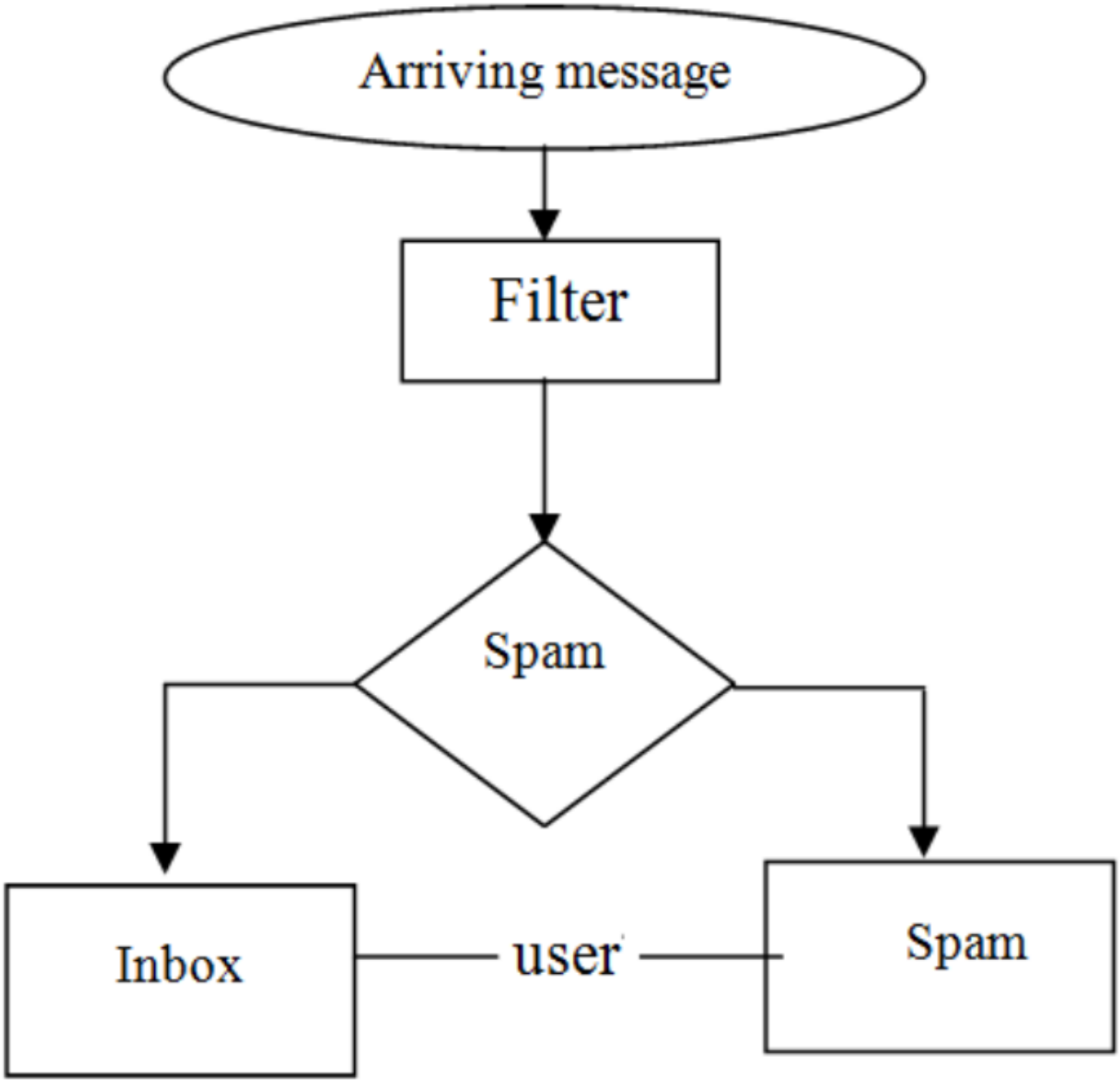

Fig 3-1 : Block Diagram

3.4 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.4.1 Technical

For the S.E.C., there is a need to make a website that gives the user a caption regarding whether the mail is spam or not. For this, Flask is used, which is the framework for Python. Machine learning algorithms are used to predict whether the mail or SMS is spam or not.

.

3.5 Design Representation

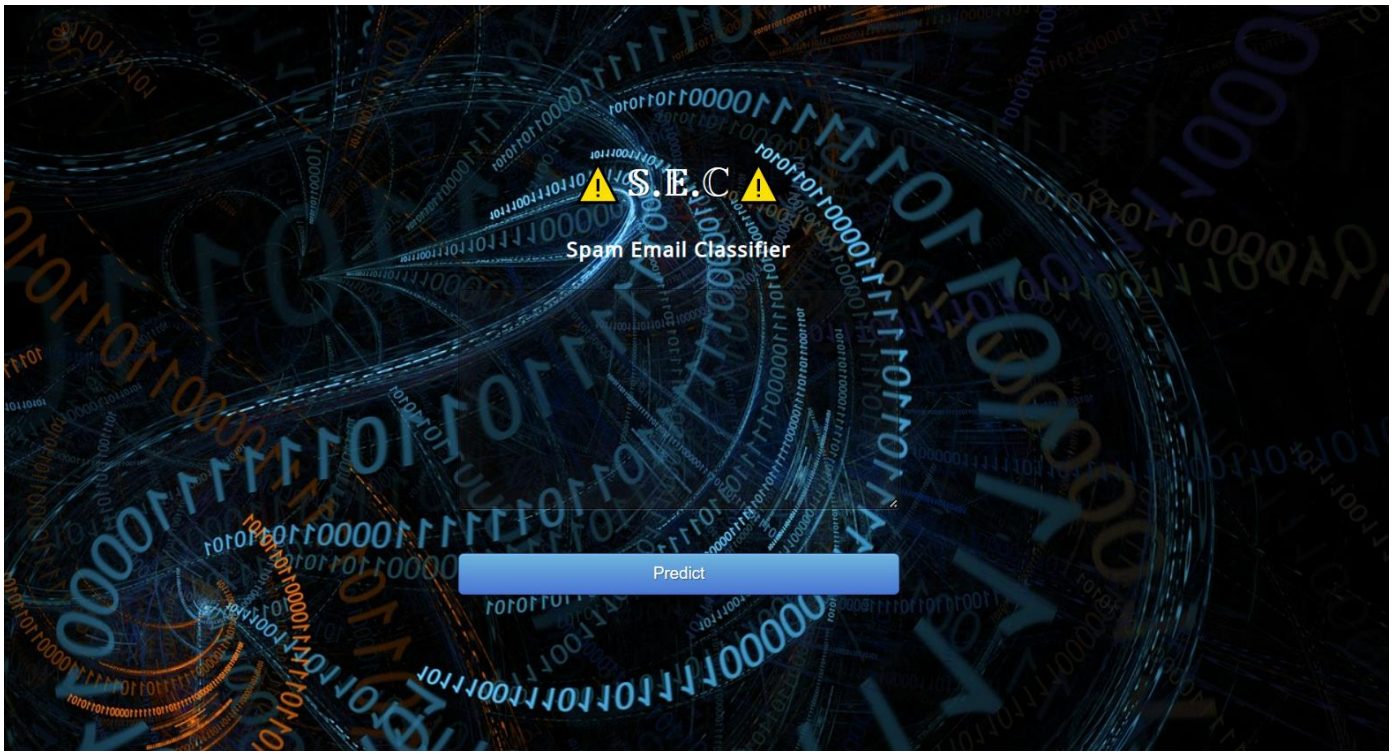

Fig 3.1 : Prediction Page

3.5.1 Class Diagram

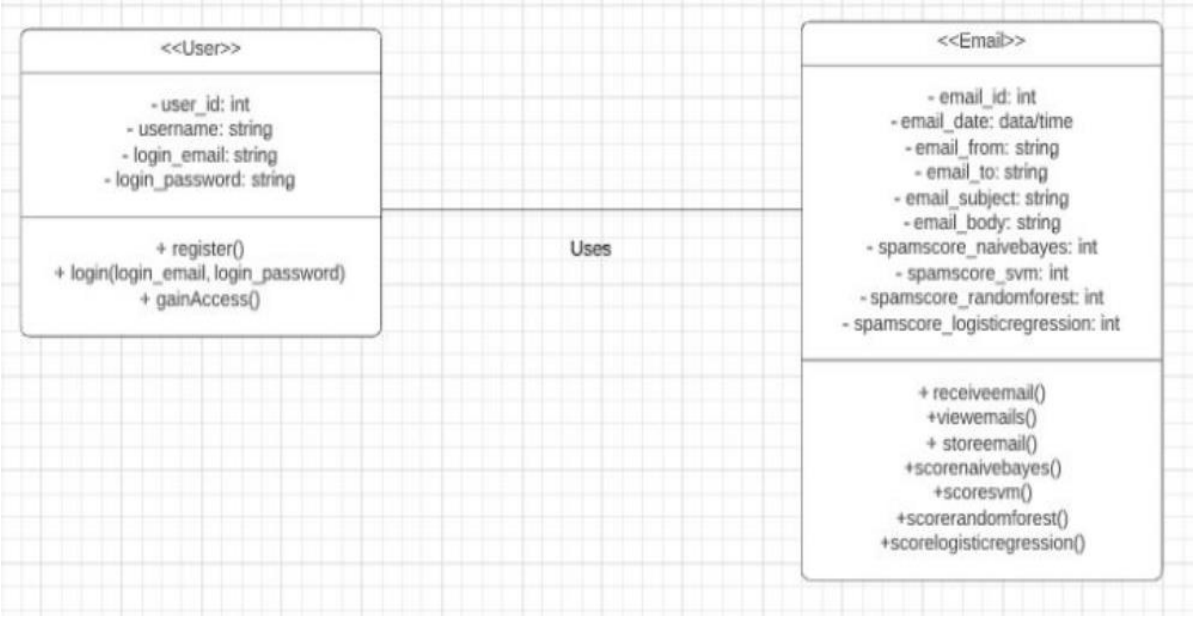

Fig 3.2 : Class Diagram

3.5.2 E R Diagram

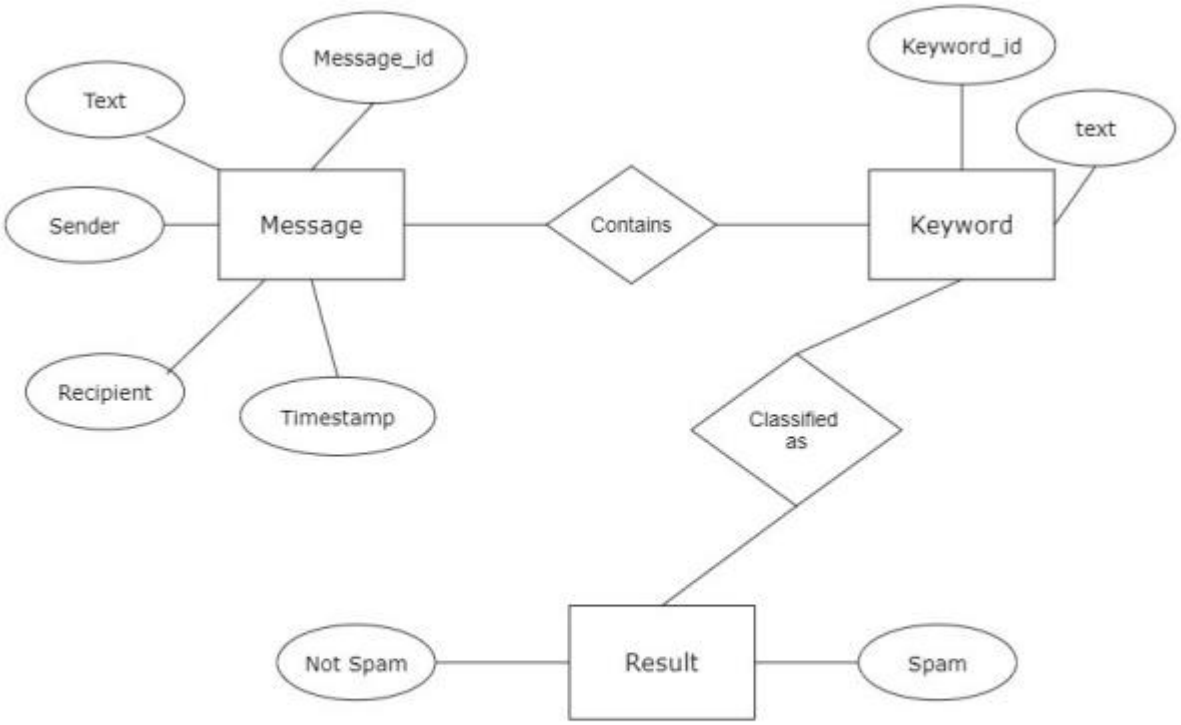

Fig 3-3: ER Diagram

3.5.3 Use Case Diagram

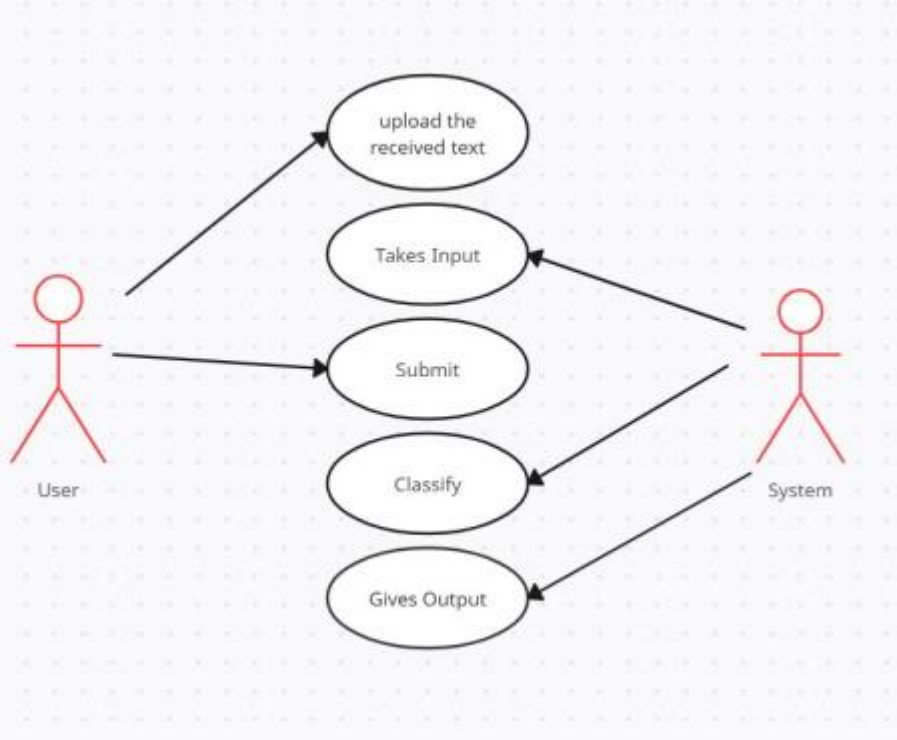

Fig 3-4: Use Case Diagram

3.5.4 Activity Diagram

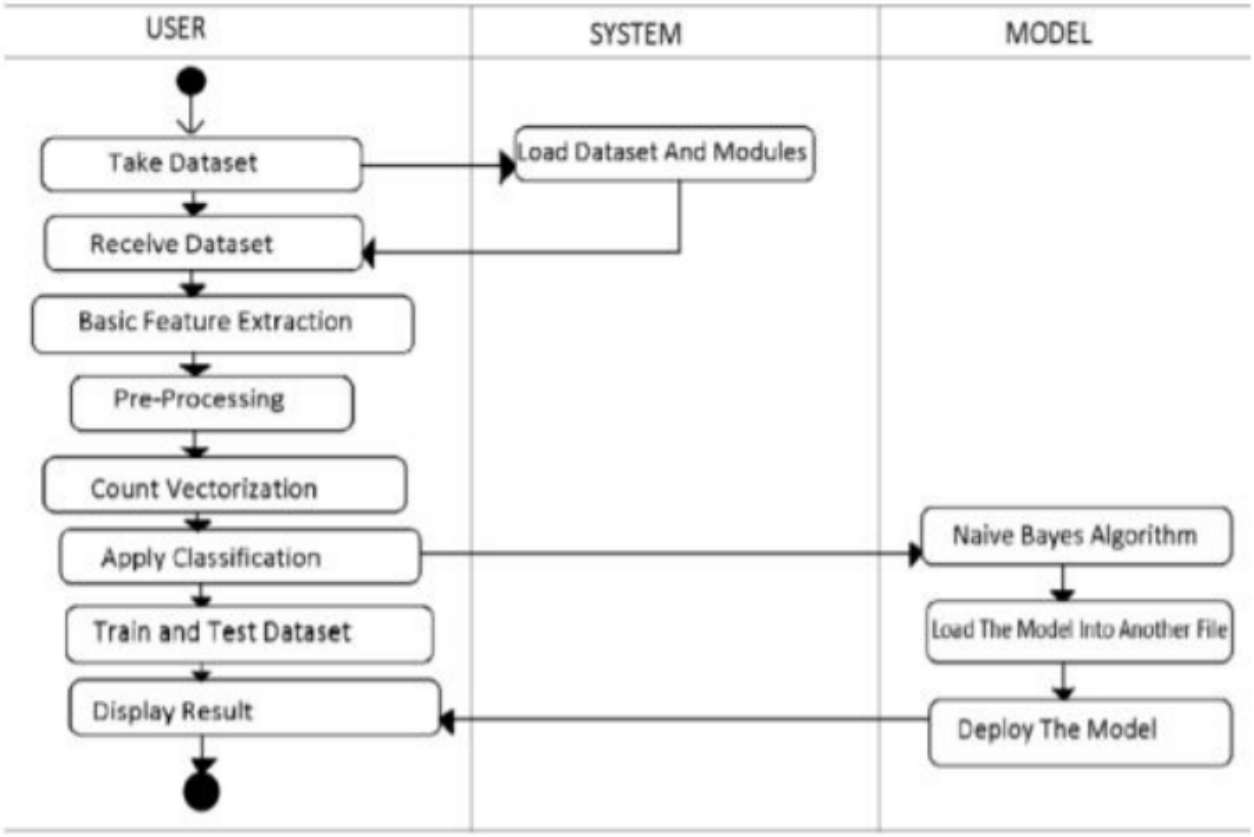

Fig 3-5: Activity Diagram

Chapter 4. Implementation

Implementation

- ❖ Machine learning algorithms: We propose using machine learning algorithms such as Naive Bayes, Random Forest, and Support Vector Machines to classify emails as spam or non-spam based on various factors such as email content, sender information, and attachments.
- ❖ Dataset preparation: We will prepare a large and diverse dataset of labeled emails to train the machine learning models, using publicly available datasets as well as additional data collected from user feedback and crowdsourcing.
- ❖ Feature engineering: We will extract and engineer relevant features from the email data, such as the presence of specific keywords, the sender's reputation, and the email's structure and formatting, to improve the accuracy of the classification models.
- ❖ Model training and optimization: We will train and optimize the machine learning models using the prepared dataset and evaluate their performance using standard evaluation metrics such as precision, recall, and F1 score. We will also fine-tune the models based on user feedback and adjust the spam filtering settings based on users' specific needs.

4.1 Technology Used

4.1.1 Front end-

- **HTML -**It stands for 'HYPERTEXT MARKUP LANGUAGE'. HTML is a standardized system for tagging text files that creates the structure for just about every page that we find and use on the web. It is HTML that adds in page breaks, paragraphs, bold lettering, italics, and more. HTML works to build this structure by using tags that tell browsers what to do with text.

- **CSS –** It stands for 'CASCADING STYLE SHEET'. CSS is used for defining the styles for web pages. It describes the look and formatting of a document which is written in a markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It is easier to make the web pages presentable using CSS. It is easy to learn and understand and used to control the presentation of an HTML document. CSS helps us to control the text color, font style, the spacing between paragraphs, sizing of columns, layout designs, and many more. It is independent of HTML, and we can use it with any XML-based markup language.

  It is recommended to use CSS because the HTML attributes are being deprecated. So, for making HTML pages compatible with future browsers, it is good to start using CSS in HTML pages.

4.1.2 BACK-END –

- **Python –**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

- **Machine Learning -**

Machine learning (ML) is a subset of artificial intelligence (AI) that involves developing algorithms and models that enable computers to learn from data and make predictions or decisions without being explicitly programmed. In other words, machine learning algorithms use statistical techniques to identify patterns and relationships in large datasets, and use this information to make predictions or decisions about new data. The goal of machine learning is to enable computers to learn and improve their performance over time, without the need for human intervention.

- FLASK(Framework)

Flask is a lightweight web application framework written in Python. It is designed to be simple and easy to use, making it a popular choice for developing small to medium-sized web applications. Flask provides basic functionality for handling HTTP requests and responses, as well as templates for rendering dynamic web pages. It also includes built-in support for handling user sessions, authentication, and database integration. Flask is considered a "micro" framework, meaning that it is minimalistic and does not include many of the advanced features of larger web frameworks such as Django. However, Flask is highly extensible and can be easily customized with a variety of third-party extensions to add additional functionality.

4.2 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and to assess the feature of the system. Testing assesses the qualityof the product. It is a process that is done during the development process.

4.2.1 Strategy Used

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

The texting method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

4.2.2 Test Case and Analysis

- **TEST CASE: 1**

Test Case ID	TC001
Test CaseSummary	Spam text Classification, the text will be given as input and then model will classify it as a Spam or Not a spam.
Input Text	Hii I am Atharva puranik I am from CS department.
Expected Result	Not a spam
Actual Result	Not a Spam
Status	PASS

Table 4-1: Test Case 1

TEST CASE 1 INPUT

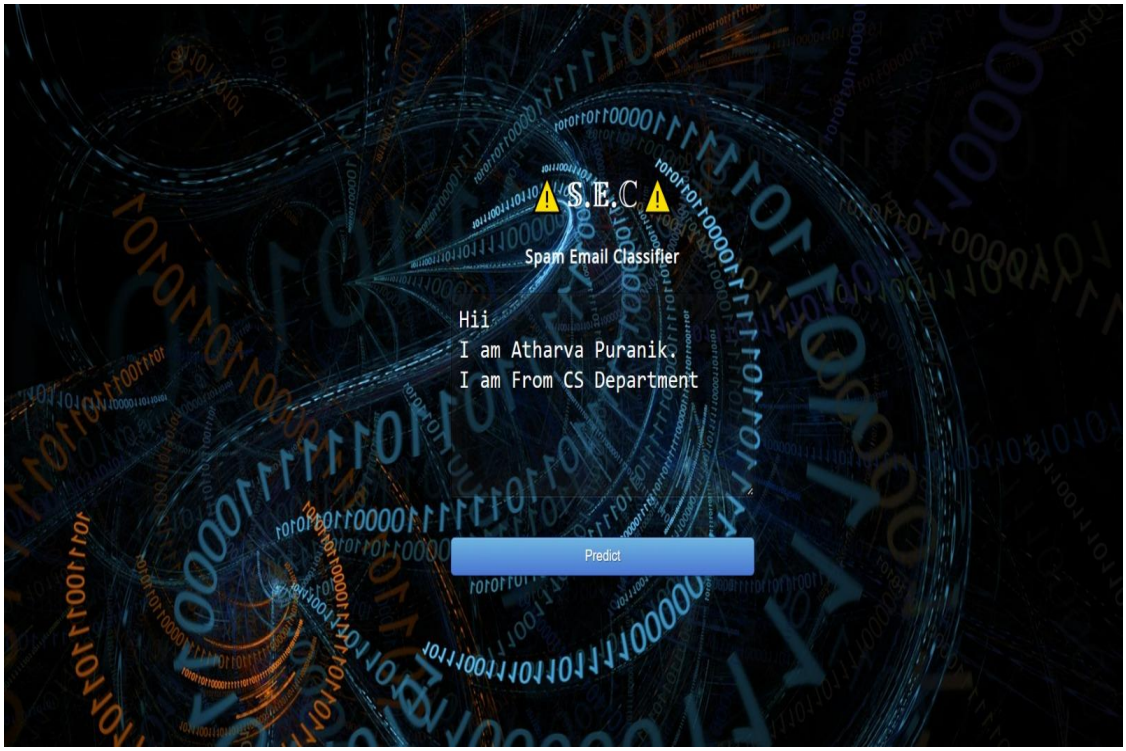

Fig 4-1: Test Case 1 Input

TEST CASE 1 OUTPUT

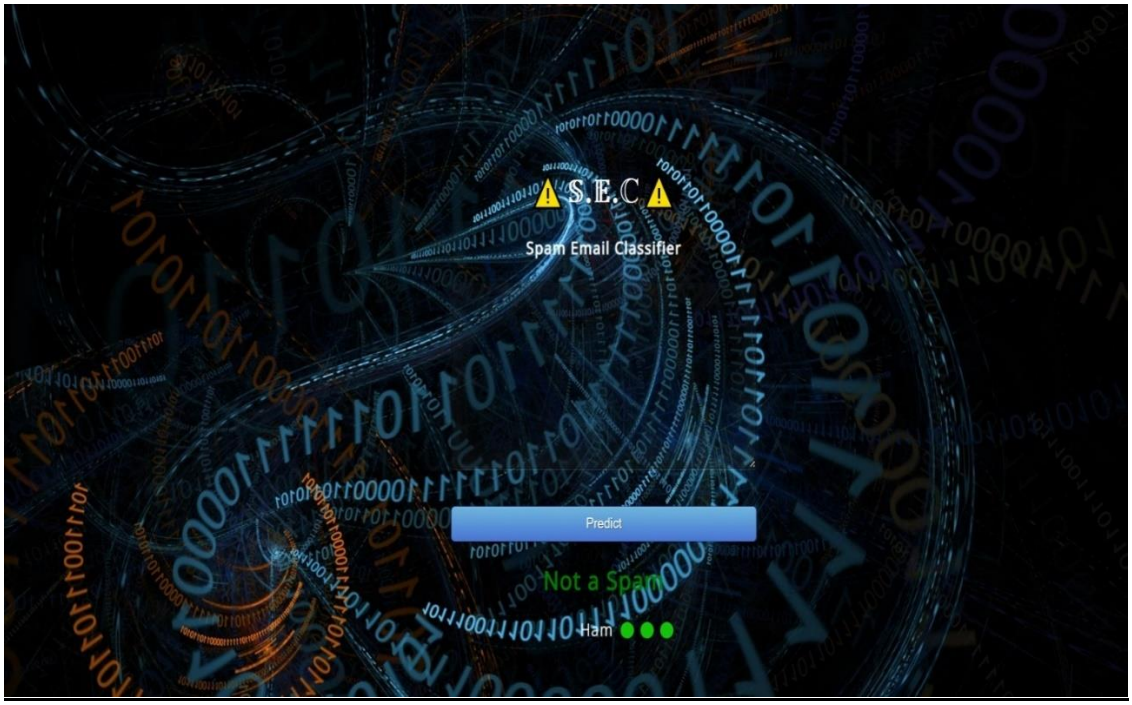

Fig 4-2: Test Case 1 Output

- **TEST CASE: 2**

Test Case ID	TC002
TestCase Summary	Spam text Classification, the text will be given as input and then model will classify it as a Spam or Not a spam.
Input Text	You won 500 rs lottery While scrolling on our page, you can redeem the price by clicking on our link.
Expected Result	Spam
Actual Result	Spam
Status	PASS

Table 4-2: Test Case 2

TEST CASE 2 INPUT

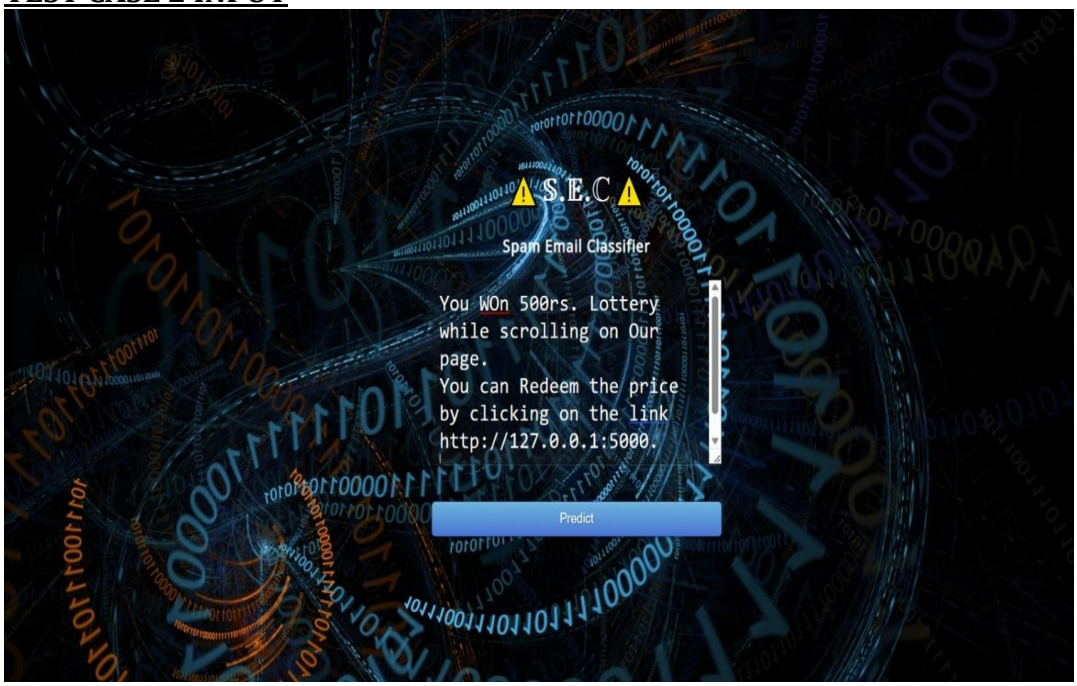

Fig 4-3: Test Case 2 Input

TEST CASE 2 OUTPUT

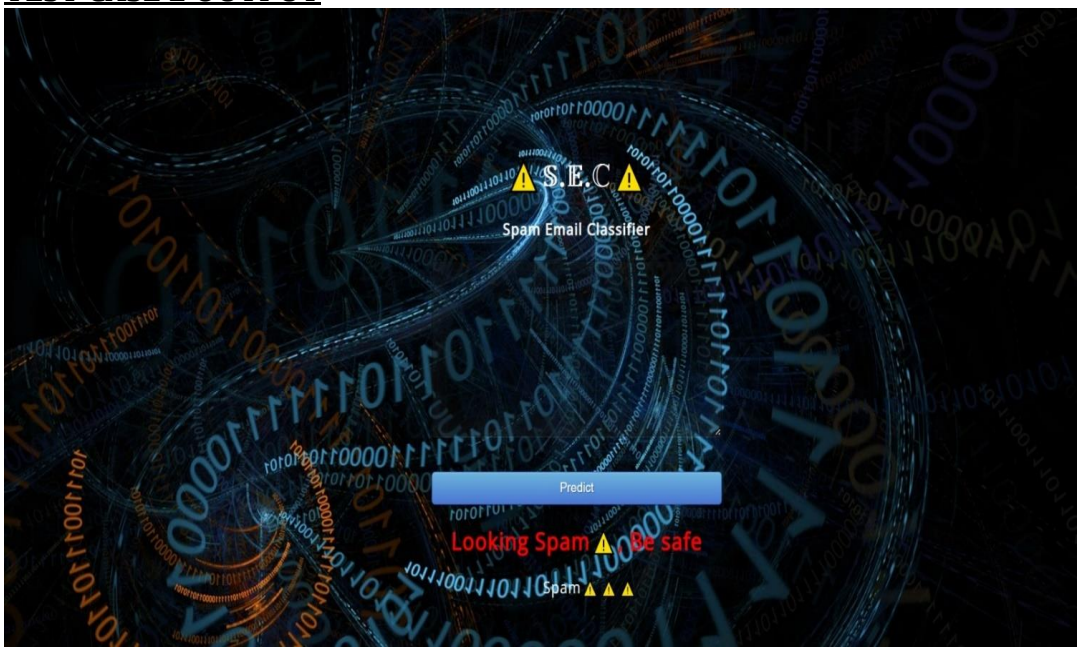

Fig 4-4: Test Case 2 Output

Chapter 5 Conclusion

Conclusion

5.1 Conclusion

Today, email is the most significant form of communication since it allows for the delivery of any message anywhere in the globe thanks to internet connectivity. Every day, more than 270 billion emails are sent and received, of which 57% are spam. Spam emails, often referred to as "non-self," are unwanted commercial or harmful emails that damage or hack personal information like bank accounts, information relating to money, or anything else that causes harm to a single person, a business, or a group of people. In addition to advertisements, they might have connections to websites hosting phishing or malware intended to steal personal data. Spam is a severe problem that end consumers find bothersome but is also financially harmful and a security concern. Therefore, this system is created so that it can identify undesired and unsolicited emails and stop them, aiding in the decrease of spam messages, which would be extremely beneficial to both individuals and the business. In the future, this system may be developed using various algorithms, and it can also get new features added to it.

5.2 Limitations of the Work

1. Limited training data: The effectiveness of a spam text classifier model heavily depends on the quality and quantity of training data. If the model is trained on a limited dataset, it may not be able to accurately identify new and evolving types of spam messages.

2. Language and cultural bias: Spam messages may be written in different languages and use different cultural references, making it difficult for a spam text classifier model to accurately identify them. Language and cultural biases can limit the effectiveness of the model.

3. Contextual understanding: Spam messages may use sophisticated tactics such as

social engineering to bypass spam filters. A spam text classifier model may not have the contextual understanding required to detect such messages.

5.3 Suggestion and Recommendations for Future Work

- Improved accuracy: As the technology for email spam classification continues to improve, the accuracy of the models is expected to increase. This will result in a better filtering of spam emails, reducing the amount of spam that users receive in their inboxes.

- Multilingual support: The ability to classify spam emails in multiple languages will become increasingly important as global communication becomes more prevalent. Developing models that can accurately classify spam emails in different languages will be a major area of research and development.

- Personalized filtering: Machine learning algorithms can be used to develop personalized spam filters for individual users. This can take into account the specific preferences of the user and their email usage patterns to provide more accurate filtering.

- Integration with other email tools: Spam email classifiers can be integrated with other email tools, such as email clients and mobile apps. This can provide a seamless user experience and improve the overall efficiency of email management.

- Real-time classification: Real-time spam email classification can help to prevent spam emails from even reaching the user's inbox. This can be achieved through the use of machine learning algorithms that can quickly and accurately classify emails as spam or not spam.

Bibliography

[1]"Spam Filtering: An Overview" by Andrzej M. J. Skulimowski: This paper provides an overview of various spam filtering
techniques and their performance.
https://ieeexplore.ieee.org/abstract/document/4457923

[2]"A Survey of Techniques for Email Spam Filtering" by A. Sahami, S. Dumais, D. Heckerman, and E. Horvitz: This paper
provides a comprehensive survey of the various techniques used for spam filtering.
https://www.microsoft.com/en-us/research/publication/a-survey-of-techniques-for-email-spam-filtering/

[3]"A Review of Machine Learning Techniques for Spam Email Detection" by N. K. Singh and A. K. Garg: This paper reviews
various machine learning techniques that have been used for spam email detection.
https://link.springer.com/article/10.1007/s10796-018-9906-1

[4]"Spam Detection Using Machine Learning Techniques: A Review" by S. Rawat, S. Singh, and V. Kumar: This paper provides
a review of various machine learning techniques used for spam detection.
https://www.sciencedirect.com/science/article/pii/S2405452618313148

[5]"Real-Time Email Spam Detection Using Machine Learning Techniques" by A. R. Siddique and A. O. E. Abu-Ein: This paper
describes a real-time email spam detection system using machine learning techniques.
https://ieeexplore.ieee.org/abstract/document/7985203

Source Code

App.py -

```
from flask import Flask, render_template, request
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    df = pd.read_csv("spam.csv", encoding="latin-1")
    df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)
    # Features and Labels
    df['label'] = df['class'].map({'ham': 0, 'spam': 1})
    X = df['message']
    y = df['label']
    # Extract Feature With CountVectorizer
    cv = CountVectorizer()
    X = cv.fit_transform(X)  # Fit the Data
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
    # Naive Bayes Classifier
    clf = MultinomialNB()
```

```
    clf.fit(X_train, y_train)
    clf.score(X_test, y_test)
    if request.method == 'POST':
        message = request.form['message']
        data = [message]
        vect = cv.transform(data).toarray()
        my_prediction = clf.predict(vect)
    return render_template('index.html', prediction=my_prediction)
if __name__ == '__main__':
    app.run()
```

---

index.html-

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Spam Detection System</title>
        <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">


</head>


<body>
```

```
<div class="login">
  <h1>⚠ S.E.C ⚠</h1>
  <h1>Spam Email Classifier</h1>

  <form action="{{ url_for('predict')}}" method="POST">
    <textarea  name="message" rows="6" cols="50" required="required"></textarea>
    <br> </br>
    <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>

      <div class="results">

  {% if prediction == 1%}
  <h2 style="color:red;">Looking Spam ⚠, Be safe</h2>
  {% elif prediction == 0%}
        <h2 style="color:green;"><b>Not a Spam ♡</b></h2>
  {% endif %}
  </div>
  </form>
</div>
```

Style.css-

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn {
 display: inline-block;
 *display: inline;
 *zoom: 1;
 padding: 4px 10px 4px;
 margin-bottom: 0;
 font-size: 13px;
```

```
  line-height: 18px;
  color: #333333;
  text-align: center;
  text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
  vertical-align: middle;
  background-color: #f5f5f5;
  background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: -webkit-gradient(
   linear,
   0 0,
   0 100%,
   from(#ffffff),
   to(#e6e6e6)
  );
  background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: -o-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: linear-gradient(top, #ffffff, #e6e6e6);
  background-repeat: repeat-x;
            filter:          progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff,
endColorstr=#e6e6e6, GradientType=0);
  border-color: #e6e6e6 #e6e6e6 #e6e6e6;
  border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);
  border: 1px solid #e6e6e6;
  -webkit-border-radius: 4px;
  -moz-border-radius: 4px;
  border-radius: 4px;
  -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2),
   0 1px 2px rgba(0, 0, 0, 0.05);
  -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2),
```

```
    0 1px 2px rgba(0, 0, 0, 0.05);
  box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2),
    0 1px 2px rgba(0, 0, 0, 0.05);
  cursor: pointer;
  *margin-left: 0.3em;
}
.btn:hover,
.btn:active,
.btn.active,
.btn.disabled,
.btn[disabled] {
  background-color: #e6e6e6;
}
.btn-large {
  padding: 9px 14px;
  font-size: 15px;
  line-height: normal;
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  border-radius: 5px;
}
.btn:hover {
  color: #333333;
  text-decoration: none;
  background-color: #e6e6e6;
  background-position: 0 -15px;
  -webkit-transition: background-position 0.1s linear;
  -moz-transition: background-position 0.1s linear;
  -ms-transition: background-position 0.1s linear;
  -o-transition: background-position 0.1s linear;
```

```
  transition: background-position 0.1s linear;
}
.btn-primary,
.btn-primary:hover {
  text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);
  color: #ffffff;
}
.btn-primary.active {
  color: rgba(255, 255, 255, 0.75);
}
.btn-primary {
  background-color: #4a77d4;
  background-image: -moz-linear-gradient(top, #6eb6de, #4a77d4);
  background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4);
  background-image: -webkit-gradient(
    linear,
    0 0,
    0 100%,
    from(#6eb6de),
    to(#4a77d4)
  );
  background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4);
  background-image: -o-linear-gradient(top, #6eb6de, #4a77d4);
  background-image: linear-gradient(top, #6eb6de, #4a77d4);
  background-repeat: repeat-x;
        filter:  progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0);
  border: 1px solid #3762bc;
  text-shadow: 1px 1px 1px rgba(0, 0, 0, 0.4);
  box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2),
```

```
    0 1px 2px rgba(0, 0, 0, 0.5);
}
.btn-primary:hover,
.btn-primary:active,
.btn-primary.active,
.btn-primary.disabled,
.btn-primary[disabled] {
  filter: none;
  background-color: #4a77d4;
}
.btn-block {
  width: 100%;
  display: block;
}

* {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  -ms-box-sizing: border-box;
  -o-box-sizing: border-box;
  box-sizing: border-box;
}

html {
  width: 100%;
  height: 100%;
  overflow: hidden;
}

body {
```

```
width: 100%;
height: 100%;
font-family: "Open Sans", sans-serif;
background: #ba5858;
color: #fff;
font-size: 18px;
text-align: center;
letter-spacing: 1.2px;
background: -moz-radial-gradient(
  0% 100%,
  ellipse cover,
  rgba(104, 128, 138, 0.4) 10%,
  rgba(138, 114, 76, 0) 40%
 ),
 -moz-linear-gradient(top, rgba(57, 173, 219, 0.25) 0%, rgba(42, 60, 87, 0.4)
     100%),
 -moz-linear-gradient(-45deg, #670d10 0%, #092756 100%);
background: -webkit-radial-gradient(
  0% 100%,
  ellipse cover,
  rgba(104, 128, 138, 0.4) 10%,
  rgba(138, 114, 76, 0) 40%
 ),
 -webkit-linear-gradient(top, rgba(57, 173, 219, 0.25) 0%, rgba(
       42,
       60,
       87,
       0.4
     )
     100%),
```

```
  -webkit-linear-gradient(-45deg, #670d10 0%, #092756 100%);
background: -o-radial-gradient(
   0% 100%,
   ellipse cover,
   rgba(104, 128, 138, 0.4) 10%,
   rgba(138, 114, 76, 0) 40%
  ),
  -o-linear-gradient(top, rgba(57, 173, 219, 0.25) 0%, rgba(42, 60, 87, 0.4)
      100%),
  -o-linear-gradient(-45deg, #670d10 0%, #092756 100%);
background: -ms-radial-gradient(
   0% 100%,
   ellipse cover,
   rgba(104, 128, 138, 0.4) 10%,
   rgba(138, 114, 76, 0) 40%
  ),
  -ms-linear-gradient(top, rgba(57, 173, 219, 0.25) 0%, rgba(42, 60, 87, 0.4)
      100%),
  -ms-linear-gradient(-45deg, #670d10 0%, #092756 100%);
background: -webkit-radial-gradient(
   0% 100%,
   ellipse cover,
   rgba(104, 128, 138, 0.4) 10%,
   rgba(138, 114, 76, 0) 40%
  ),
  linear-gradient(
   to bottom,
   rgba(57, 173, 219, 0.25) 0%,
   rgba(42, 60, 87, 0.4) 100%
  ),
```

```
   linear-gradient(135deg, #670d10 0%, #092756 100%);
      filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#3E1D6D',
endColorstr='#092756',GradientType=1 );
}
.login {
  position: absolute;
  top: 40%;
  left: 50%;
  margin: -150px 0 0 -150px;
  width: 400px;
  height: 400px;
}

.login h1 {
  color: #fff;
  text-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
  letter-spacing: 1px;
  text-align: center;
}

textarea {
  width: 100%;
  margin-bottom: 10px;
  background: rgba(0, 0, 0, 0.3);
  border: none;
  outline: none;
  padding: 10px;
  font-size: 25px;
  color: #fff;
  text-shadow: 1px 1px 1px rgba(0, 0, 0, 0.3);
```

```
  border: 1px solid rgba(0, 0, 0, 0.3);
  border-radius: 4px;
  box-shadow: inset 0 -5px 45px rgba(100, 100, 100, 0.2),
    0 1px 1px rgba(255, 255, 255, 0.2);
  -webkit-transition: box-shadow 0.5s ease;
  -moz-transition: box-shadow 0.5s ease;
  -o-transition: box-shadow 0.5s ease;
  -ms-transition: box-shadow 0.5s ease;
  transition: box-shadow 0.5s ease;
}
input:focus {
  box-shadow: inset 0 -5px 45px rgba(100, 100, 100, 0.4),
    0 1px 1px rgba(255, 255, 255, 0.2);
}
```