Optimizing Spam Filtering with Machine Learning

The project submitted to Smart Internz

By

Murugavel .S

Loganathan .J

Naveen .N

Poongothai .L

Department of Computer Science

ARIGNAR ANNA GOVERNMENT ARTS
COLLEGE - MUSIRI

(Affilited to Bharathidasan University)

Trichy – 621 211

Contents

Optimizing Spam Filtering with Machine Learning

1. INTRODUCTION

1.1 About the Project

SMS spam filter is a software program that helps in detecting and blocking unwanted text messages or SMS that are considered as spam or unsolicited messages. The filter works by analyzing the content of the SMS and comparing it against a pre-defined set of rules or patterns that are associated with spam messages

The use of SMS spam filters has become increasingly important due to the rising number of spam messages being sent to mobile phones. These messages can be annoying and time-consuming to deal with, and may even contain malicious links or scams that can cause harm to the recipient.

SMS spam filters use various techniques to identify and block spam messages. Some of these techniques include keyword-based filtering, machine learning, and sender reputation analysis. Keyword-based filtering involves analyzing the text of the SMS for specific words or phrases that are commonly associated with spam messages. Machine learning involves training the filter using a large dataset of known spam messages and non-spam messages, allowing the filter to learn to identify new spam messages based on patterns in the data. Sender reputation analysis involves examining the reputation of the sender based on their history of sending spam messages, and blocking messages from known spam senders

SMS spam filters are an important tool for protecting mobile phone users from unwanted and potentially harmful messages, and can help to improve the overall user experience when it comes to SMS communication.

1.2 Purpose

The purpose of an SMS spam filter is to protect mobile phone users from unwanted and potentially harmful text messages or SMS. SMS spam messages can be a nuisance, as they can clutter up a user's inbox and waste their time sifting through unwanted messages. Moreover, some SMS spam messages may contain phishing scams, fraudulent offers, or malicious links that can cause harm to the recipient's device or personal information

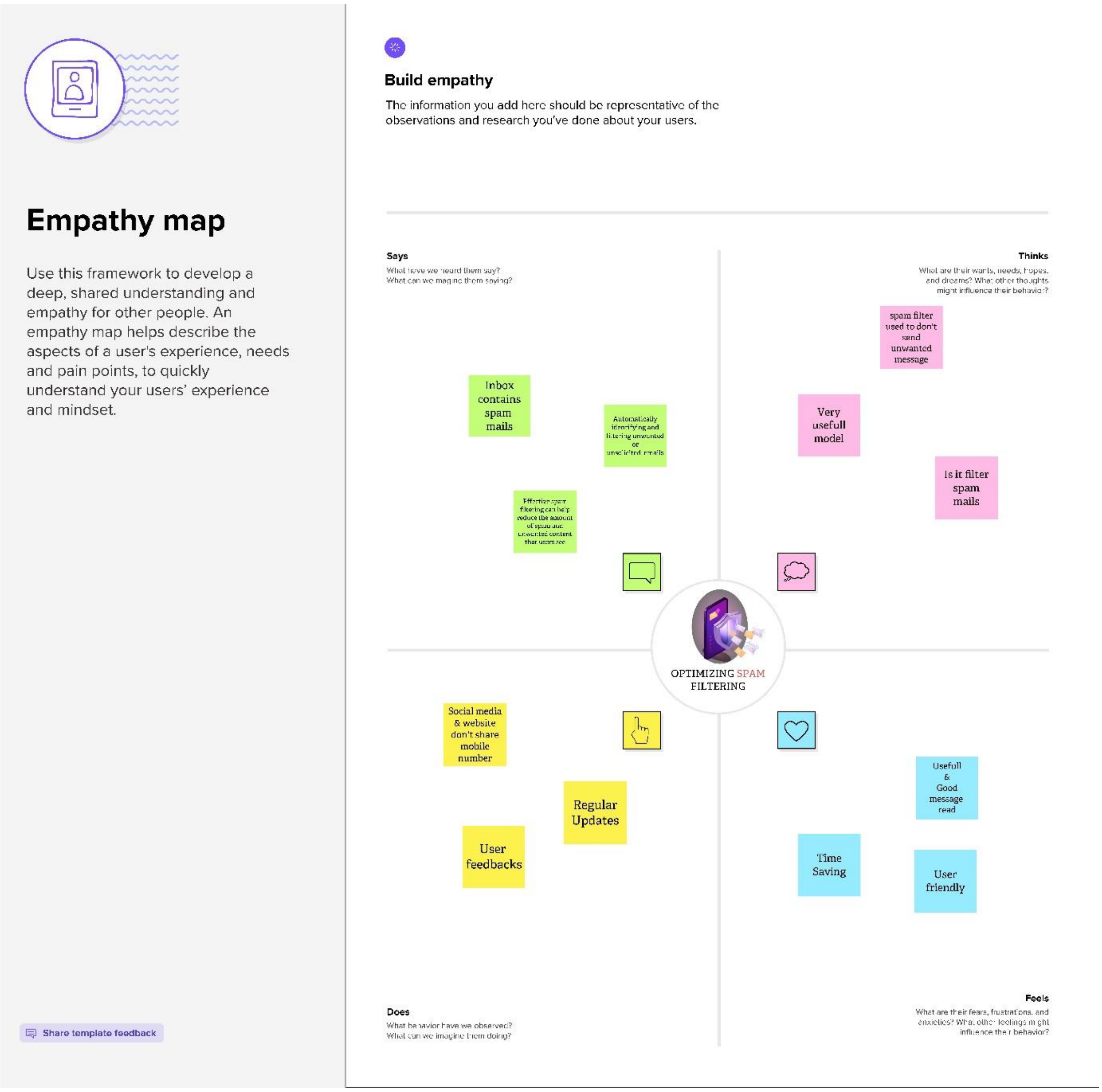
SMS spam filters help to mitigate these risks by analyzing incoming messages and blocking those that are identified as spam or unsolicited. This allows users to focus on legitimate messages and reduces the likelihood of them falling victim to phishing scams or other forms of fraud.

In addition to protecting users, SMS spam filters can also help to improve the overall efficiency of SMS communication. By filtering out unwanted messages, users can more easily find and respond to important messages, which can save time and increase productivity.

2. PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy Map

Using this empathy map, the SMS spam filter can create content that addresses the concerns of mobile phone users and provides solutions to their problems. For example, the filter can emphasize the importance of blocking spam messages to protect against scams and malware, while also highlighting the benefits of having a cleaner inbox and more efficient SMS communication. The content could also include instructions on how to use the filter effectively, and provide reassurance that the user's personal information is safe and secure. By understanding the needs and concerns of mobile phone users, the SMS spam filter can create more effective and engaging content that resonates with its audience.



Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

Share template feedback

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

What have we heard them say?
What can we imagine them saying?

- Inbox contains spam mails
- Automatically identifying and filtering unwanted or unsolicited emails
- Effective spam filtering can help reduce the amount of spam and unwanted content that users see

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

- spam filter used to don't send unwanted message
- Very usefull model
- Is it filter spam mails

OPTIMIZING SPAM FILTERING

Does

What behavior have we observed?
What can we imagine them doing?

- Social media & website don't share mobile number
- Regular Updates
- User feedbacks

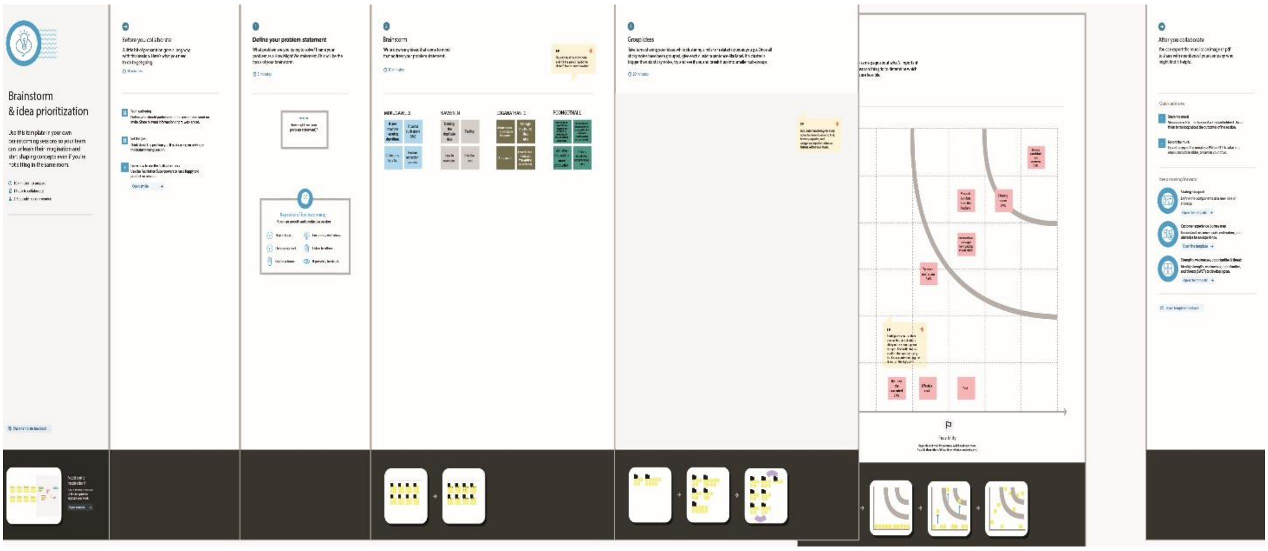
Feels

What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

- Usefull & Good message read
- Time Saving
- User friendly

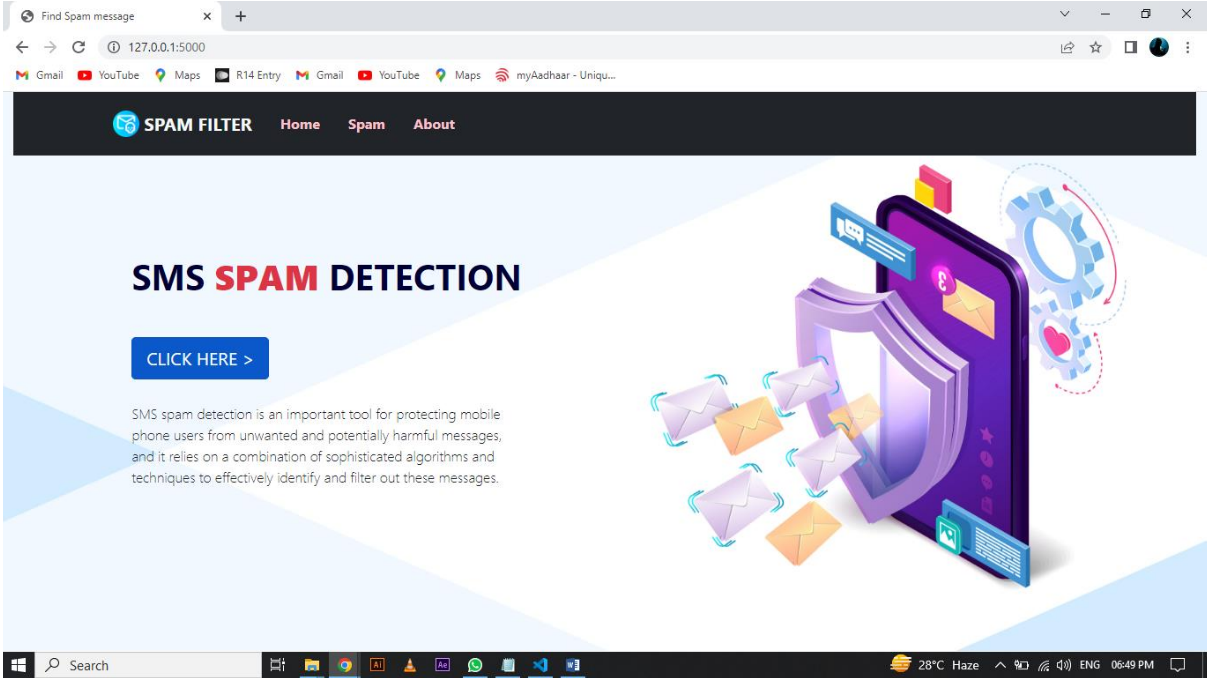
2.2 Ideation & Brainstorming Map

Ideation and brainstorming map, the SMS spam filter can generate and evaluate a range of ideas for improving its functionality and effectiveness, ultimately leading to a better user experience and increased protection against spam and other potential threats.



Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

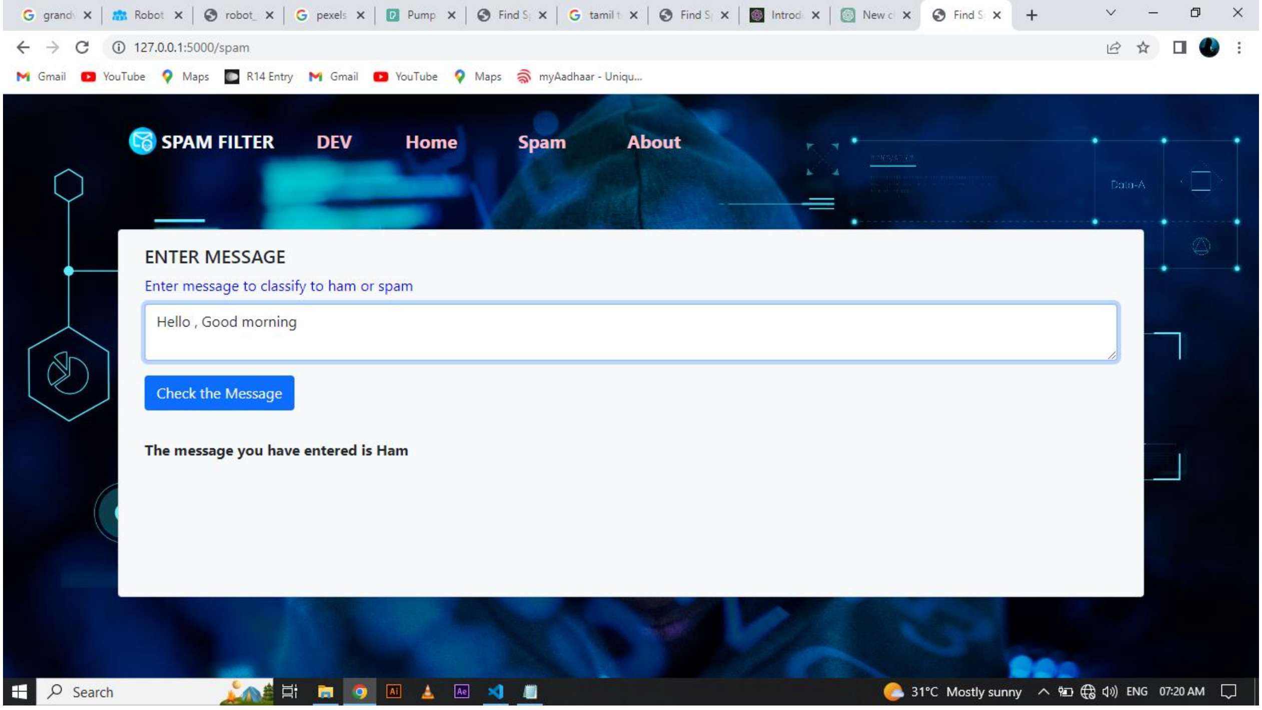
3. RESULT



SMS SPAM DETECTION

SMS spam detection is an important tool for protecting mobile phone users from unwanted and potentially harmful messages, and it relies on a combination of sophisticated algorithms and techniques to effectively identify and filter out these messages.

3.1 Input & Result for Ham Message



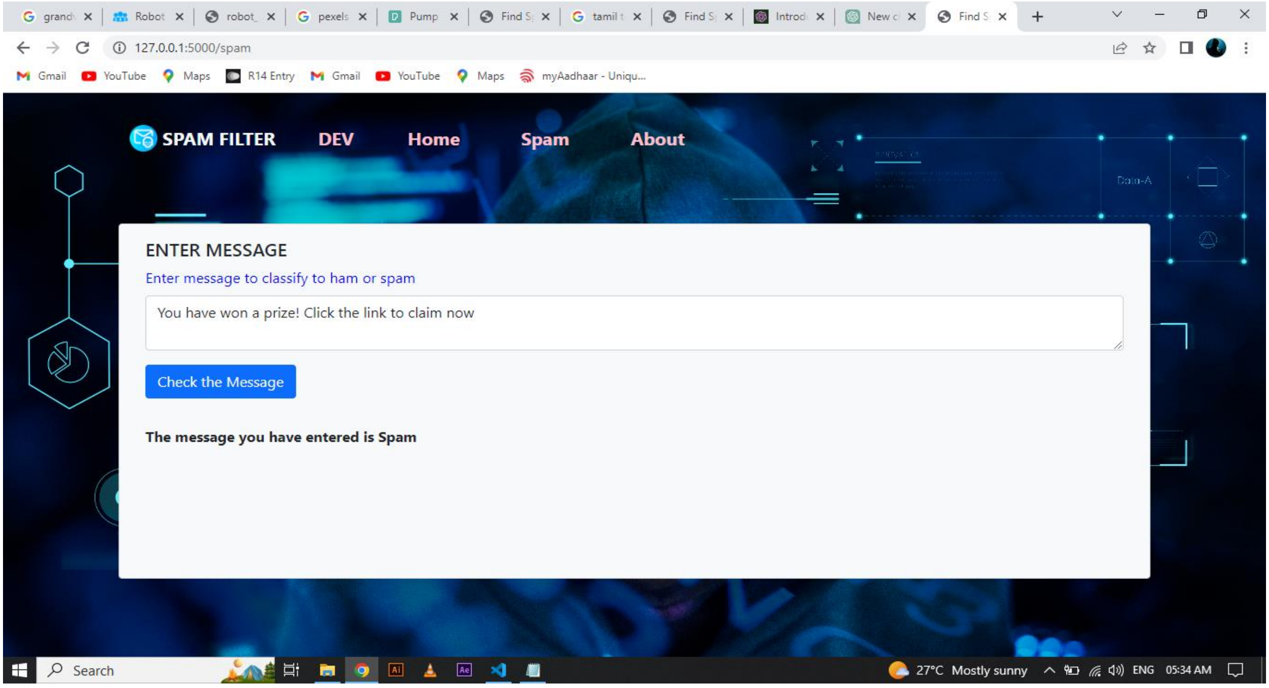
ENTER MESSAGE

Enter message to classify to ham or spam

Hello , Good morning

The message you have entered is Ham

3.2Input & Result for Spam Message



ENTER MESSAGE

Enter message to classify to ham or spam

You have won a prize! Click the link to claim now

The message you have entered is Spam

ADVANTAGES & DISADVANTAGES

Advantages

- ➢ Block unwanted messages: SMS spam filters can effectively block unwanted messages, preventing them from reaching users' inboxes and avoiding the frustration and annoyance that comes with receiving unwanted messages.

- ➢ Improved user experience: By reducing the amount of spam messages received, users have an improved experience with their SMS inbox, making it easier to identify and respond to important messages.

- Increased security: SMS spam filters can prevent phishing scams and other security threats by blocking messages from unverified sources.

Disadvantages

- False positives: SMS spam filters can mistakenly identify legitimate messages as spam, leading to important messages being blocked or filtered out.

- Complexity: Developing and maintaining an effective SMS spam filter can be complex, requiring significant resources and technical expertise.

- Cost: Implementing and maintaining an SMS spam filter can be expensive, particularly for small businesses and individuals.

- Limited effectiveness: While SMS spam filters can be effective, they may not catch all spam messages, particularly those that are well-crafted or targeted.

APPLICATION

- SMS spam filters are typically implemented through software applications or services that analyze incoming messages and determine whether they are spam or legitimate. Some of the common applications used in SMS spam filtering include

- Machine Learning: Machine learning algorithms can be trained on large datasets of messages to identify patterns and characteristics of spam messages. The algorithms can then use this knowledge to classify incoming messages as spam or legitimate.

- Rule-Based Filters: Rule-based filters use a set of predefined rules to identify spam messages. These rules are typically based on specific keywords or phrases commonly found in spam messages.

- Bayesian Filters: Bayesian filters use statistical methods to determine the probability that a message is spam. The filter analyzes the message's content and assigns a score based on the likelihood of it being spam.

- Hybrid Filters: Hybrid filters combine multiple techniques to improve accuracy. For example, a hybrid filter might use a rule-based filter to identify obvious spam messages and a machine learning algorithm to analyze more complex messages.

- Cloud-Based Services: Cloud-based services use large-scale servers and databases to process and filter SMS messages. These services often employ machine learning and other advanced techniques to provide highly accurate spam filtering.

CONCLUSION

SMS spam filters are a critical tool in preventing unwanted messages from reaching users' inboxes. These filters use various techniques and algorithms, such as machine learning and rule-based filtering, to identify and block spam messages. While SMS spam filters provide several advantages, such as improved user experience and increased security, they can also have disadvantages, such as false positives and high costs.

FUTURE SCOPE

- Integration with artificial intelligence (AI): With the increasing use of AI and machine learning in various industries, integrating these technologies into SMS spam filters could improve their accuracy and efficiency.

- Advanced filtering techniques: Future SMS spam filters could incorporate more advanced filtering techniques, such as natural language processing (NLP) and sentiment analysis, to identify and block more complex spam messages.

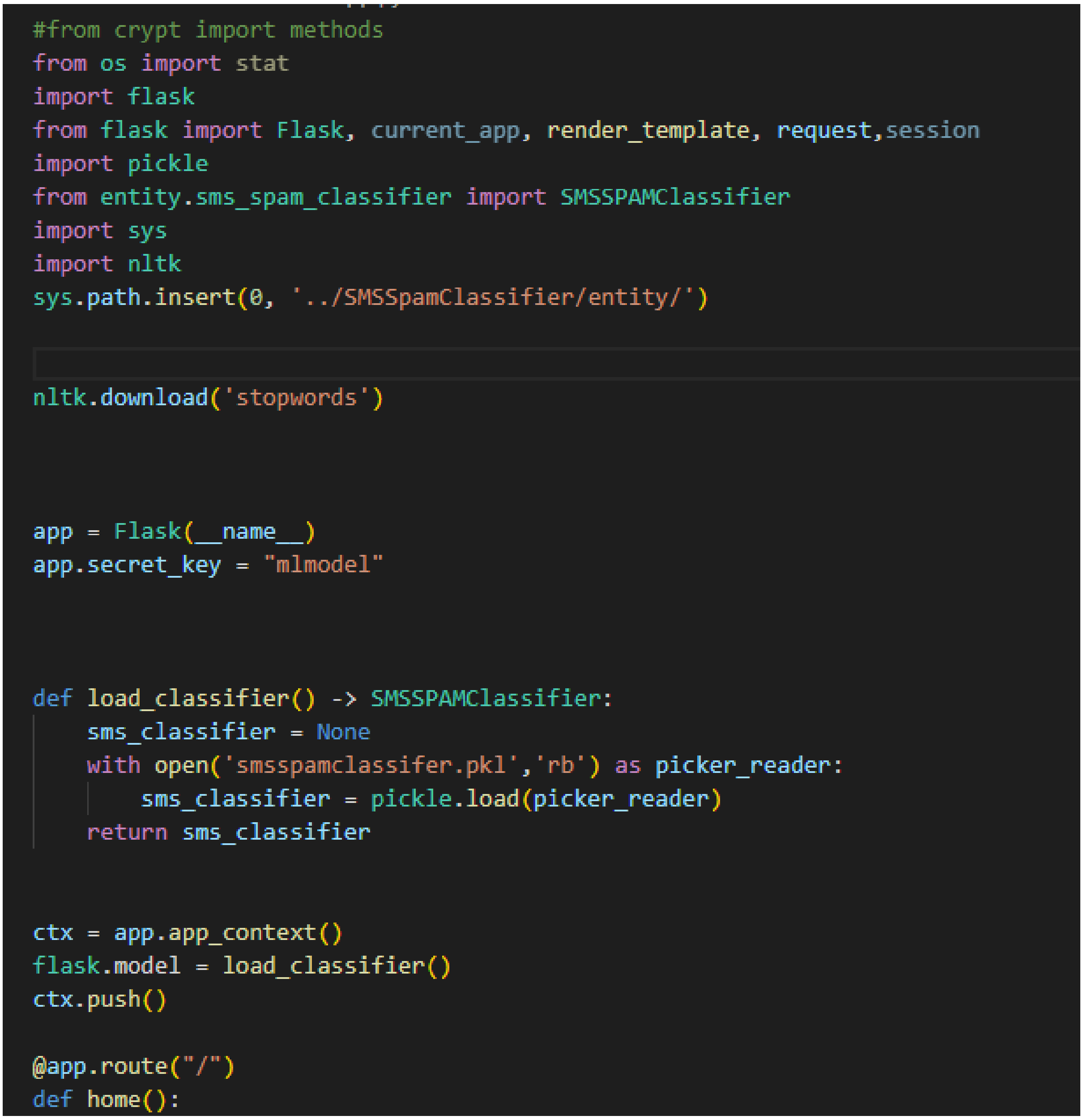
- Personalized filtering: Personalized filtering techniques could be used to customize spam filters based on individual users' preferences and behaviour, resulting in better filtering accuracy.

- Improved collaboration: Improved collaboration between service providers, regulators, and technology vendors could result in more effective spam filtering solutions.

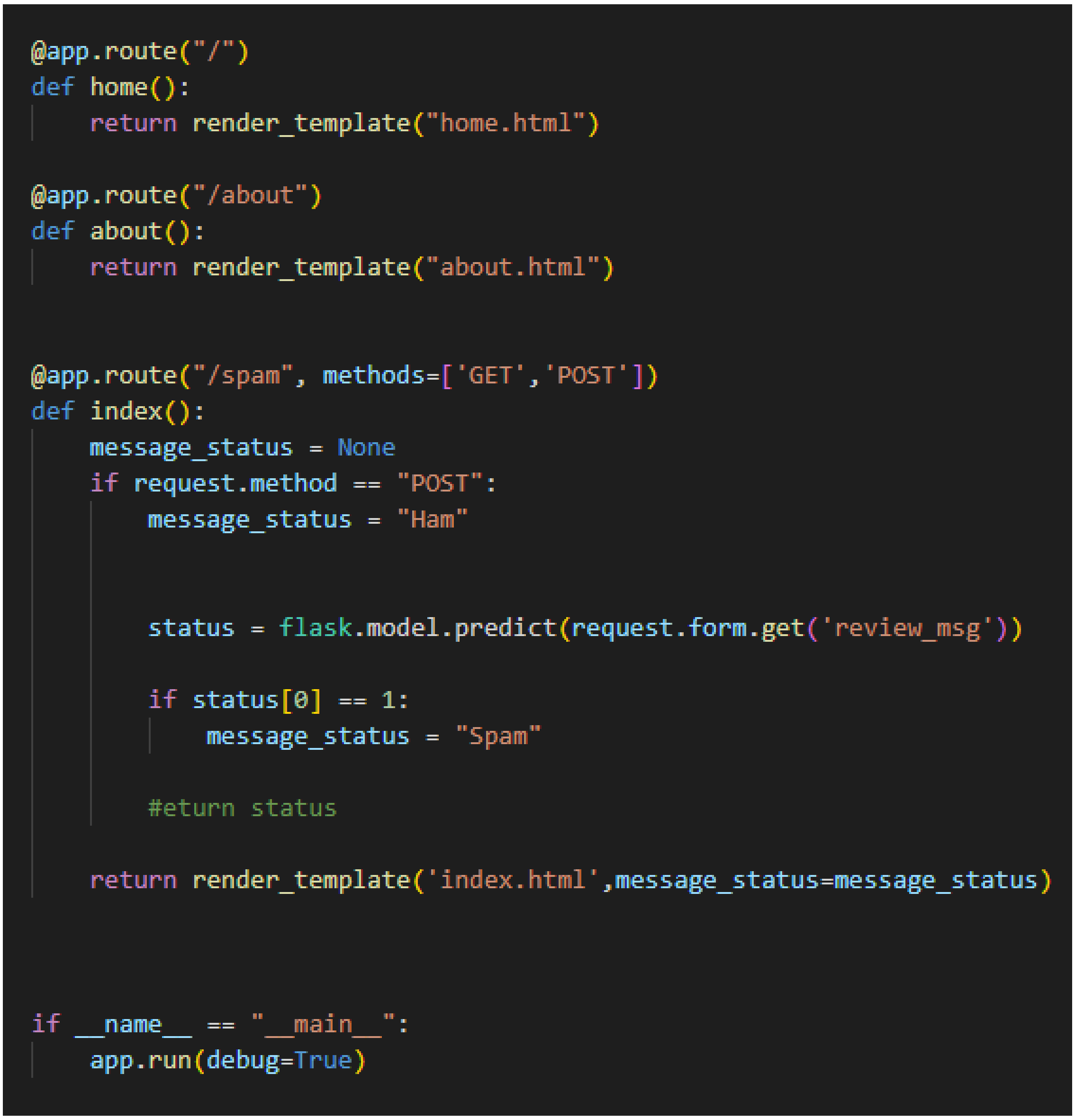
- Blockchain-based filtering: Blockchain technology could be used to create a distributed and transparent system for SMS spam filtering, providing better security and transparency.

APPENDIX

app.py



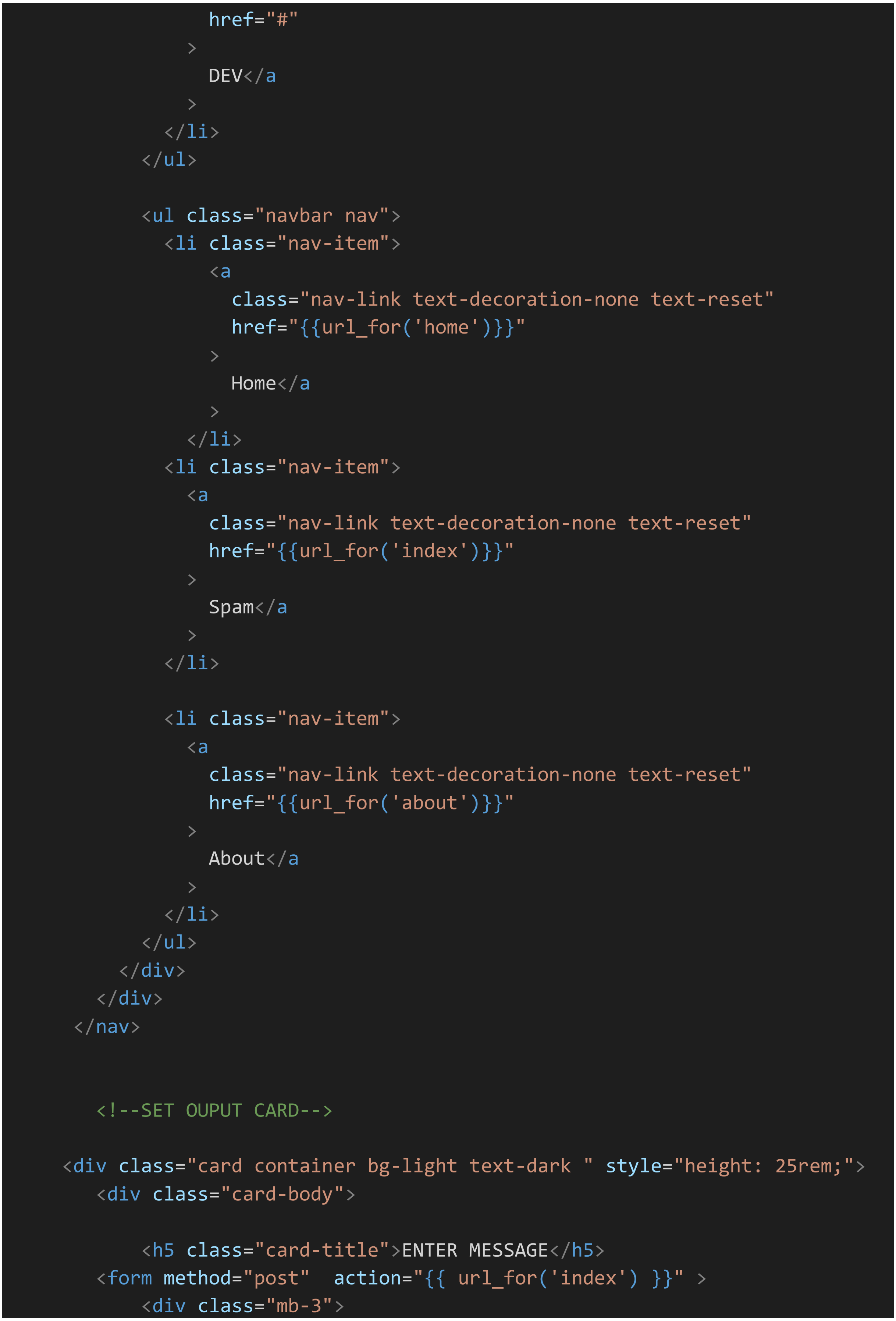
```python
#from crypt import methods
from os import stat
import flask
from flask import Flask, current_app, render_template, request,session
import pickle
from entity.sms_spam_classifier import SMSSPAMClassifier
import sys
import nltk
sys.path.insert(0, '../SMSSpamClassifier/entity/')


nltk.download('stopwords')



app = Flask(__name__)
app.secret_key = "mlmodel"



def load_classifier() -> SMSSPAMClassifier:
    sms_classifier = None
    with open('smsspamclassifer.pkl','rb') as picker_reader:
        sms_classifier = pickle.load(picker_reader)
    return sms_classifier


ctx = app.app_context()
flask.model = load_classifier()
ctx.push()

@app.route("/")
def home():
```



```python
@app.route("/")
def home():
    return render_template("home.html")

@app.route("/about")
def about():
    return render_template("about.html")


@app.route("/spam", methods=['GET','POST'])
def index():
    message_status = None
    if request.method == "POST":
        message_status = "Ham"


        status = flask.model.predict(request.form.get('review_msg'))

        if status[0] == 1:
            message_status = "Spam"

        #eturn status

    return render_template('index.html',message_status=message_status)



if __name__ == "__main__":
    app.run(debug=True)
```

Ham and Spam.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Find Spam message</title>
    <link rel="stylesheet" href="static/css/style.css">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
</head>
<body>
    <div class="container">

     <!--SET NAV BAR-->

    <nav class="navbar navbar-expand-xl navbar-light bg-bg-transparent text-
warning fw-bolder">
        <div class="container">

          <a class="navbar-brand text-light" href="{{url_for('home')}}">
            <img src="static/images/email.png" width="30" height="30"
class="d-inline-block align-top" alt="">
            SPAM FILTER
          </a>

          <button
            class="navbar-toggler bg-light"
            type="button"
            data-bs-toggle="collapse"
            data-bs-target="#navbarNav"
            aria-controls="navbarNav"
            aria-expanded="false"
            aria-label="Toggle navigation"
          >
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
              <li class="nav-item">
                <a
                  class="nav-link active text-decoration-none text-reset"
```



```
             href="#"
           >
             DEV</a
           >
         </li>
       </ul>

       <ul class="navbar nav">
         <li class="nav-item">
             <a
               class="nav-link text-decoration-none text-reset"
               href="{{url_for('home')}}"
             >
               Home</a
             >
           </li>
         <li class="nav-item">
           <a
             class="nav-link text-decoration-none text-reset"
             href="{{url_for('index')}}"
           >
             Spam</a
           >
         </li>

         <li class="nav-item">
           <a
             class="nav-link text-decoration-none text-reset"
             href="{{url_for('about')}}"
           >
             About</a
           >
         </li>
       </ul>
     </div>
   </div>
 </nav>


   <!--SET OUPUT CARD-->

<div class="card container bg-light text-dark " style="height: 25rem;">
   <div class="card-body">

       <h5 class="card-title">ENTER MESSAGE</h5>
   <form method="post"  action="{{ url_for('index') }}" >
       <div class="mb-3">
```

```
            <label for="exampleInputEmail1" class="form-label fw-bold">Enter
message to classify to ham or spam</label>
            <textarea type="text" class="form-control" name="review_msg"
id="id_message" placeholder="Enter your message"></textarea>

            </div >
            <div class="d-flex align-items-center">
                <button type="submit" class="btn btn-primary">Check the
Message</button>
            </div>
        </form>

        {% if message_status %}
        <div class="fs-6 fw-bold output ">
            The message you have entered is {{message_status}}
        </div>
        {% endif %}
    </div>
</div>
</div>
</div>
</div>


<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js
" integrity="sha384-
7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>
</body>
</html>
```