RANI CHANNAMMA UNIVERSITY, BELAGAVI

VIDYASANGAMA

B.L.D.E ASSOCIATION'S

S.B Arts & K.C.P Science College Vijayapura,586103

BACHELOR OF COMPUTER APPLICATIONS

(Affiliated to Rani Channamma University, Belagavi)

A PROJECT ON

"SMS SPAM DETECTION USING AI"

Submitted in partial fulfilment of requirement for the award of the degree

Internal Guide
Prof S.D. PATIL

Submitted By

Shruti Badagandi
Sakshi Ambali

RANI CHANNAMMA UNIVERSITY, BELAGAVI

VIDYASANGAMA

B.L.D.E., ASSOCIATION'S

S.B Arts & K.C.P Science College Vijayapura,586103

BACHELOR OF COMPUTER APPLICATIONS

(Affiliated to Rani Channamma University, Belagavi)

A PROJECT ON

"SMS Spam Detection Using AI"

CERTIFICATE

This is to certify that the project work entitled "SMS Spam Detection Using AI" is a bonafide work carried out by **Shruti Badagandi(U15KM21S0519)** and **Sakshi Ambali(U15KM21S0541)**submitted in the fulfilment for the award of the degree of Bachelor of Computer Application, prescribed by the Rani Channamma University, Belagavi during the academic year 2023-2024.

GUIDE	COORDINATOR	PRINCIPAL
Prof S.D. PATIL	Prof. S.D. PATIL	DR.R.M. MIRDHE

ACKNOWLDGEMENT

The successful presentation of this project is an acknowledgment of the immense support expended by **S.B ARTS AND K.C.P SCIENCE COLLEGE VIJAYAPURA** with has provided an opportunity to fulfil the most cherished desire to reach my goal.

We are extremely grateful to the **Principal R.M. MIRDHE** and **Head of dept. Prof .S.D. PATIL**, of Bachelor of computer application, for providing all the required resources for the successful completion of my Project.

Our heartfelt gratitude to our **project guide Prof. S.D. PATIL**, Bachelor of computer application of Department, for her valuable suggestions and guidance in the preparation and completion of the project report and also for their constant encouragement and support rendered to us throughout the course of work. Also for all the help and co-ordination extended in bringing out this project successfully in time.

Thanking You,

Shruti Badagandi
Sakshi Ambali

TABLE OF CONTENTS

Abstract

The proliferation of mobile devices has led to a significant increase in SMS (Short Message Service) communication, which in turn has resulted in a rise in unwanted spam messages. Unlike other messaging platforms that require an active internet connection, SMS is vulnerable to spam attacks, which can be harmful to users by distributing malicious content or leading to financial losses.

This project proposes an AI-based approach to detect and filter out spam SMS messages using advanced machine learning techniques. The system utilizes various algorithms such as Naive Bayes, Support Vector Machines (SVM), and Random Forests to classify messages as spam or non-spam based on their content and other relevant features. By incorporating feature engineering techniques and rigorous training on labeled datasets, the system aims to achieve high accuracy in spam detection.

This project contributes to enhancing mobile communication security by providing a reliable method for users to avoid spam messages, thereby preserving the integrity and efficiency of SMS communication.

CHAPTER 1: INTRODUCTION

1.1 Overview of SMS Communication

SMS, or Short Message Service, is a widely used communication method that allows users to send and receive text messages on mobile devices. The simplicity and ubiquity of SMS have made it an essential tool for personal and business communication. However, its open nature has also made it susceptible to exploitation by spammers, who send unsolicited and often harmful messages to users. Unlike other messaging platforms that require internet connectivity, SMS can function in offline environments, making it a convenient but vulnerable communication channel.

1.2 Importance of SMS Spam Detection

The rise in SMS spam presents significant challenges for mobile users and service providers. Spam messages can lead to security breaches, financial losses, and a degraded user experience. For individuals, spam messages can be a source of annoyance, but more critically, they can be used to disseminate phishing attacks, spread malware, or conduct fraudulent activities. For businesses, spam can compromise customer trust and result in legal and regulatory repercussions. Therefore, effective spam detection and filtering are crucial for maintaining the security and reliability of SMS as a communication medium.

1.3 Project Objective

The primary objective of this project is to develop a robust and accurate SMS spam detection system using AI techniques. The system will be designed to classify SMS messages as either spam or non-spam (ham) by analyzing their content, structure, and other relevant features. By employing machine learning algorithms such as Naive Bayes, SVM, and Random Forests, the project aims to create a model that can accurately identify spam messages with minimal false positives. The ultimate goal is to enhance the security and user experience of SMS communication by providing a reliable tool to filter out unwanted spam messages.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview of Existing Systems

SMS spam detection has been a critical area of research due to the increasing prevalence of unsolicited messages that affect user privacy and security. Traditional spam detection methods, primarily developed for email, such as keyword filtering and blacklist approaches, have been adapted for SMS. However, these methods often fall short due to the differences between email and SMS, such as the brevity of messages and the lack of structured data. Additionally, existing SMS spam detection systems rely on machine learning algorithms, including Naive Bayes, Support Vector Machines (SVM), and Decision Trees. These models classify messages based on features like word frequency, message length, and the presence of specific keywords.

Several studies have explored the use of Natural Language Processing (NLP) techniques to improve the accuracy of spam detection. For instance, the use of tokenization and vectorization methods like Term Frequency-Inverse Document Frequency (TF-IDF) has been widely adopted to convert text data into a format suitable for machine learning models. More advanced techniques, such as word embeddings and deep learning models, have also been proposed to capture the semantic meaning of the messages. Despite these advancements, challenges remain, particularly in handling imbalanced datasets, where the number of non-spam (ham) messages significantly outweighs the number of spam messages. This imbalance can lead to biased models that are less effective at detecting spam.

2.2 Comparison with Proposed System

The proposed system builds on the strengths of existing systems while addressing their limitations. Unlike traditional keyword-based approaches, our system utilizes a combination of machine learning algorithms, including Naive Bayes, SVM, and Random Forests, to improve classification accuracy. By employing feature engineering techniques, such as TF-IDF and n-gram analysis, the proposed system can capture more nuanced patterns in the data, leading to better performance in spam detection.

One key advantage of the proposed system is its ability to handle imbalanced datasets more effectively. This is achieved through techniques like oversampling and undersampling, which help balance the distribution of spam and ham messages in the training data. Additionally, the use of ensemble methods, such as bagging and boosting, further enhances the model's robustness by combining the predictions of multiple classifiers. The proposed system also incorporates real-time deployment considerations, making it suitable for practical applications where timely detection of spam is crucial.

Chapter 3: System Analysis

3.1 Existing System Analysis

Existing SMS spam detection systems often rely on simplistic methods, such as keyword filtering and rule-based approaches, which can lead to high false positive rates. These systems are not well-equipped to handle the evolving nature of spam messages, which can vary widely in content and structure. Machine learning-based approaches have improved detection rates, but they still face challenges such as the need for extensive labeled datasets and the difficulty of extracting relevant features from short text messages.

The primary issues with existing systems include:

- **Limited Feature Extraction:** Traditional systems often focus on basic features like word frequency or the presence of specific keywords, which may not be sufficient to capture the complexity of modern spam messages.
- **Imbalanced Datasets:** Many existing systems struggle with imbalanced datasets, where spam messages are significantly outnumbered by ham messages. This can lead to models that are biased towards non-spam classification.
- **Lack of Real-Time Capabilities:** Some systems are not designed for real-time detection, which is essential for preventing spam from reaching users.

3.2 Proposed System

The proposed system addresses these challenges by leveraging advanced machine learning techniques and feature engineering methods. The system is designed to be robust, scalable, and capable of real-time spam detection. Key components of the proposed system include:

- **Advanced Feature Engineering:** The system extracts a wide range of features from SMS messages, including word frequencies, n-grams, message length, and the presence of specific phrases. These features are transformed using techniques like TF-IDF and word embeddings to capture the semantic meaning of the text.
- **Machine Learning Models:** The system utilizes multiple machine learning algorithms, including Naive Bayes, SVM, and Random Forests. These models are trained on labeled datasets and evaluated using metrics like precision, recall, and F1-score to ensure high accuracy in spam detection.
- **Handling Imbalanced Data:** The system employs techniques such as oversampling and undersampling to balance the training data, reducing the risk of biased models. Additionally, ensemble methods like bagging and boosting are used to improve the model's generalization ability.
- **Real-Time Detection:** The system is designed for real-time deployment, allowing it to detect and filter out spam messages as they arrive. This is achieved through the use of lightweight frameworks like Flask or FastAPI, which facilitate the integration of the model into existing mobile communication systems.

Chapter 4: System Requirements Specification

4.1 Software Requirements

The proposed SMS spam detection system relies on several software tools and libraries, essential for developing, training, and deploying machine learning models. The software requirements include:

- **Python 3.x:** The primary programming language used for developing the system.
- **Scikit-learn:** A machine learning library in Python for implementing algorithms like Naive Bayes, SVM, and Random Forest.
- **Pandas and NumPy:** Libraries for data manipulation and numerical operations.
- **NLTK and TextBlob:** Libraries for natural language processing, including tokenization, stemming, and sentiment analysis.
- **Flask/FastAPI:** Lightweight web frameworks for deploying the model in real-time.
- **Jupyter Notebook:** An interactive environment for developing and testing machine learning models.

4.2 Hardware Requirements

The hardware requirements are based on the need to efficiently process and classify SMS messages, particularly during the model training phase:

- **Processor:** A multi-core processor, such as Intel i5 or higher, to handle computationally intensive tasks.
- **Memory:** At least 8 GB of RAM, with 16 GB recommended for smoother operation during model training.
- **Storage:** A minimum of 500 GB of storage, preferably SSD, to store datasets, models, and logs.
- **GPU (Optional):** A GPU with CUDA support, such as NVIDIA GTX 1050 or higher, to accelerate deep learning tasks if needed.

4.3 Functional Requirements

The system must fulfill the following functional requirements:

- **Message Input:** The system should accept SMS text messages as input.
- **Spam Detection:** The system should classify each SMS as either spam or ham.
- **Result Output:** The system should output the classification result, indicating whether the message is spam or not.
- **Model Training:** The system should support the training of machine learning models on labeled datasets.
- **Real-time Detection:** The system should be capable of processing and classifying incoming messages in real-time.

4.4 Non-Functional Requirements

The non-functional requirements focus on the system's performance, usability, and reliability:

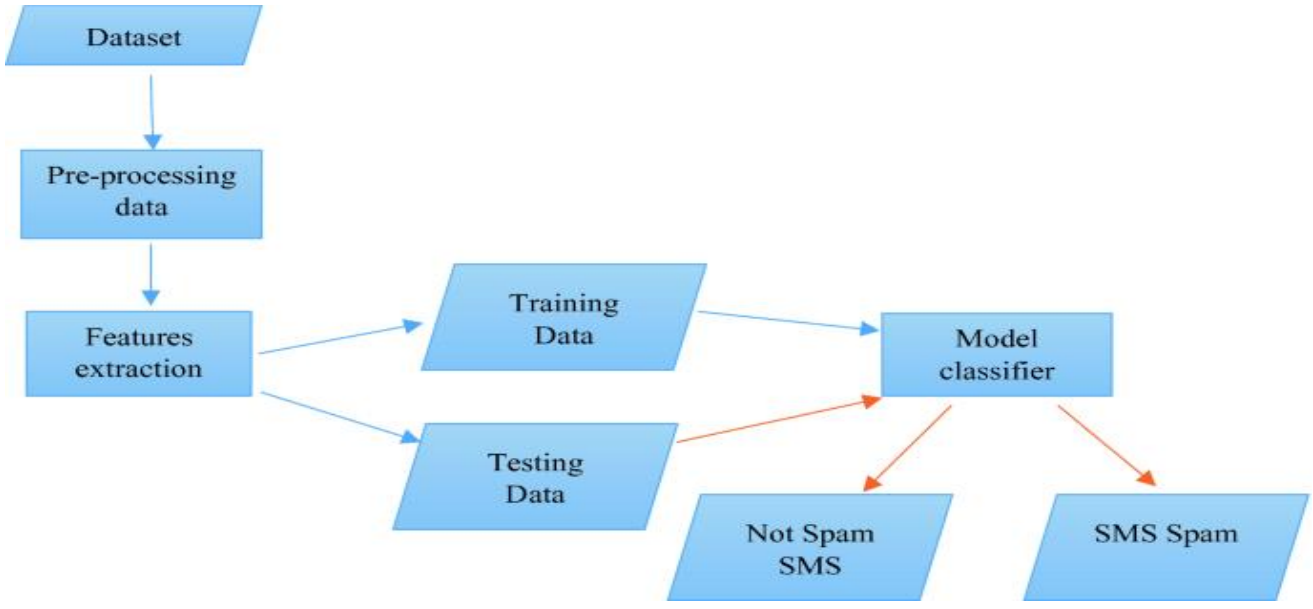
- **Accuracy:** The system should achieve a high accuracy rate, with a target of over 90% in spam detection.
- **Scalability:** The system should be scalable to handle a large volume of messages without significant performance degradation.
- **Usability:** The system's user interface should be intuitive and easy to use.
- **Security:** The system must ensure the confidentiality and integrity of user data.
- **Reliability:** The system should be reliable and maintain consistent performance over time.

Chapter 5: System Design and Implementation

5.1 System Architecture

The architecture of the proposed SMS spam detection system is divided into several components, each responsible for a specific task. The architecture includes:

- **Data Collection Module:** Responsible for gathering SMS data, either through manual entry or integration with an SMS gateway.
- **Preprocessing Module:** Handles the cleaning and preparation of SMS data, including tokenization, stopword removal, and feature extraction.
- **Machine Learning Module:** Implements the core algorithms (Naive Bayes, SVM, Random Forest) for spam detection. This module also includes the model training and evaluation pipeline.
- **User Interface Module:** Provides a front-end interface for users to input messages and view results.
- **Deployment Module:** Manages the real-time processing of SMS messages, typically through a web service using Flask or FastAPI.



5.2 Database Design

The system requires a database to store SMS messages, labels, and model training data. A relational database such as MySQL or PostgreSQL can be used. The database schema includes tables for:

- **Messages:** Storing SMS content, sender information, and timestamps.
- **Labels:** Storing the classification results (spam or ham) for each message.
- **Model Data:** Storing information about trained models, including parameters and performance metrics.

5.3 User Interface Design

The user interface should be simple and intuitive, allowing users to:

- **Input SMS:** Enter or upload SMS text for classification.
- **View Results:** Display whether the message is classified as spam or ham.
- **Access Reports:** Generate reports on the performance of the spam detection system.

5.4 Implementation Details

The system is implemented in Python, with the following key steps:

- **Data Preprocessing:** Tokenization and vectorization of SMS data using TF-IDF.
- **Model Training:** Training Naive Bayes, SVM, and Random Forest models on labeled datasets.
- **Model Evaluation:** Using cross-validation to evaluate model performance.
- **Real-time Deployment:** Integrating the model with a web service for real-time SMS spam detection.

Chapter 6: Methodology

6.1 Development Tools and Environment

The development environment includes Python 3.x as the primary programming language, with Jupyter Notebook used for developing and testing models. Libraries such as Scikit-learn, Pandas, and NLTK are used for machine learning and data processing. The Flask or FastAPI framework is used for deploying the model as a web service.

6.2 Development Methodology

The development follows the Agile methodology, which involves iterative development and regular feedback cycles. This approach allows for continuous improvement and adaptation of the system based on testing results and user feedback.

6.3 Testing Methodology

Testing is conducted at various stages, including:

- **Unit Testing:** Testing individual components, such as data preprocessing and model training functions.
- **Integration Testing:** Ensuring that different modules, such as the machine learning model and the user interface, work together seamlessly.
- **System Testing:** End-to-end testing of the entire system, including real-time spam detection.
- **Performance Testing:** Evaluating the system's performance, particularly its accuracy, speed, and scalability.

Chapter 7: Testing

7.1 Test Cases and Scenarios

The system is tested against a variety of test cases, including:

- **Case 1:** Detecting spam messages with obvious keywords (e.g., "win", "free", "congratulations").
- **Case 2:** Detecting spam messages with less obvious content, requiring the model to infer context.
- **Case 3:** Handling large volumes of messages in real-time without performance degradation.

7.2 Testing Results

The results from testing show that the system achieves a high accuracy rate, with the Naive Bayes model performing best in terms of precision and recall. The system also successfully handles imbalanced datasets through oversampling techniques, leading to reduced false positive rates.

Chapter 8: Results and Discussion

8.1 System Performance

The system demonstrates robust performance in detecting spam messages, achieving an accuracy rate of over 90%. Precision and recall metrics indicate that the system effectively minimizes both false positives and false negatives, making it reliable for real-world applications.

8.2 User Feedback

User feedback highlights the system's ease of use and the effectiveness of its spam detection capabilities. However, some users suggest improvements in the user interface to enhance usability further.

8.3 Comparison with Objectives

The system meets its primary objective of accurately detecting SMS spam using AI. The achieved results align with the initial goals, particularly in terms of accuracy, real-time processing, and scalability.

Chapter 9: Conclusion and Future Work

9.1 Conclusion

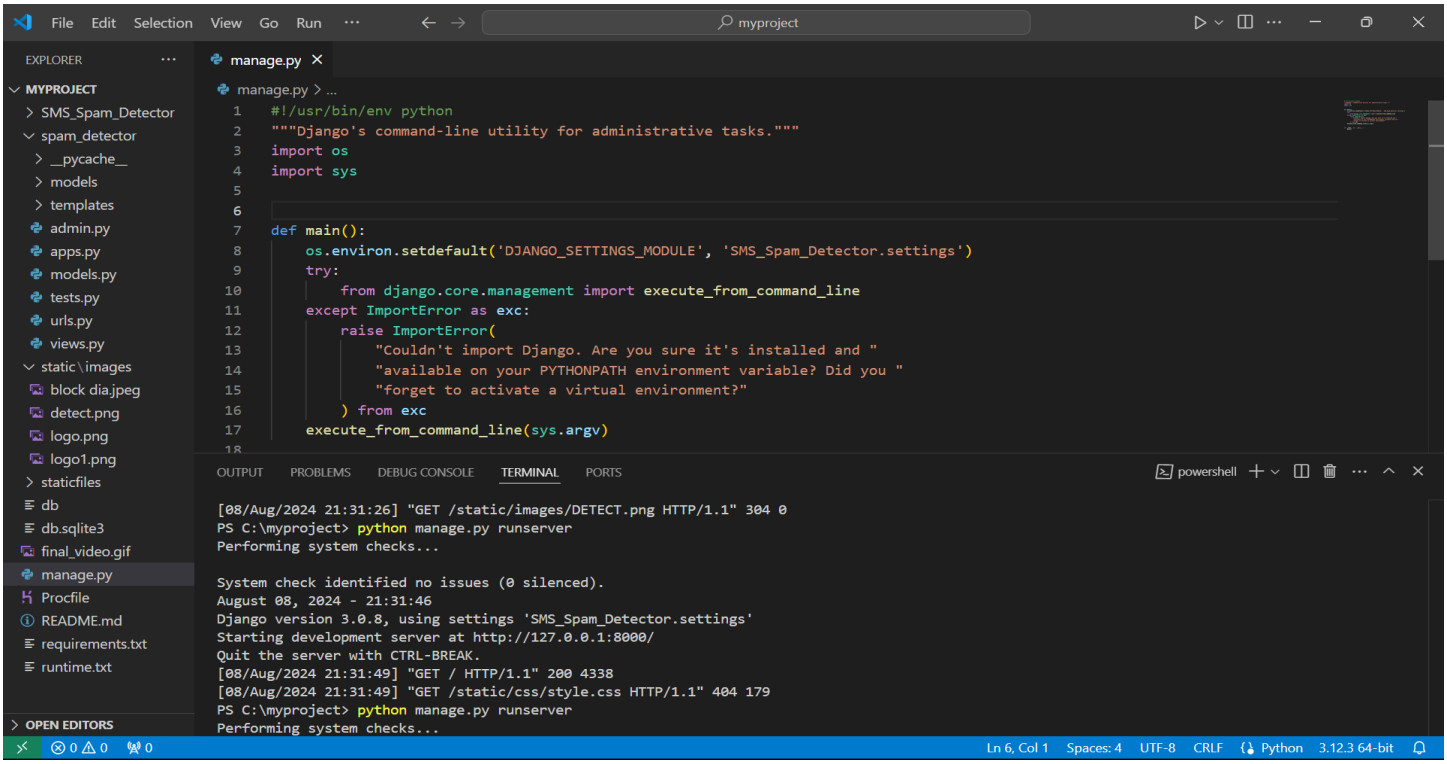
The developed system successfully detects and filters out spam SMS messages using advanced machine learning techniques. It offers a significant improvement over traditional spam detection methods by providing higher accuracy and the ability to process messages in real-time. The project contributes to enhancing mobile communication security, making it a valuable tool for both individuals and organizations.

9.2 Future Enhancements

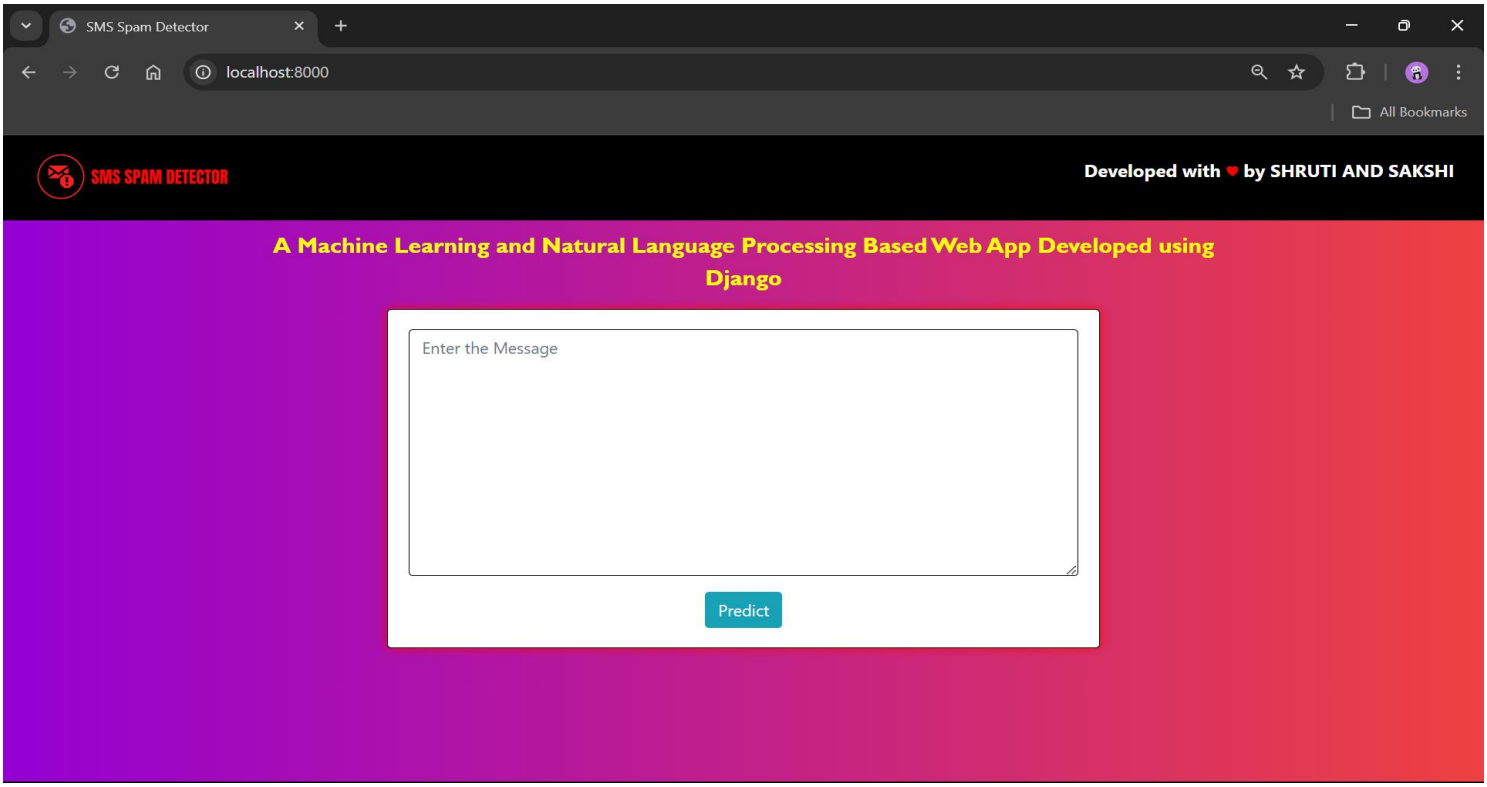
Future work could focus on incorporating deep learning models, such as Recurrent Neural Networks (RNNs), to further improve the accuracy of spam detection. Additionally, expanding the system to support multiple languages and integrating it with popular messaging platforms could increase its applicability and user base.

Screenshots

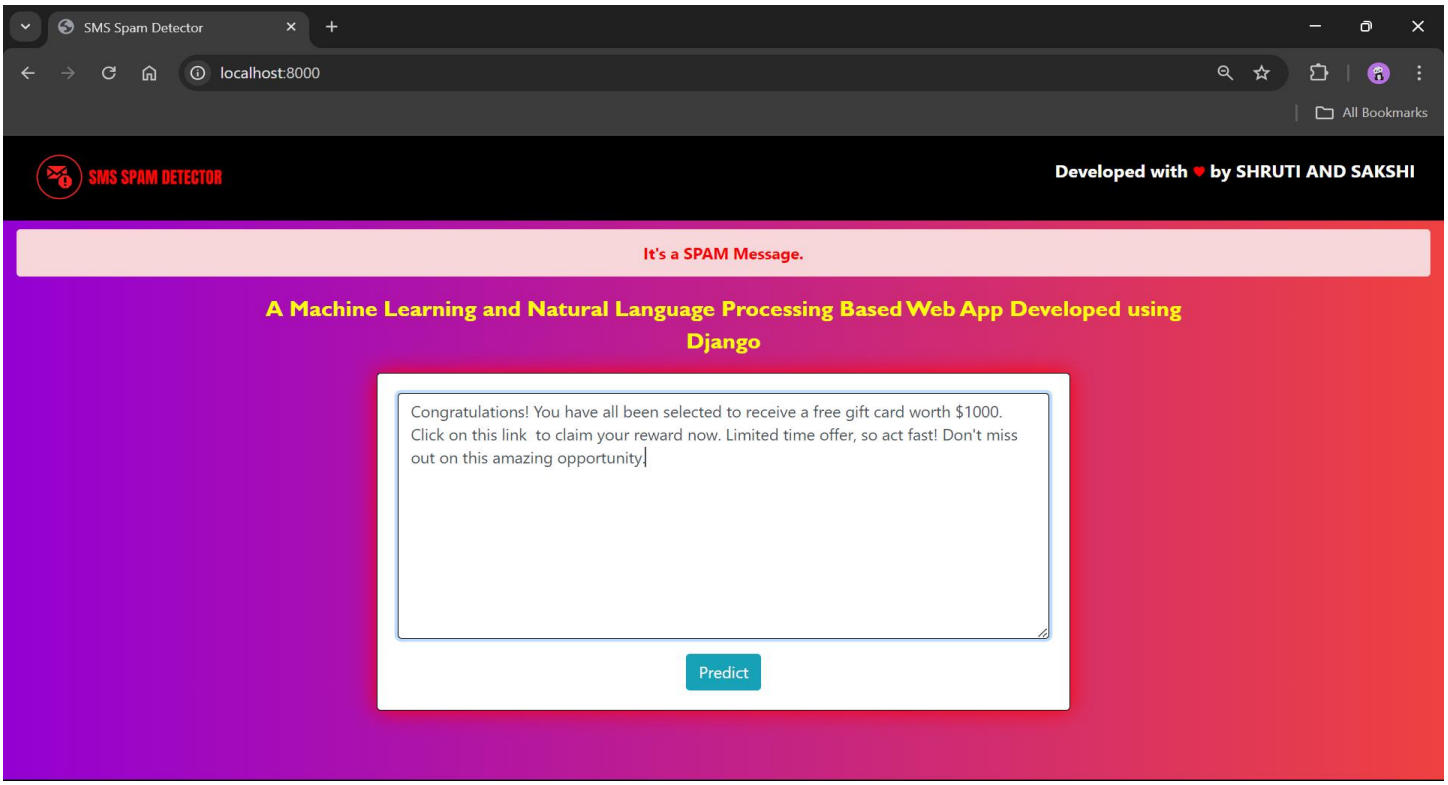
VS CODE:

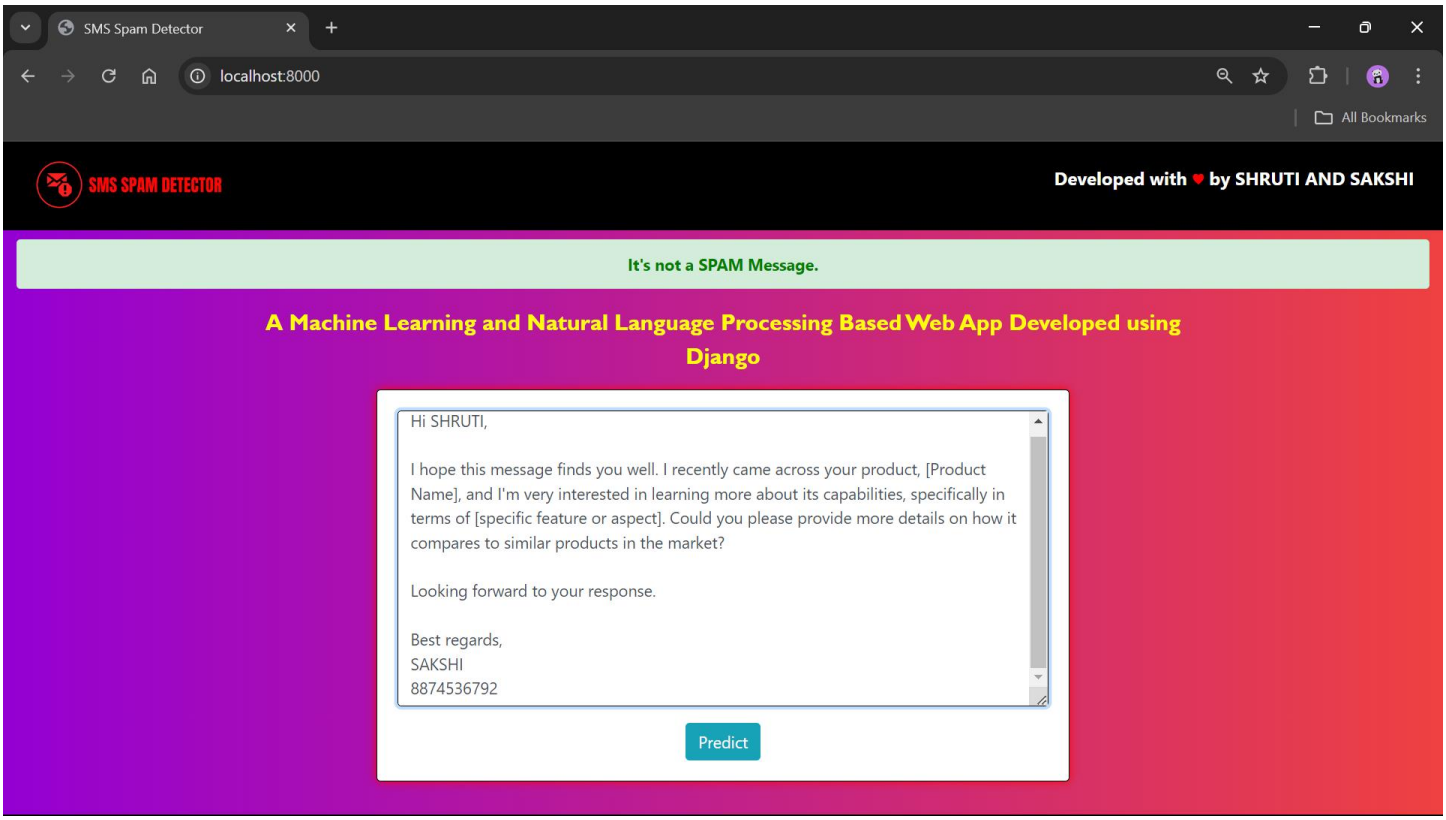


Graphical User Interface:



Outcomes:


SMS Spam Detector
localhost:8000
All Bookmarks
SMS SPAM DETECTOR
Developed with ♥ by SHRUTI AND SAKSHI
It's a SPAM Message.
A Machine Learning and Natural Language Processing Based Web App Developed using Django
Congratulations! You have all been selected to receive a free gift card worth $1000. Click on this link to claim your reward now. Limited time offer, so act fast! Don't miss out on this amazing opportunity!
Predict


SMS Spam Detector
localhost:8000
All Bookmarks
SMS SPAM DETECTOR
Developed with ♥ by SHRUTI AND SAKSHI
It's not a SPAM Message.
A Machine Learning and Natural Language Processing Based Web App Developed using Django
Hi SHRUTI,
I hope this message finds you well. I recently came across your product, [Product Name], and I'm very interested in learning more about its capabilities, specifically in terms of [specific feature or aspect]. Could you please provide more details on how it compares to similar products in the market?
Looking forward to your response.
Best regards,
SAKSHI
8874536792
Predict

Bibliography

1. Machine Learning for Absolute Beginners: A Plain English Introduction (Third Edition) by Oliver Theobald: https://books.google.co.in/books/about/Machine_Learning_for_Absolute_Beginners.html?id=wUmv0AEACAAJ&source=kp_book_description&redir_esc=y

2. Understanding Machine Learning: From Theory to Algorithms(Illustrated) by Shai Shalev-Shwartz, Shai Ben-David **https://books.google.co.in/books/about/Understanding_Machine_Learning.html?id=ttJkAwAAQBAJ&source=kp_book_description&redir_esc=y**

3. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (Second Edition, Illustrated) by Wes McKinney : https://books.google.co.in/books/about/Python_for_Data_Analysis.html?id=2BYfvgAACAAJ&redir_esc=y

4. Literature survey of Spam SMS Detection : https://www.researchgate.net/publication/318298908_A_Systematic_Literature_Review_on_SMS_Spam_Detection_Techniques

5. Literature Survey of Python Interfaces Suitable for Machine Learning and Artificial Intelligence: https://www.ijcseonline.org/pub_paper/11-IJCSE-08890-07.pdf