Email Spam Classification

Dr M Seshu Bhavani, K.Soumya, D.Sravani, A.Sreeja, Y. Vyshnavi

[1]Associate Proffesor, Department of CSE(AI&ML), Bhoj Reddy Engineering College for Women, India

[2,3]B.Tech Students, Department of CSE(AI&ML), Bhoj Reddy Engineering College for Women, India

ABSTRACT

Email spam continues to be a significant challenge in digital communication, often leading to productivity loss, data breaches, and security threats. Traditional rule-based spam filters are static and ineffective against constantly evolving spam techniques. This project presents a machine learning-based email spam classification system that leverages natural language processing (NLP) for efficient and accurate detection. Using TF-IDF vectorizationthe system extracts meaningful features from email text. Several supervised learning algorithms, including Naïve Bayes, Logistic Regression, and Support Vector Machine (SVM), are trained and evaluated to identify spam with high precision. The trained model is integrated into a lightweight web interface using Streamlit, allowing users to input or upload email content and receive instant classification results. The system demonstrates high accuracy and adaptability, offering a scalable and real-time solution for modern spam detection challenges.

1. INTRODUCTION

Email has become one of the most widely used methods of communication in both personal and professional domains. However, the increasing volume of unsolicited and potentially harmful messages, commonly referred to as "spam," poses serious challenges to users and service providers alike. Spam emails not only clutter inboxes and waste time but also serve as a vehicle for phishing attacks, malware distribution, and financial scams.

Traditional spam filters rely heavily on rule-based mechanisms, blacklists, and keyword matching techniques. While these methods offer some level of protection, they are often static and fail to adapt to the constantly evolving strategies used by spammers. Moreover, they tend to suffer from high false positive and false negative rates, which reduce their reliability and effectiveness.

Existing System

Traditional spam detection systems primarily rely on **rule-based filtering**, **keyword matching**, and **blacklist mechanisms**. These systems function by scanning the content of incoming emails for predefined patterns, phrases, or known malicious addresses. If a match is found, the email is flagged as spam and either moved to a junk folder or blocked entirely.

One commonly used technique in these systems is the use of **blacklists**, which store the email addresses or domains of known spammers. Another technique involves **heuristic-based analysis**, where specific patterns in subject lines, sender names, or attachments are evaluated against a static set of rules.

Proposed System

To overcome the limitations of traditional spam filtering techniques, the proposed system adopts a **machine learning-based approach** combined with **natural language processing (NLP)** for efficient and intelligent spam classification. This system is designed to automatically learn from data, adapt to evolving spam patterns, and provide accurate real-time classification through a user-friendly web interface.

2. REQUIREMENT ANALYSIS

Functional Requirements

In this project, we have designed the following modules:

User Module

Users can upload an email through the webpage to determine whether it is spam or ham.

Model

- PreProcesssing
- Feature Extraction
- Classification

Non-Functional Requirements

Usability:

- The user interface should be intuitive and easy to navigate.
- The system should provide clear instructions and feedback to users.

Reliability:

- The system should be reliable and handle expected loads without crashing.
- The system should ensure data consistency and integrity.

Maintainability:

- The system should be simple to maintain and update.

Performance:

- The system should deliver real-time or near real-time responses (within 1–2 seconds) for email classification.
- Model prediction and output display must occur with minimal latency.

Sofware Requirements

Operating System:

- reasoning about the structure of the system.

Windows 7, 8, 10, or more

Text Editor :

Sublime

Framework :

Streamlit

Libraries:

Scikit-learn – For machine learning algorithms and feature selection (TF-IDF)

NLTK – For natural language processing (text tokenization,

stopword removal, etc.) Language: Python

Pickle – For saving and loading the trained model

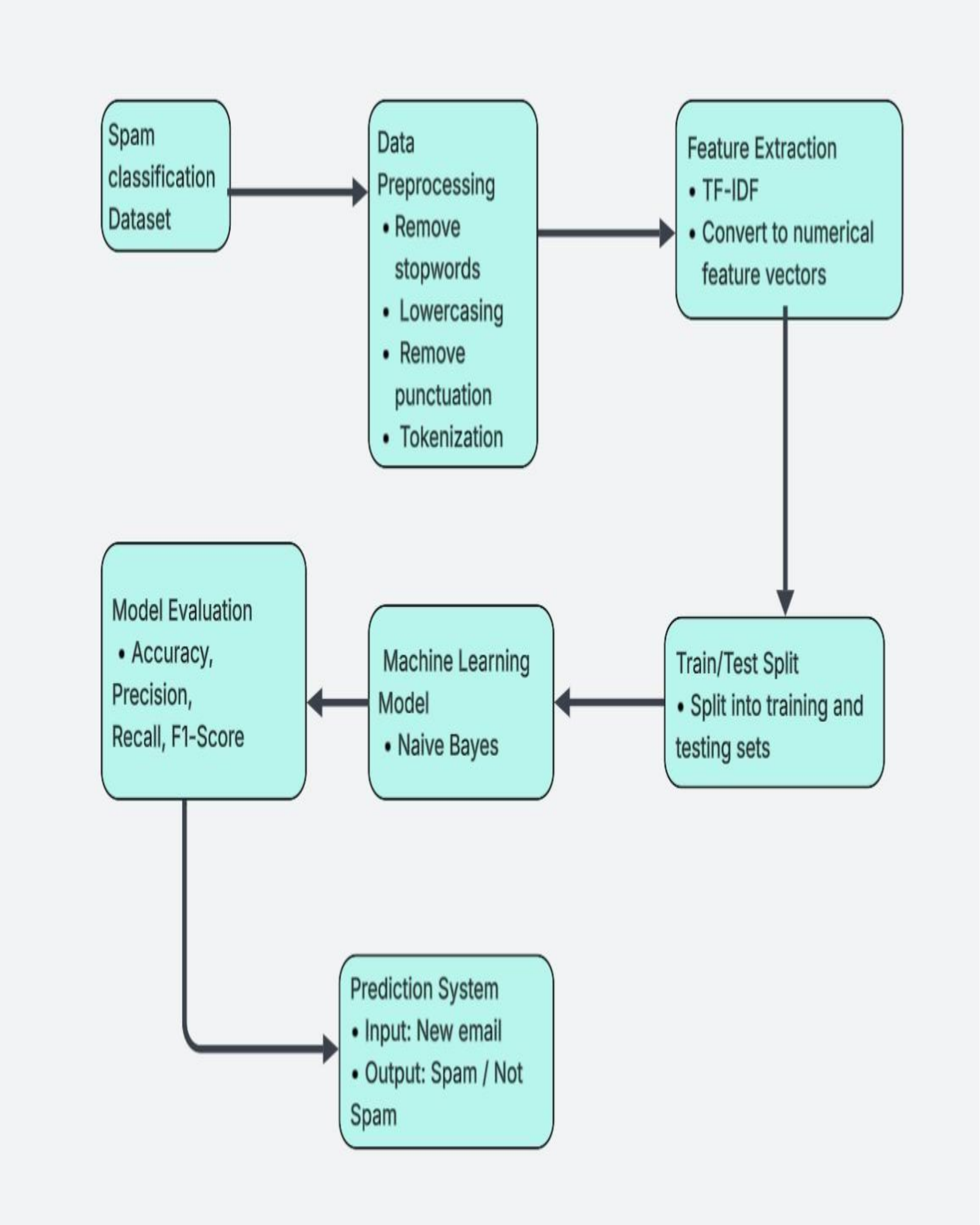
Hardware Requirements

Processor:	Intel i5 or higher
RAM:	8GB+ for smooth
execution Storage:	256GB SSD
(recommended)	
Internet:	Required for model
training & deployment Display:	Full HD for better
visualization	

3. DESIGN

- Design represents the number of components we are using as a part of the project and the flow of request processing i.e., what components in processing the request and in which order.
- An architecture description is a formal description and representation of a system organized in a way that supports

System Architecture



System Architecture of Email spam classification

3.1 Software Process Model

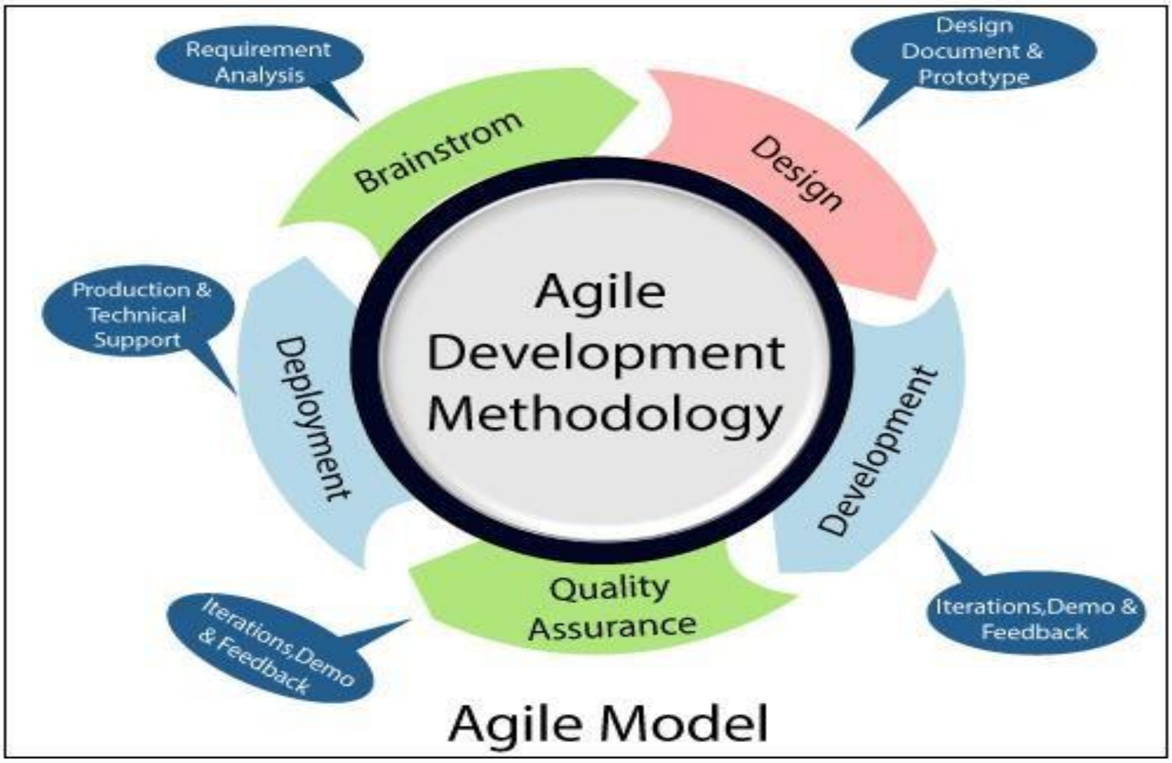

Requirement Analysis
Brainstrom
Design
Design Document & Prototype
Agile Development Methodology
Production & Technical Support
Deployment
Development
Quality Assurance
Iterations,Demo & Feedback
Iterations,Demo & Feedback
Agile Model

Fig 2 Software process model

4. IMPLEMENTATION

Technologies

The proposed system is implemented using Python-based tools and libraries, enabling efficient email preprocessing, machine learning model training, and deployment through a web-based interface. Below are the key steps in the implementation along with the technologies used at each stage

1. **Environment Setup**

   Technology Used: Python , Jupyter Notebook, Streamlit

   Set up a Python environment with necessary libraries for machine learning and NLP tasks.

2. **Data Collection**

   Technology Used: CSV files

   A labeled dataset of spam and ham emails is loaded for training and testing the model.

3. **Text Preprocessing**

   Technology Used: NLTK (Natural Language Toolkit) Operations performed:

   Lowercasing Stopword removal Tokenization Punctuation removal

4. **Feature Extraction**

   Technology Used: Scikit-learn

   Uses TF-IDF Vectorizer to convert email text into numerical feature vectors. Applies to select the most relevant features.

   Random Forest

   Evaluation Metrics: Accuracy, Precision, Recall, F1-score

5. **Model Saving**

   Technology Used: Pickle

The trained model is serialized and saved as a .pkl file for deployment.

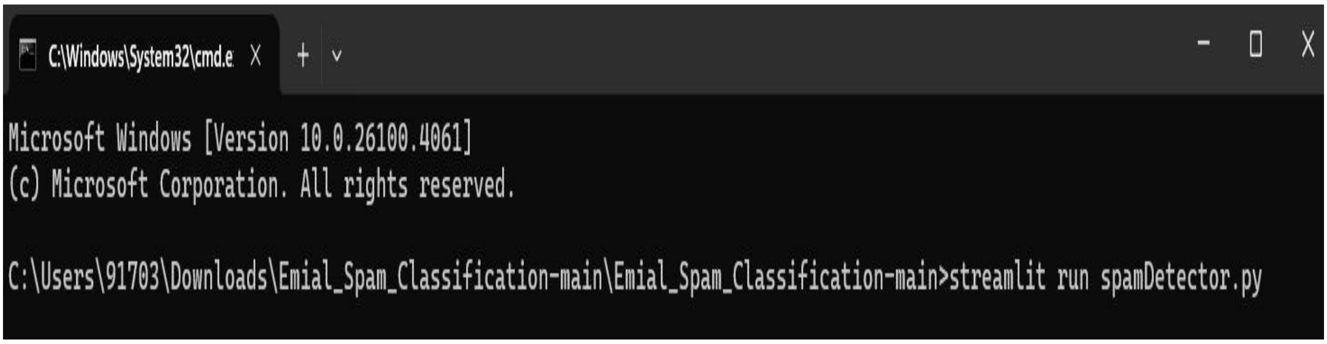
6. **Web Deployment**

Technology Used: Streamlit

A lightweight web interface is built where users can input or upload email content. Displays real-time classification result as Spam or Ham.

This modular implementation ensures high accuracy, scalability, and ease of use for end users while following best practices in machine learning workflows.

5. SCREENSHOTS



```
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91703\Downloads\Emial_Spam_Classification-main\Emial_Spam_Classification-main>streamlit run spamDetector.py
```

Fig 1 Activate Project

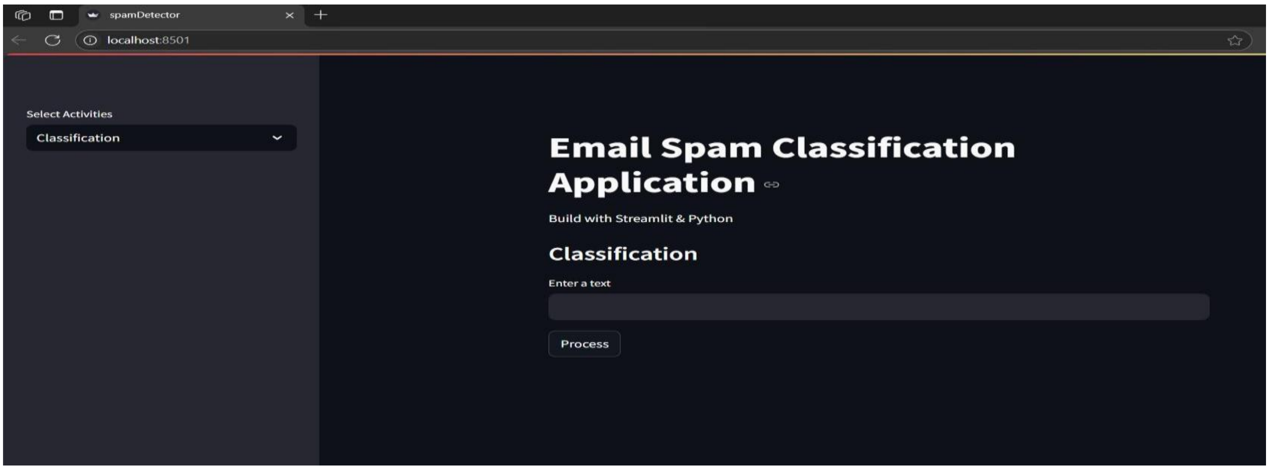
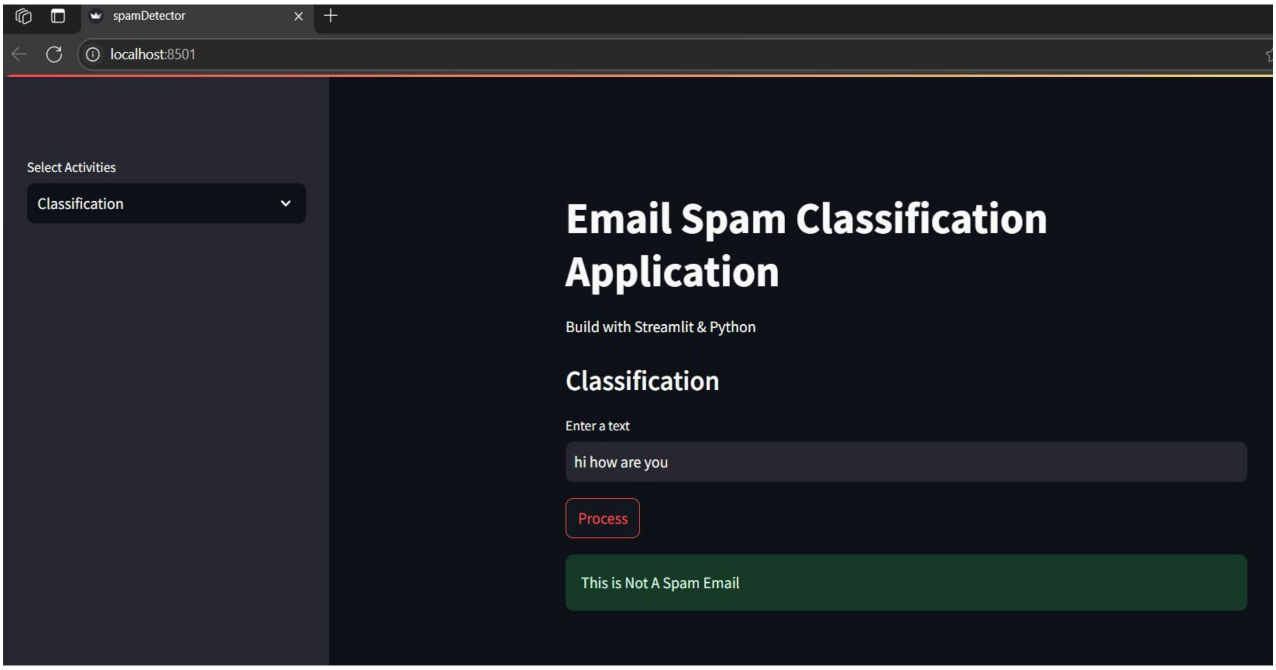


Fig 2 Main Page



Email Spam Classification Application

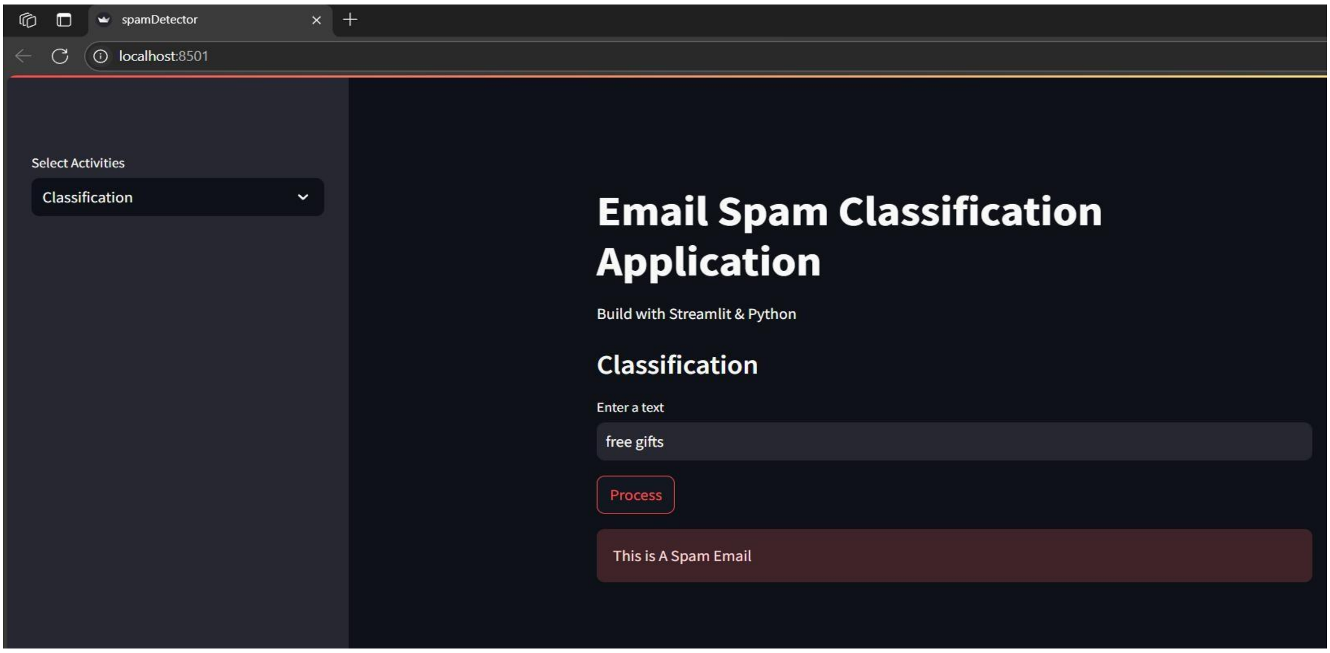
Build with Streamlit & Python

Classification

Enter a text: hi how are you

This is Not A Spam Email

Fig   3 Output-1



Email Spam Classification Application

Build with Streamlit & Python

Classification

Enter a text: free gifts

This is A Spam Email

Fig 4 Output-2

CONCLUSION

In this project, an intelligent email spam classification system was developed using machine learning and natural language processing techniques. The system effectively processes raw email content, converts it into numerical vectors using TF-IDF, and classifies it as either spam or ham using trained machine learning models such as Naïve Bayes and Logistic Regression. The integration of the trained model into a user-friendly Streamlit web interface allows for real-time predictions and seamless interaction.

This approach overcomes the limitations of traditional rule-based filters by enabling adaptive learning, scalability, and higher classification accuracy. The experimental results confirm that the system performs reliably and efficiently in detecting spam emails, offering a practical solution for individual and organizational email security.

REFERENCES

[1] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A Bayesian approach to filtering junk e-mail,” Proc. AAAI Workshop on Learning for Text Categorization, 1998.

[2] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, “SMS Spam Collection Dataset,” 2011. Available: https://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

[3] Scikit-learn Documentation. Available: https://scikit-learn.org

[4] NLTK Documentation. Available: https://www.nltk.org

[5] Streamlit Documentation. Available: https://streamlit.io