DCSA, PANJAB UNIVERSITY CHANDIGARH

Project Title:

"Machine Learning-Based Email Spam Classification"

Submitted By: Sakshi Sharma

Class: MSc Computer Science (Hons.) – 2^{nd} sem

Roll no. 18

Dept.: DCSA

Submitted To: Mrs. Kamakshi Pundir
& Mr. Rishabh Batra

CONTENTS

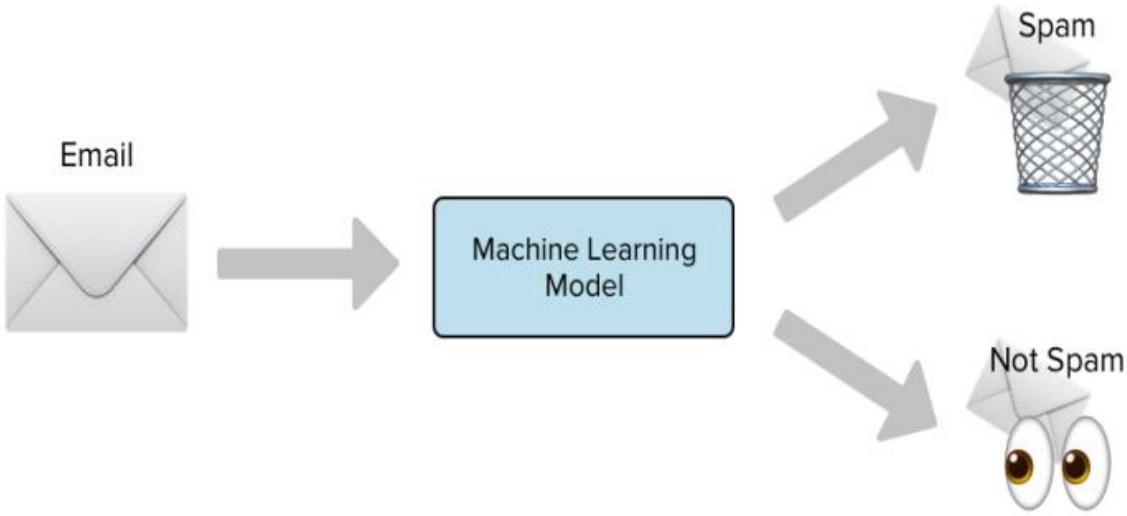
- Abstract
- Introduction
- Literature Survey
- Problem Statement
- Objectives
- Scope
- Applications
- Methodology
- Workflow
- Steps to be followed
- Comparison of models
- GUI Overview
- Screenshots
- Future Scope
- Conclusion
- References

Abstract:

With the exponential increase in email usage, distinguishing between spam and legitimate messages has become a critical task. This project presents a machine learning-based approach to spam email detection using ML techniques. The dataset used comprises labelled emails categorized as "spam" and "ham" (non-spam). The data underwent several preprocessing steps, including tokenization, stop-word removal, and feature extraction through Count Vectorizer and TF-IDF techniques. Several classification algorithms including Naive Bayes, Logistic Regression were trained and evaluated. Among these, Logistic Regression achieved the highest accuracy of approximately 93%. This study demonstrates the effectiveness of machine learning techniques in building reliable and efficient spam detection systems contributing to improved email security and user experience.

Introduction:

Email or electronic mail spam refers to the using of email to send unsolicited emails or advertising emails to a group of recipients. Unsolicited emails mean the recipient has not granted permission for receiving those emails. Spam is a waste of storage, time and message speed. Machine learning approach will be used for spam detection. Text assessment of contents of mails is an extensively used method to the spams. Evaluating whether the given e-mail is spam involves looking at the e-mail's content, sender reputation, IP address, e-mail headers, and other elements. In recent years, most e-mail messages have fallen into the "Ham" and "Spam" categories. Spam messages are the garbage, unsolicited mass, or commercial communications in an inbox, whereas Ham messages are the intended or safe legitimate messages. This separation of e-mail communications into "Ham" and "Spam" categories aids in automatically deleting spam messages. Spam e-mail detection can be cost-effective compared to manual spam filtering, customizable to meet the specific requirements of people and organizations. It can increase productivity by letting users concentrate on accurate communications. It can also improve security by protecting users from harmful content. Nevertheless, there are some disadvantages, such as false positives that might block crucial e-mails, resource-intensive deployment and maintenance, and reliance on technology that may be subject to interruptions or failures. Moreover, spam e-mail detection systems must be created and invested in continual research and development to keep up with spammers' new strategies.



Literature Survey

Spam detection has been a widely researched problem within the domain of natural language processing (NLP) and machine learning (ML) due to the growing volume of unsolicited and malicious email traffic. Numerous approaches have been proposed over the years to improve the accuracy and efficiency of email classification systems.

M. Sahami et al. (1998) were among the pioneers in applying machine learning techniques for spam filtering, demonstrating the effectiveness of Naive Bayes classifiers for text categorization tasks. Their approach laid the foundation for many future spam filters by utilizing term frequency and word probabilities.

Subsequent research has extended to more advanced models such as Support Vector Machines (SVMs) and Random Forests. Drucker et al. (1999) found that SVMs outperformed traditional rule-based filters and naive Bayes classifiers in email classification, particularly when dealing with high-dimensional text data.

In recent years, deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been explored for spam detection due to their ability to learn semantic representations from raw text. For instance, Zhang et al. (2015) demonstrated the efficacy of character-level CNNs for text classification tasks, including spam detection.

Moreover, Natural Language Processing techniques like TF-IDF, word embeddings (e.g., Word2Vec, GloVe), and transformer-based models (e.g., BERT) have been leveraged to enhance feature representation in spam filtering systems. These advancements have led to significant improvements in precision, recall, and overall classification performance.

Despite these advancements, spam detection systems continue to face challenges such as evolving spam tactics, concept drift, and handling imbalanced datasets. As a result, ongoing research focuses on improving model adaptability, interpretability, and robustness against adversarial attacks.

Problem Statement:

With the exponential growth of email communication, users are increasingly exposed to unwanted and potentially harmful messages, commonly referred to as spam. These messages not only clutter inboxes but also pose serious security risks, including phishing attacks and malware distribution. Traditional rule-based spam filters often struggle to adapt to the evolving tactics used by spammers. Therefore, there is a pressing need for intelligent, automated systems that can effectively distinguish between legitimate (ham) and spam emails. This project aims to design and implement a machine learning-based spam detection system that leverages natural language processing techniques and classification algorithms to accurately identify and filter spam emails, thereby enhancing email security and user productivity.

Objectives:

The objectives of this project are:

- **To collect and preprocess a labelled dataset** of spam and ham emails suitable for machine learning applications.

- **To build an accurate classification model** that distinguishes spam from non-spam (ham) emails using machine learning techniques.

- **To preprocess and extract features** from email text using Natural Language Processing (NLP) methods for effective model training.

- **To evaluate and compare algorithms** based on performance metrics such as accuracy, precision, recall, and F1-score.

- **To develop an efficient and reliable system** capable of real-time spam detection with minimal false positives and false negatives.

Scope:

This project focuses on developing a machine learning-based system to detect and classify email messages as either *spam* or *ham* (non-spam). The scope of this project includes:

- Utilizing publicly available datasets such as the **UCI SMS Spam Collection** or **SpamAssassin** for training and testing.
- Applying standard **Natural Language Processing (NLP)** techniques for email text preprocessing and feature extraction.
- Implementing and comparing various **supervised machine learning algorithms**, including Naive Bayes, Logistic Regression, Support Vector Machine (SVM), and Random Forest.
- Evaluating the models using performance metrics such as **accuracy**, **precision**, **recall**, and **F1-score**.
- Building a prototype capable of predicting whether a new email is spam or not, based on learned patterns from the training data.

This project does **not** focus on real-time email filtering systems or integrating the model into a live email client but serves as a foundational step towards such applications.

Applications:

□ Email Service Providers:

- Major platforms like Gmail, Outlook, and Yahoo Mail integrate spam detection systems to automatically filter out unwanted messages from user inboxes.

□ Enterprise Email Security:

- Organizations use spam filters to protect employees from phishing attacks, scams, and malware distributed through email.

□ Cloud-Based Security Solutions:

- Spam detection models are deployed as part of cloud-based email security services to provide scalable and real-time protection for businesses.

□ Mobile Email Clients:

- Mobile apps utilize spam detection models to improve user experience by reducing inbox clutter.

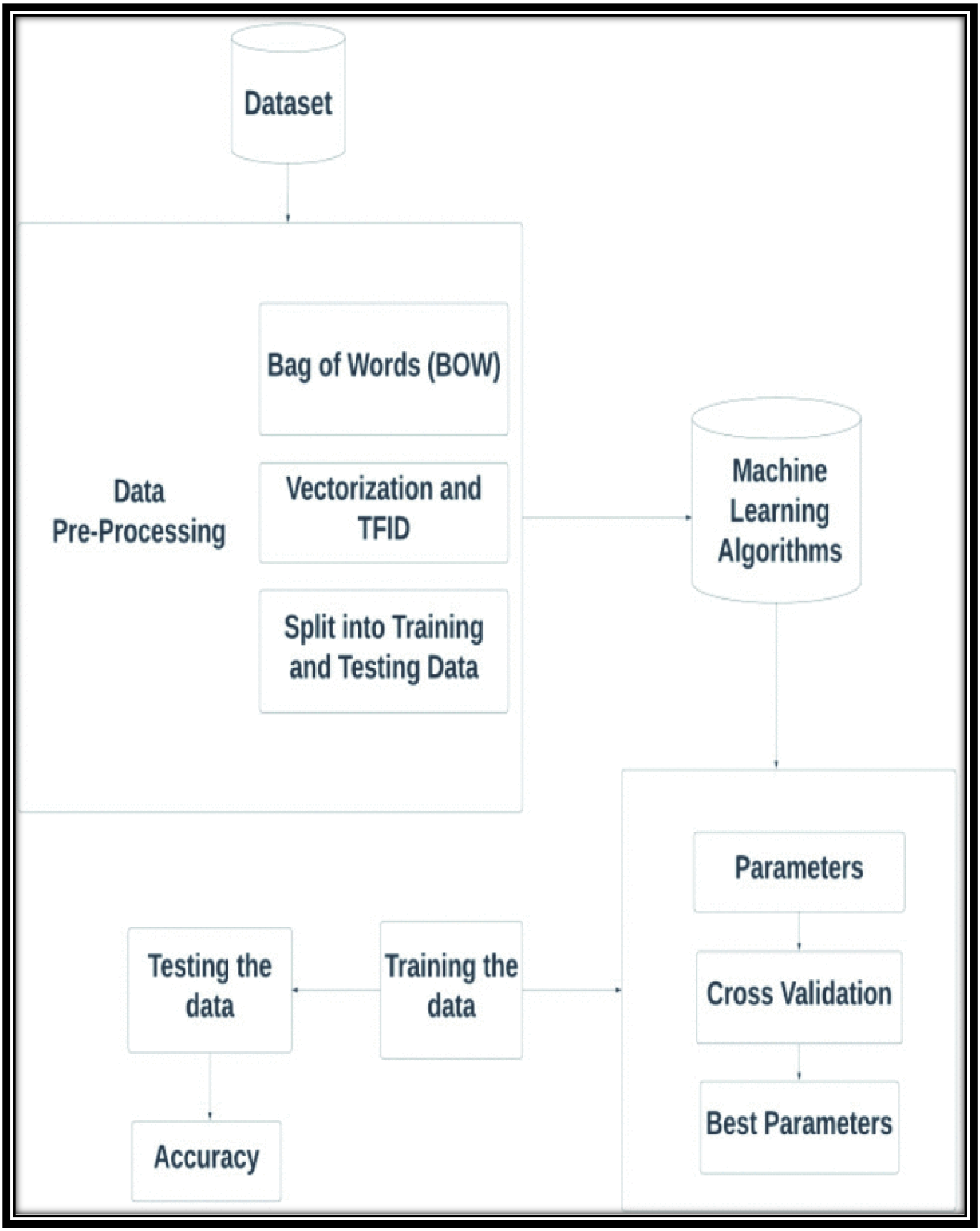
□ Customer Support Systems:

- Automated systems can detect and segregate spam or irrelevant messages sent through support channels, improving response efficiency.

□ Marketing Platforms:

- Email marketing tools can identify and flag messages that may be classified as spam to help businesses improve email deliverability.

Methodology:

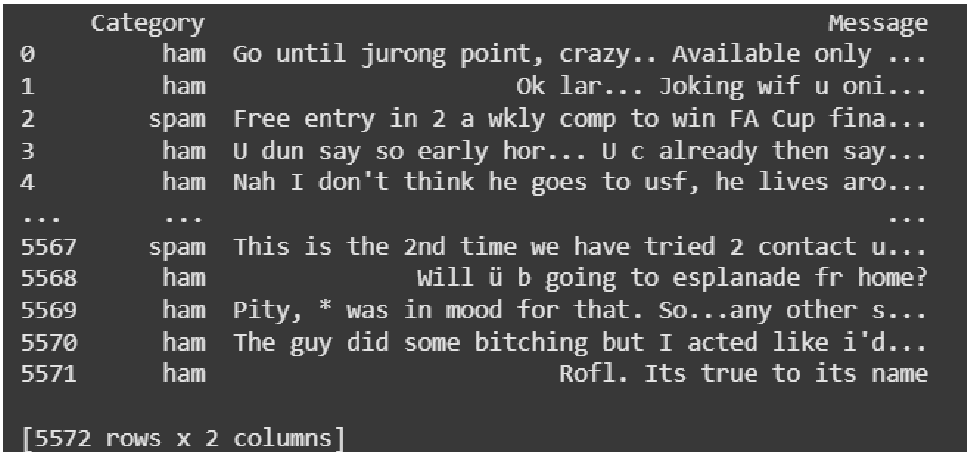


Workflow:

Dataset Collection:

The first step is gathering the dataset containing text messages labelled as spam or not spam (ham). Data named "mail_data.csv" is imported from Kaggle, which is available publicly. The file contains 5572 data rows and has two columns: Category and Message.

Dataset taken from kaggle, link: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset/code

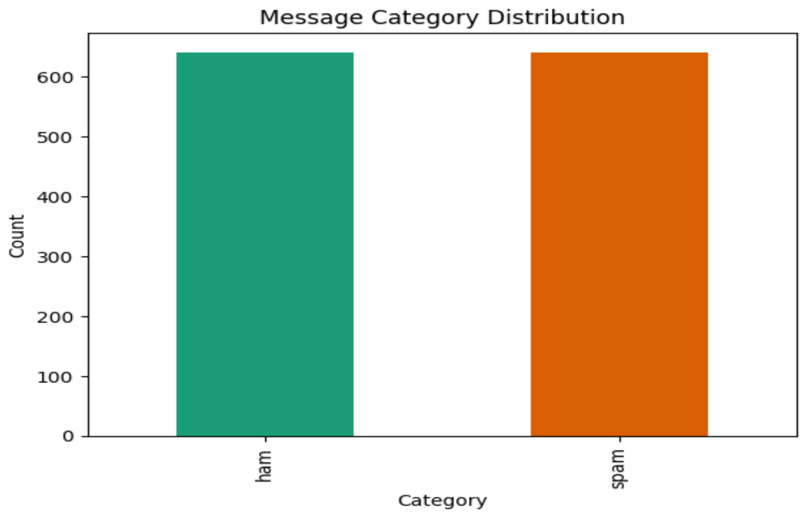


```
     Category                                            Message
0         ham  Go until jurong point, crazy.. Available only ...
1         ham                      Ok lar... Joking wif u oni...
2        spam  Free entry in 2 a wkly comp to win FA Cup fina...
3         ham  U dun say so early hor... U c already then say...
4         ham  Nah I don't think he goes to usf, he lives aro...
...       ...                                                ...
5567     spam  This is the 2nd time we have tried 2 contact u...
5568      ham               Will ü b going to esplanade fr home?
5569      ham  Pity, * was in mood for that. So...any other s...
5570      ham  The guy did some bitching but I acted like i'd...
5571      ham                         Rofl. Its true to its name

[5572 rows x 2 columns]
```

Data Pre-Processing:

- **Vectorization and TF-IDF:** Term Frequency-Inverse Document Frequency (TF-IDF) transformation is applied to reweight the raw word counts based on their importance, reducing the impact of common words.
- **Data Balancing:** Since real-world datasets often have more ham messages than spam, data balancing techniques (such as undersampling, oversampling, or SMOTE) are applied to ensure that both classes are fairly represented during model training. This helps to prevent the model from being biased toward the majority class.



Label Encoding:

We are labelling the categorical data in categories like spam and ham, where the ham entries are labelled as 1, and the spam entries are marked as 0. Then we separate the dataset as text and label.

Splitting the data into training data & test data:

Splitting the data set refers to breaking the dataset into two or more parts for training and testing purposes. The dataset was split into 80% training data and 20% testing data.

Feature Extraction:

Feature Extraction is a method of converting raw data into numerical features which aim to be handled by keeping the information in the original dataset. Feature extraction helps generate better outcomes than machine learning techniques on the raw data.

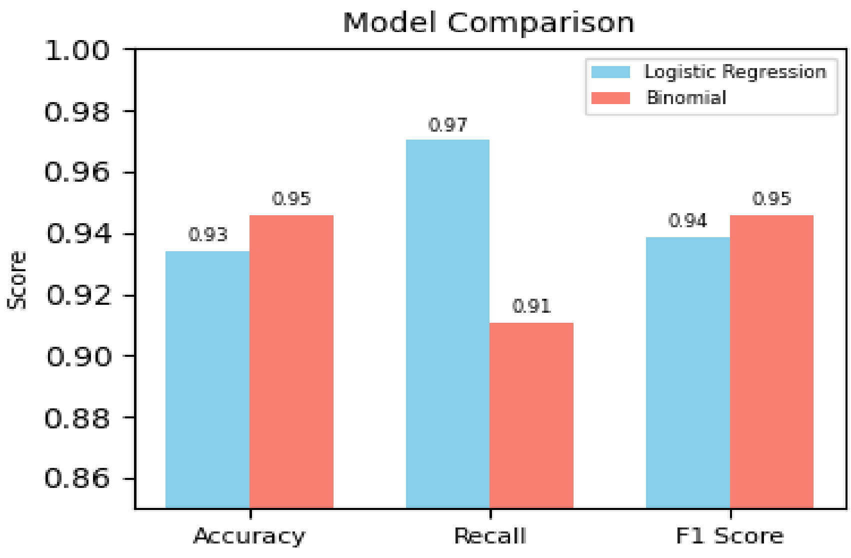
Model Training and Testing:

- The model is trained using the processed training data.
- The trained model is then tested on the testing data to evaluate its predictive performance.

Performance Measure:

The trained model is evaluated using the following metrics:

- **Accuracy:** Measures the overall percentage of correctly classified messages.
- **F1 score:** The harmonic mean of precision and recall, providing a single metric that balances the two.
- **Recall:** Measures the proportion of actual spam messages that are correctly identified.



Steps to be followed:

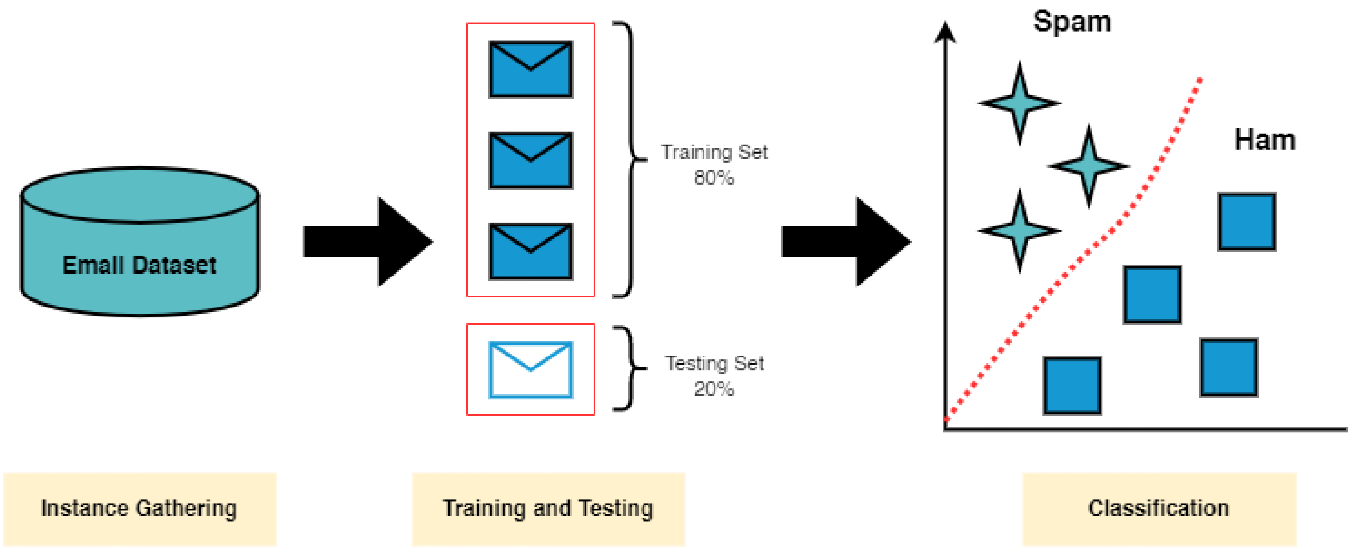
1. **Problem Understanding and Literature Survey**
   Conducted a detailed review of existing approaches to spam detection, including traditional machine learning techniques (e.g., Naive Bayes, Logistic regression) and modern NLP-based models. Identified common datasets, preprocessing methods, and performance benchmarks.

2. **Dataset Collection and Exploration**
   Obtained a labelled dataset (mail_data.csv) containing spam and ham emails. Performed exploratory data analysis (EDA) to understand data distribution and identify common patterns in spam vs. non-spam content.

3. **Data Preprocessing**
   Applied text preprocessing techniques including:
   - Lowercasing and punctuation removal
   - Stop-word removal and tokenization
   - Vectorization using TF-IDF or Count Vectorizer

4. **Model Implementation**
   Implemented baseline machine learning models such as:
   - **Naive Bayes** for its simplicity and efficiency in text classification
   - **Logistic Regression** and **Support Vector Machine (SVM)** for comparative evaluation
     Trained models and obtained initial results for accuracy and other performance metrics.

5. **Performance Evaluation**
   Evaluated models using metrics like accuracy, precision, recall, F1-score, and confusion matrix. Identified areas where performance can be improved (e.g., reducing false positives).



6. **Model Deployment in GUI:**
   The models were serialized using Pickle. These pre-trained models were loaded into the Colab GUI, allowing real-time spam detection based on user input.

7. **User Interaction through GUI:**
   Users select a model, enter email content, and receive instant feedback through a visually enhanced interface.

8. **Result Display and Feedback:**
   Based on the selected model's prediction, the result ("Spam" or "Not Spam") is displayed with clear visual cues.

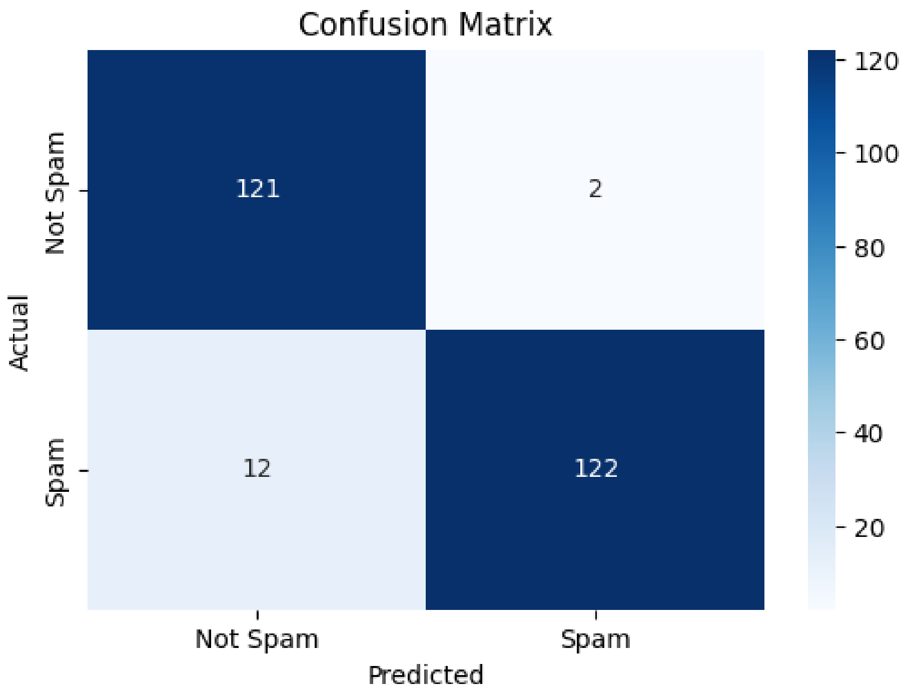
Comparison of models:

Metric	Logistic Regression	Naive Bayes
Accuracy on Test Data	🟢 0.9339	🟢 0.9455
Recall on Test Data	🔵 0.9701	🔵 0.9104
F1 Score on Test Data	🟣 0.9386	🟣 0.9457

Based on the evaluation metrics, both Logistic Regression and Naive Bayes perform well for the spam detection task.

However, there are subtle differences:

- Naive Bayes achieves a slightly higher Accuracy (94.55%) and F1 Score (94.57%), suggesting it provides a better balance between precision and recall.
- Logistic Regression achieves a higher Recall (97.01%), meaning it is more effective at identifying actual spam messages, which is crucial in minimizing false negatives.



□ True Negatives (Not Spam correctly predicted): 121

□ False Positives (Not Spam incorrectly predicted as Spam): 2

□ False Negatives (Spam incorrectly predicted as Not Spam): 12

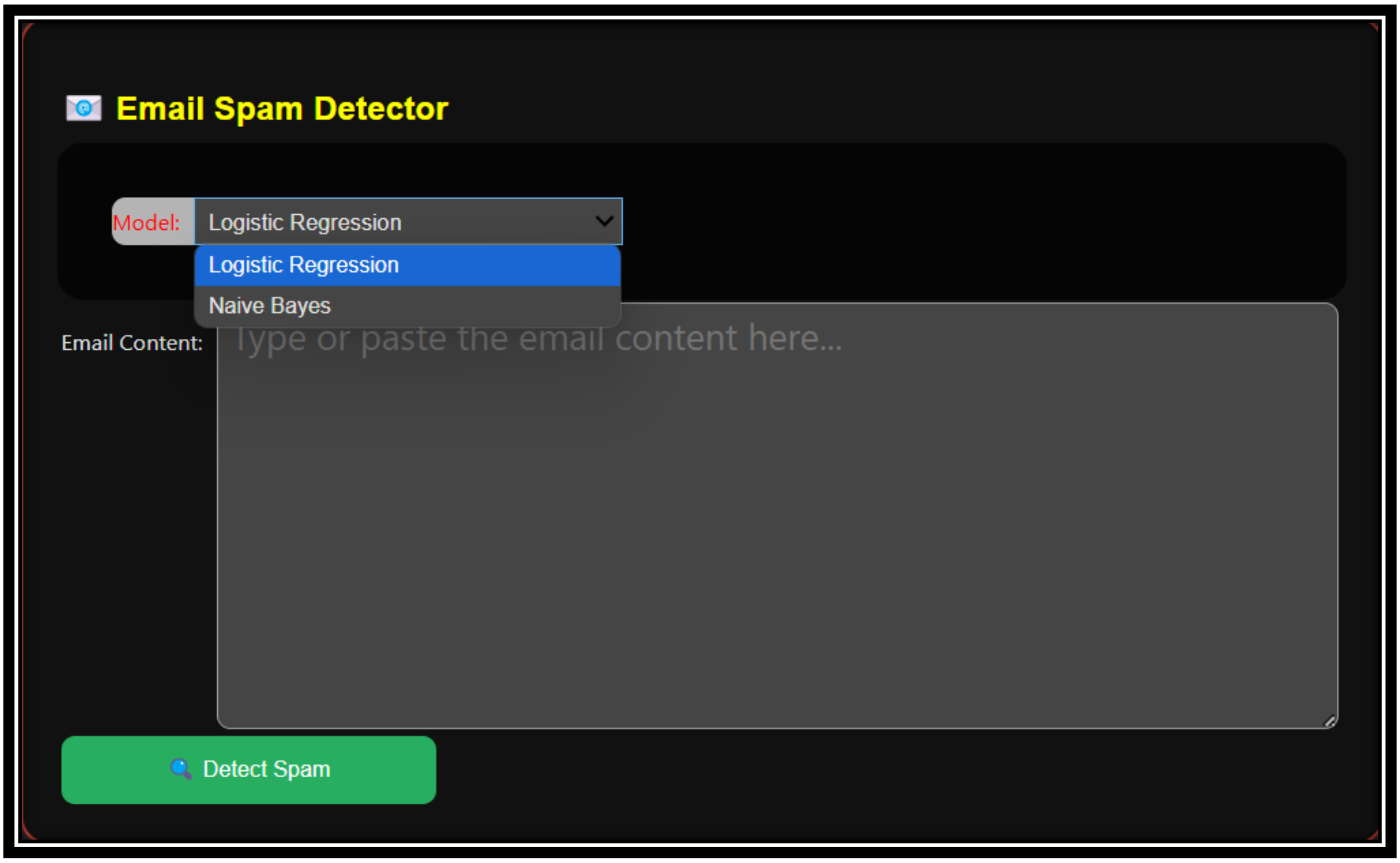
□ True Positives (Spam correctly predicted): 122

GUI Overview:

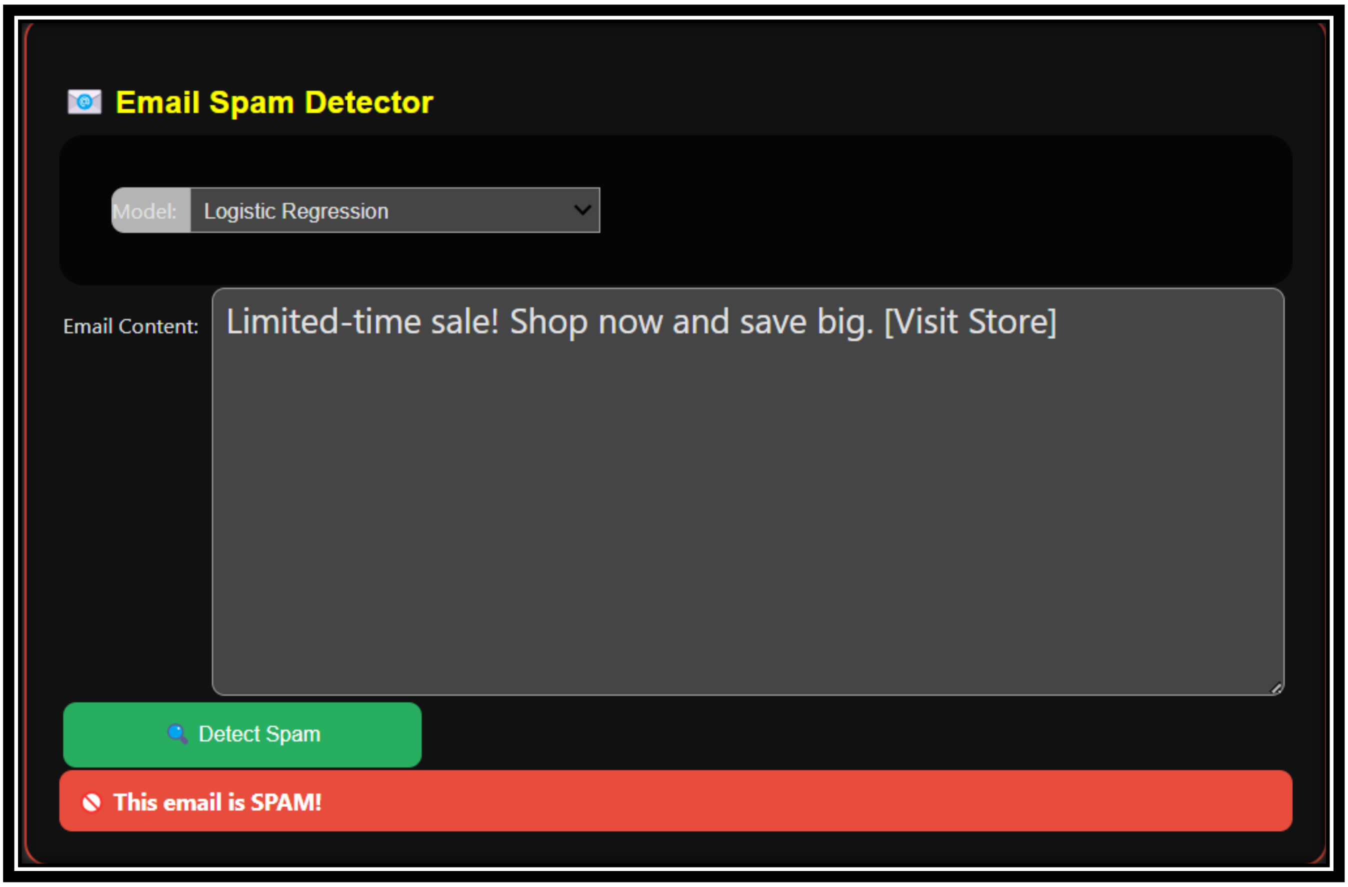
The Graphical User Interface (GUI) for the Email Spam Detection project was developed using Python in Google Colab with the help of the ipywidgets library.
The interface allows users to:

- Select the desired machine learning model (Logistic Regression or Naive Bayes) from a dropdown menu.
- Input or paste email content into a text area.
- Click a "Detect Spam" button to classify the email.

Once the user submits the email content, the system displays a colored output indicating whether the email is Spam (🚫 red box) or Not Spam (✅ green box), providing an intuitive and visually clear response.



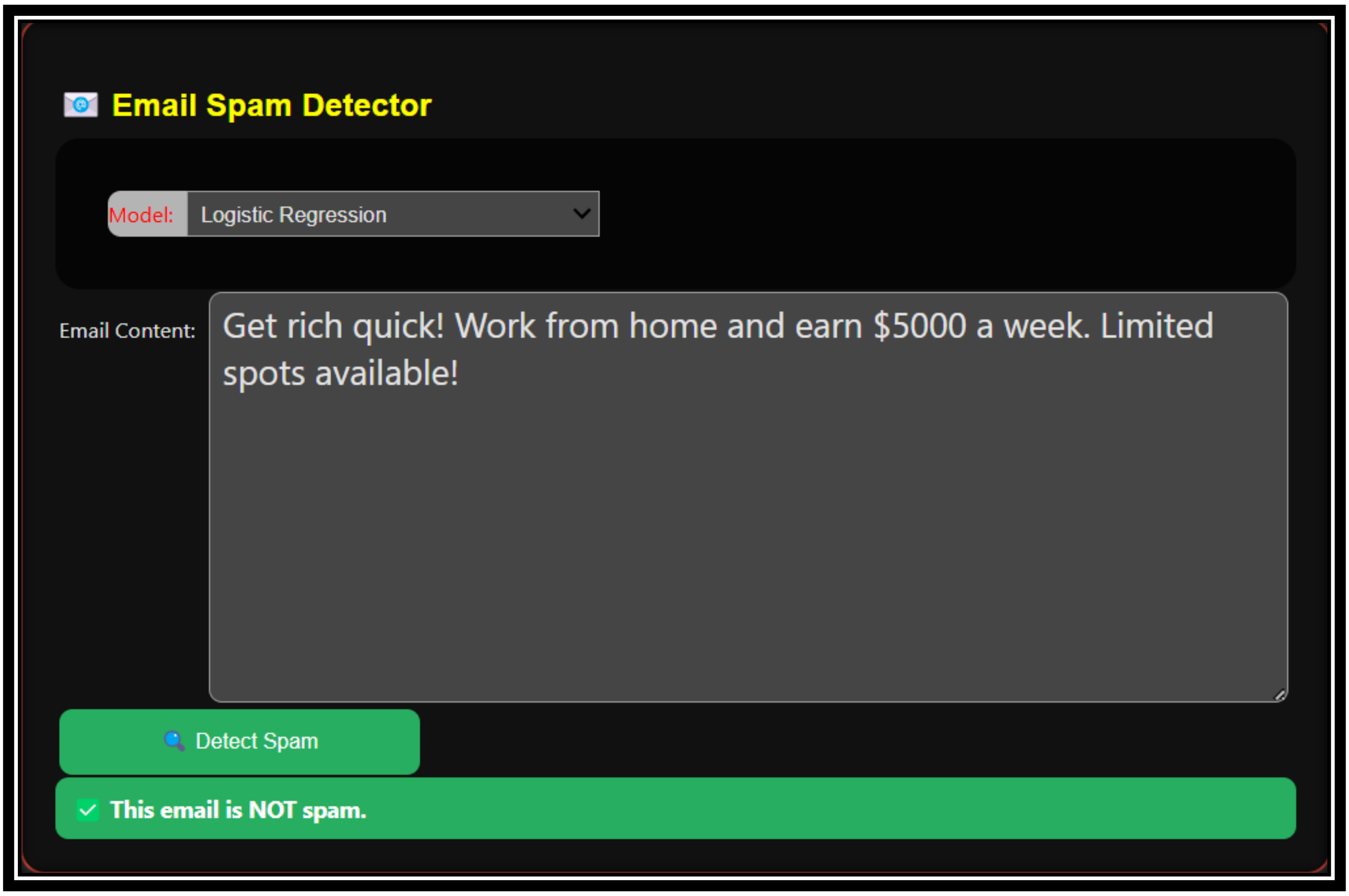
Screenshots:



Model: Logistic Regression

Email Content: Limited-time sale! Shop now and save big. [Visit Store]

This email is SPAM!



Model: Logistic Regression

Email Content: Get rich quick! Work from home and earn $5000 a week. Limited spots available!

✓ This email is NOT spam.

Future Scope:

□ **Advanced Models**:

- Explore **deep learning models** (e.g., RNN, LSTM, BERT).
- Try **ensemble methods** (e.g., Random Forest, XGBoost).

□ **Enhanced Text Preprocessing**:

- Implement **stemming** and **lemmatization** to improve feature extraction.
- Use **word embeddings** (e.g., Word2Vec, GloVe, BERT).

□ **Real-Time Spam Detection**:

- Deploy the model in **real-time** systems (e.g., apps, web services).
- Use **online learning** to update models with new data.

□ **Multilingual Detection**:

- Extend the model for **multilingual** spam detection.

□ **Model Interpretability**:

- Implement **LIME** or **SHAP** for better model transparency.

□ **Scalability**:

- Test the model on **larger, more complex datasets** for improved scalability.

Conclusion:

This project demonstrated the effectiveness of machine learning techniques in detecting spam emails through a systematic approach involving data preprocessing, feature extraction, model training, and evaluation. By using a labelled dataset of spam and ham emails, and applying text processing techniques such as tokenization, stop-word removal, and vectorization (Count Vectorizer and TF-IDF), relevant features were extracted to train classification models. Among the algorithms implemented, Logistic Regression achieved the highest accuracy of approximately 93%, outperforming Naive Bayes in terms of precision and overall classification performance.

The results confirm that machine learning provides a reliable and scalable solution for spam detection, capable of adapting to evolving patterns in email communication. The study also highlights the importance of proper data preprocessing and feature engineering in achieving high model accuracy. Overall, this project contributes to the development of intelligent and automated spam filtering systems that can enhance email security and user productivity.

References:

- Email Spam Detection Using Machine Learning Algorithms by IEEE

  Published in: 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)

  Authors: Nikhil Kumar, Sanket Sonowal, Nishant

- E-mail Spam Detection Using Machine Learning

  Published in: 2023 4th International Conference for Emerging Technology (INCET)

  Authors: Babita Sonare, Gulbakshee J. Dharmale, Aditya Renapure, Harshit Khandelwal