PIMPRI CHINCHWAD EDUCATION TRUST's

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

SECTOR NO. 26, NIGDI PRADHIKARN, PUNE – 411044.

ML Mini Project Report

Spam Detection: Enhancing Email Security: Machine Learning-Based Spam Detection.

Academic Year: 2024 - 25

Student Name and PRN

Sr no.	Name	PRN
1	Prathamesh Dudhale	122B1B078
2	Pranjal Godse	122B1B090
3	NishantKumar Gupta	122B1B094

Under Guidance of : Dr. Mubin Tamboli

PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Sector No. 26, Pradhikaran, Nigdi, Pune – 411044

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **"Machine-Learning Based Email Spam Detection"** submitted by: -

Prathmesh Dudhale [122B1B078]

Pranjal Godse [122B1B090]

Nishantkumar Gupta [122B1B094]

under the supervision of **Prof. Mubin Tamboli** for the partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Engineering at Pimpri Chinchwad College of Engineering, affliated to Savitribai Phule Pune University, Pune** in the academic year 2024-25.

Prof: - Dr. Mubin Tamboli

(Project Guide)

Place: Pune

Date:

1.Introduction

Background: -

Spam mails are emerging as a severe concern that affects the communication at very personal as well as a business level. Automated and efficient filtering of spam mail is necessary due to an increased number of spam mails. Machine learning is proving to be an effective means through which spam could be differentiated from non-spam mail using elaborate algorithms and methodology-driven techniques. A well-trained machine learning model play very important role and can adapt to new patterns, and also can improve detection accuracy over time. This project can aim to develop a highly efficient spam detection system using supervised learning methods, ensuring enhanced email security and better experience for user.

Problem Statement: -

The primary challenge in spam detection is distinguishing legitimate emails from spam with high accuracy while minimizing false positives and false negatives. Many existing spam detection techniques struggle with evolving spam tactics, causing either excessive filtering of legitimate emails or failure to catch spam. This project aims to develop a highly efficient spam detection system using supervised learning methods, ensuring enhanced email security and better user experience.

Objectives :-

- Develop an efficient machine learning model for email spam detection.
- Improve email security by reducing spam messages.
- Use NLP techniques for feature extraction and classification.

Scope of the Project

- This system basically classifies emails as spam or non-spam emails using ML algorithms.
- System uses supervised learning models with labelled email datasets.
- Implementation basically focuses on real-time spam filtering for more accuracy.
- The project aims to minimize false positives and negatives in spam detection.

2. Literature Review

Overview of existing solution: -

Traditional spam filters basically depends on rule-based systems, blacklists, or keyword matching to detect spam emails. Bayesian classifiers and Support Vector Machines (SVM) was commonly used in earlier models. These systems perform very well in structured environments and for known spam patterns.

Limitations of existing solution: -

- Static rules and patterns are not effective against dynamic spam tactics.
- High false positive rate causes important emails to be marked as spam.
- Lack of adaptability for new spam techniques.
- Limited scalability and performance when processing large data of emails.

3. Proposed System: -

Algorithm: -

Random forest Algorithm

The project employs the Random Forest Algorithm, a powerful ensemble learning technique that combines multiple decision trees to improve accuracy and reduce overfitting. It works by creating numerous decision trees during training and averaging their predictions to classify emails as spam or non-spam.

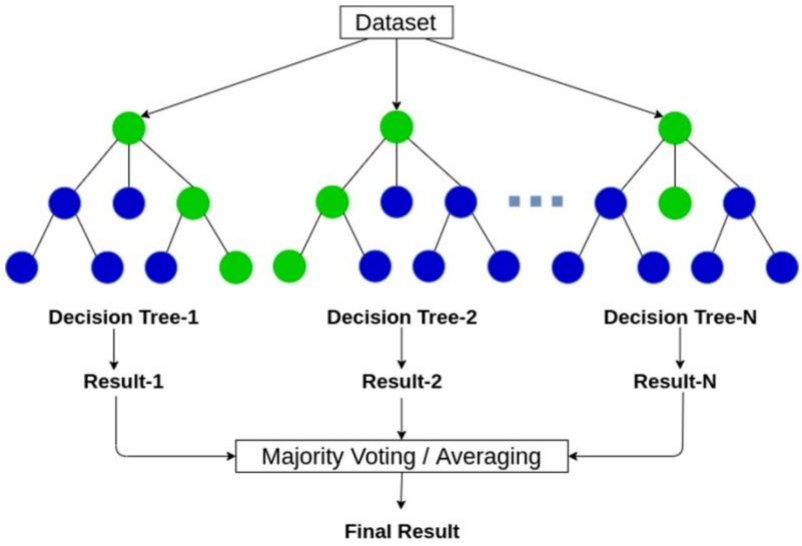
Random Forest Formula:

- Generate multiple decision trees using random subsets of data.
- Each tree provides a classification vote.
- The final prediction is based on the majority vote of all trees.

Advantages of Random Forest:

- High accuracy compared to individual decision trees.
- Robust to noise and overfitting.
- Works well with large datasets and high-dimensional data.

Random Forest



Methodology: -

Data Collection

- Dataset: The email dataset and publicly available spam email datasets.
- Features include:
  - Word Frequency
  - Presence of specific spam-related keywords
  - Email header analysis

Data Preprocessing

- Removing stop words, punctuation, and special characters.
- Converting text to lowercase to maintain uniformity.
- Breaking down emails into individual words.

  Converting words into their base form.
- Splitting the dataset into training and testing sets.

Feature Engineering

- Extracting essential text-based features such as:
  - Frequency of common spam words.

  - Presence of excessive capitalization and special symbols.
  - Length of the email subject and body.
- Applying feature selection techniques to improve model efficiency.

Model Selection and Training

- Using the Random Forest Classifier, an ensemble learning method.
- Training multiple decision trees and aggregating their predictions.
- Hyperparameter tuning:
  - Number of trees
  - Depth of trees
  - Minimum samples per leaf
- Cross-validation to evaluate model generalization.

Model Evaluation

- Performance metrics used:
  - Accuracy: Measures overall correctness of predictions.
  - Precision: Indicates how many predicted spam emails are truly spam.
  - Recall: Measures the ability to detect spam emails correctly.
  - F1-score: Harmonic mean of precision and recall.
  - Confusion Matrix: Analyses true positives, true negatives, false positives, and false negatives.

Deployment Considerations

- Integrating the trained model into an email server or API.
- Real-time classification of incoming emails.
- Updating the model periodically with new spam trends.

Advantages: -

- Adaptability: The model can be retrained with new data to stay updated with evolving spam tactics.
- Real-time Processing: This can be integrated into email services for live spam detection.
- Scalability: The algorithm can handle large datasets efficiently.
- Robustness: Resistant to overfitting due to multiple decision trees.

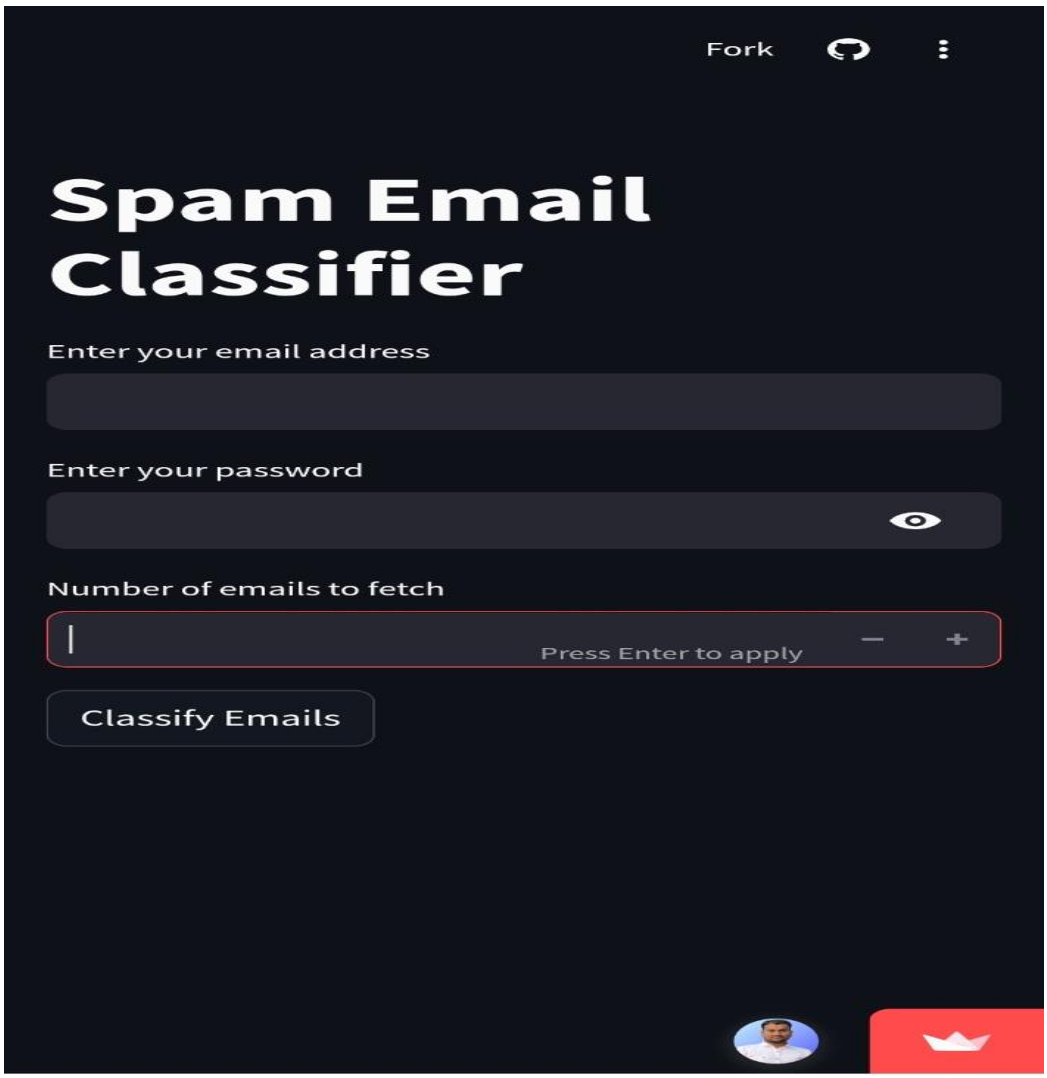
4. Implementation and Result: -

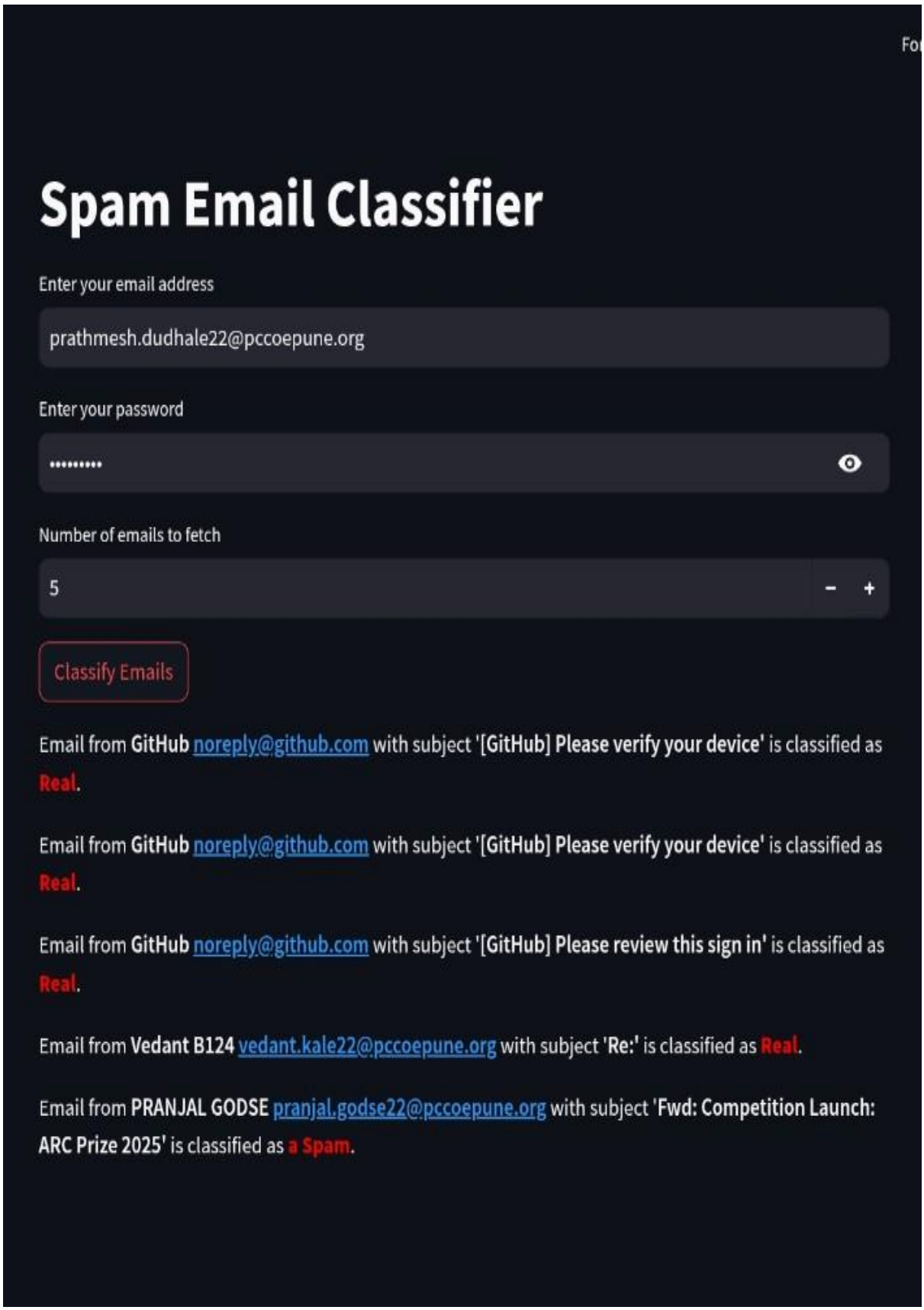
Tools and Technologies: -

- Programming Language: Python
- Libraries: sci-kit learn, pandas, NumPy, matplotlib
- Machine Learning Model: Random Forest Classifier
- Data Processing: Natural Language Processing (NLP),
- Tools : Streamlit

Result: -

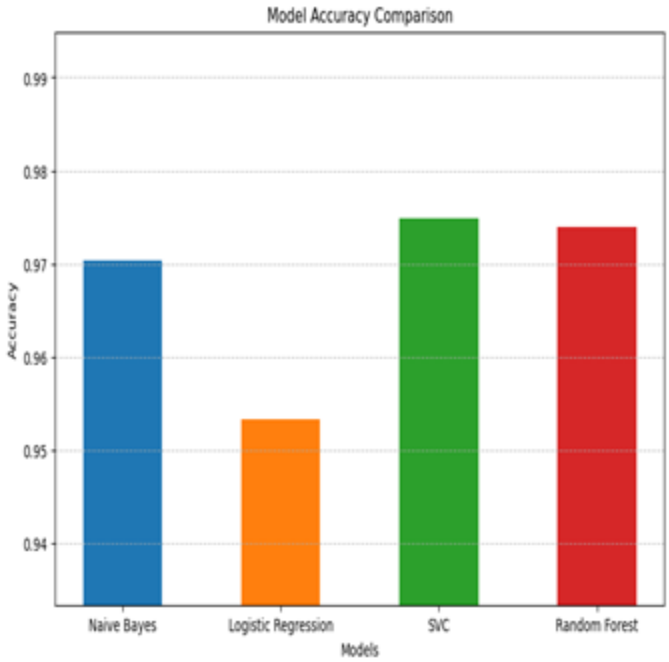
- Best Model: Random Forest Classifier.
- Accuracy: 98% on the test dataset.
- Confusion Matrix Analysis:
    - High precision and recall scores.
    - Minimal false positives and false negatives.



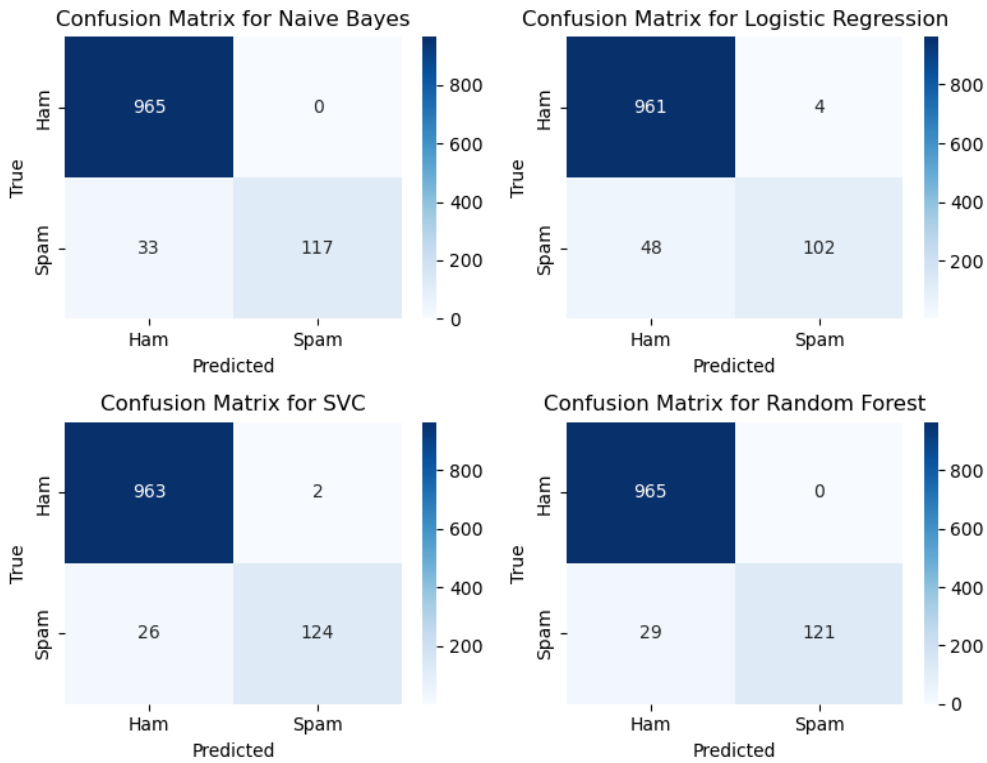

Spam Email Classifier
Enter your email address
prathmesh.dudhale22@pccoepune.org
Enter your password
Number of emails to fetch
5
Classify Emails
Email from GitHub noreply@github.com with subject '[GitHub] Please verify your device' is classified as Real.
Email from GitHub noreply@github.com with subject '[GitHub] Please verify your device' is classified as Real.
Email from GitHub noreply@github.com with subject '[GitHub] Please review this sign in' is classified as Real.
Email from Vedant B124 vedant.kale22@pccoepune.org with subject 'Re:' is classified as Real.
Email from PRANJAL GODSE pranjal.godse22@pccoepune.org with subject 'Fwd: Competition Launch: ARC Prize 2025' is classified as a Spam.

Visualization

- Comparision of the models



- Actual vs. Predicted Values plot demonstrated model's prediction accuracy



Conclusion: -

successfully implements an email spam detection system using the Random Forest classifier, which provides high accuracy and robustness against overfitting. The ensemble-based approach enhances reliability, making it suitable for real-world spam detection. Future improvements may include deep learning techniques and real-time filtering mechanisms to further enhance efficiency.

References: -

Welcome to Python.org

Get started with Streamlit - Streamlit Docs

pandas - Python Data Analysis Library

scikit-learn: machine learning in Python — scikit-learn 1.5.2 documentation

Links:-

App link - https://prathmeshdudhale.streamlit.app/

Repository link - https://github.com/PrathmeshDudhale96/ML-Project.git