International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 06 | June - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Email Spam Filtering System using Machine Learning

N. Renuka
BTech
School of Engineering, Hyd, India
2111cs020397@mallareddyuniversity.ac.in

Shaik Rezwan Ali
BTech
School of Engineering, Hyd, India
2111cs020399@mallareddyuniversity.ac.in

T. Rishitha Reddy
BTech
School of Engineering, Hyd, India
2111cs020401@mallareddyuniversity.ac.in

A. Revanth Reddy
BTech
School of Engineering, Hyd, India
2111cs020398@mallareddyuniversity.ac.in

L. Rishitha Goud
BTech
School of Engineering, Hyd, India
2111cs020400@mallareddyuniversity.ac.in

Guide: Prof. C M Preethi

Assistant Professor

School of Engineering,

MALLA REDDY UNIVERSITY, HYD, INDIA

Abstract: This project involves building an email management system using machine learning. It will sort emails into categories, help users organize their emails, and handle emails in multiple languages. For email classification, it will use the Naive Bayes algorithm. User feedback will be used to personalize the filtering process. Separate models will be developed for English and Hindi using Natural Language Processing (NLP). To handle infrequent login attempts, the system will detect emails containing login information. The project will focus on data collection, model evaluation, security, and scalability. A user-friendly interface will be designed, and advanced techniques like sentiment analysis and improved Hindi NLP will be explored.Flags emails containing login information of frequent logged in information of the user.This system can greatly improve how users manage their emails by addressing key problems and offering a versatile and dependable solution.

Introduction:

This project optimizes email management through Naive Bayes algorithms for spam detection and personalized organization in English and Hindi. Utilizing distinct Natural Language Processing models, it ensures seamless multilingual support. Emphasizing user security, the system flags suspicious login-related emails during low-activity

periods from unfamiliar interfaces. Advanced features, like sentiment analysis and improved Hindi NLP, enhance the user experience. The project prioritizes comprehensive data collection, rigorous model evaluation, robust security, and a user-friendly interface.

II. Literature Review:

Spam emails remain a significant nuisance and security threat for email users. Effective spam filtering is crucial for protecting inboxes and mitigating phishing attacks. This literature survey explores three prominent algorithms used in spam filtering: Naive Bayes, Support Vector Machines (SVMs), and Logistic Regression. We will discuss their strengths, weaknesses, and how they contribute to a comprehensive spam filtering approach. Naive Bayes: Efficient, effective, scalable, but limited in complexity and susceptible to new tactics. Support Vector Machines (SVMs) Powerful for complex patterns, highly accurate, but computationally expensive. Logistic Regression Offers balance between accuracy and efficiency, good for feature analysis and prediction, may not be as strong for highly complex spam. Spam filters often combine algorithms with blacklists and heuristic rules for a comprehensive approach. The best algorithm choice depends on factors like email volume, processing speed, and targeted spam types. In addition to algorithm selection, continuous monitering and adaptation are crucial to counter evolving spam tactics. Ensemble methods, which combine multiple algorithms, cam further enhance filtering accuracy by leveraging the

strengths of each approach, Furthermore, incorporating user feedback mechanisms allows for dynamic adjustments, improving overall detection rates. Ultimately, a multifaceted approach integrating various techniques ensures robust protection against spam threats while minimizing false positives.

Existing System:

1.Traditional email systems often rely on basic spam filters, lacking advanced algorithms for effective spam detection.
2.Limited support for multilingual communication, especially in languages like Hindi.
3.Generalized email organization without considering individual user preferences or utilizing sentiment analysis.
4.Security measures may be basic, lacking the ability to identify and flag suspicious login-related emails during low-activity periods.

Proposed System :

1.Implementation of Naive Bayes algorithms for robust spam detection, enhancing the accuracy of filtering unwanted emails.
2.Introduction of multilingual support, with specialized models for English and Hindi, ensuring improved communication capabilities.
3.User-specific email organization, utilizing feedback and sentiment analysis to tailor categorization based on individual preferences.
4.Strengthened security measures to identify and flag suspicious login-related emails, particularly during periods of low login activity from unfamiliar interfaces.
5.User-friendly interface design, incorporating advanced features like sentiment analysis and enhanced Hindi natural language processing for an improved user experience.

III.Problem Statement:

This project addresses email management challenges using Naive Bayes algorithms for spam detection and user-specific organization in English and Hindi. It prioritizes user security, identifies suspicious login-related emails, and features a user-friendly interface with advanced capabilities like sentiment analysis and enhanced Hindi NLP.

IV. Methodology:

METHODS AND ALGORITHMS :

CountVectorizer:

This is a text processing technique that converts a collection of text documents into a matrix of token counts. Each row in the matrix represents a document, and each column represents a unique word (token) in the corpus. The value in each cell

of the matrix represents the frequency of the corresponding word in the document.

Multinomial Naive Bayes (MultinomialNB):

Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of independence between features. The Multinomial Naive Bayes variant is suitable for classification with discrete features (such as word counts) and is commonly used in text classification tasks.

Pipeline:

A pipeline is used to chain together multiple processing steps into a single object. It allows for seamless integration of data preprocessing and model training/evaluation.

Train-Test Split (train_test_split):

This method splits the dataset into two subsets: one for training the model and the other for testing its performance. It helps evaluate the model's ability to generalize to unseen data.

It randomly splits the dataset into training and testing sets according to the specified ratio (in this case, 80% training data and 20% testing data).

Model Evaluation Metrics:

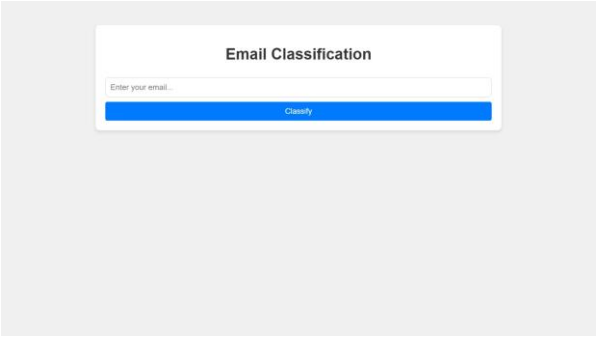
Confusion Matrix: A confusion matrix is a table that summarizes the performance of a classification model. It shows the counts of true positive, true negative, false positive, and false negative predictions.

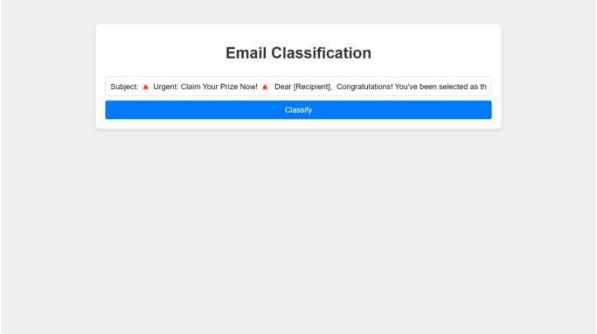
Classification Report: This report provides metrics such as precision, recall, and F1-score for each class in the dataset. Precision measures the proportion of true positive predictions among all positive predictions, recall measures the proportion of true positives correctly identified, and F1-score is the harmonic mean of precision and recall.

Accuracy: Accuracy measures the proportion of correctly classified examples out of the total examples. While accuracy is a commonly used metric, it may not be sufficient for imbalanced datasets where one class is much more prevalent than others.

V. Experimental Results:

OUTPUT:





VI. Conclusion:

This interface provides a smooth and efficient way to access and manage your emails, ensuring you can always find the information you need quickly. By integrating these technologies, this project aims to create a secure and streamlined email experience, enhancing both communication and efficiency.

VII. Future Work:

User Feedback Integration: Allow users to provide feedback on email classifications to further refine the model and reduce false positives/negatives.

Real-Time Monitoring: Develop mechanisms to monitor email traffic in real-time, detecting and responding to emerging spamming techniques quickly.

Adversarial Training: Train the model against adversarial attacks to make it more robust against sophisticated spamming techniques.

Behavioral Analysis: Incorporate behavioral analysis of users' email interactions to identify suspicious patterns and improve spam detection.

VIII. REFERENCES:

1.T. Stephenson, "An introduction to bayesian network theory and usage," Jan. 2000.

2.V. Christina, S. Karpagavalli, and G. Suganya, "A study on email spam filtering techniques,"

International Journal of Computer Applications, vol. 12, no. 1, Dec. 2010. doi: 10.5120/1645-2213

3.S. Yoo, "Machine learning methods for personalized email prioritization," Jan. 2010.

4.Idris and A. Selamat, "Improved email spam detection model with negative selection algorithm and particle swarm optimization," Applied Soft Computing, vol. 22, pp. 11–27, Sep. 2014. doi:

10.1016/j.asoc.2014.05.002.

5.J. Alqatawna, H. Faris, K. Jaradat, M. Al-Zewairi, and O. Adwan, "Improving knowledge based spam detection methods: The effect of malicious related features in imbalance data distribution," International Journal of Communications, Network and System Sciences, vol. 08, no. 05, pp. 118– 129, 2015. doi: 10.4236/ijcns.2015.85014.