ENHANCING SPAM EMAIL DETECTION USING MACHINE LEARNING ALGORITHMS

Mr. B. V. Sathish Kumar[1], Nelluri Tej Kumar[2], Marthati Revanth[3], Nallam Sai Manikanta[4],Kadari Harikrishna[5]

[1] *Assisstant Professor, Electronics and Communication Department, Vasireddy Venkatadri Institute of technology , Andhra Pradesh, India*
[2-5] *Undergraduate Students, Electronics and Communication Department Vasireddy Venkatadri Institute of technology , Andhra Pradesh, India*

ABSTRACT

This paper introduces an innovative solution for enhanced spam email detection utilizing the Naive Bayes Multinomial algorithm. In response to the persistent challenge of spam emails inundating inboxes, our system leverages machine learning to achieve superior accuracy in distinguishing between spam (unsolicited) and ham (legitimate) emails. By harnessing the power of probabilistic classification, our approach effectively filters out unwanted messages, contributing to a more secure and streamlined email experience. Through continuous refinement, we prioritize accuracy improvement and have implemented mechanisms for user feedback to address any issues and enhance the system's performance further. Moreover, we have developed a user-friendly web interface that enables users to conveniently check the classification of their emails in real-time. This interface empowers users to efficiently manage their inbox by quickly identifying and filtering potential spam messages. Through rigorous experimentation and evaluation, our system demonstrates promising results, showcasing significant improvements in spam detection accuracy compared to traditional methods, not only enhances email security but also enhances productivity by minimizing the time and effort required to manage spam emails. With a remarkable accuracy rate of 97.85%, our system demonstrates significant advancements in spam email detection, affirming its efficacy in distinguishing between unsolicited spam and legitimate emails

Keywords: - *Spam Email Detection , Accuracy , Naive Bayes Multinomial Algorithm, probability and web page*

1. INTRODUCTION

The ubiquity of spam emails presents a persistent challenge in contemporary communication landscapes, inundating inboxes and compromising user experience and security. In response, this paper introduces an innovative solution aimed at enhancing spam email detection through the utilization of the Naive Bayes Multinomial algorithm. Leveraging the power of machine learning and probabilistic classification, our system endeavors to achieve superior accuracy in distinguishing between spam (unsolicited) and ham (legitimate) emails. The proliferation of spam emails not only undermines productivity but also poses significant security risks, including phishing attempts and malware dissemination. Therefore, the development of robust and efficient spam detection mechanisms is imperative to safeguarding user privacy and ensuring a streamlined email experience.

Moreover, recognizing the dynamic nature of spam emails and the evolving tactics employed by spammers, emphasizes continuous improvement and adaptation. To this end, we have implemented mechanisms for user feedback, enabling users to report any issues encountered and contribute to the refinement of the system. Additionally, we have prioritized the development of a user-friendly web interface, providing users with convenient access to real-time email classification checks. By empowering users to efficiently manage their inbox and filter potential spam messages, our system aims to not only enhance email security but also improve overall productivity.

Through rigorous experimentation and evaluation, it seeks to demonstrate significant advancements in spam detection accuracy compared to traditional methods, thereby addressing a critical need in contemporary email communication.

2. LITERATURE SURVEY

The field of machine learning (ML) has made significant strides in various domains, including text categorization and information retrieval. Sahami et al. [1] (1998) introduced a Bayesian approach to filter junk e-mail, highlighting the importance of probabilistic models in handling large-scale text data. McCallum et al. [2] (2000) further extended this concept by automating the construction of internet portals using ML techniques, showcasing the potential for ML in shaping online information environments. Manning et al. [3] (2008) provided a comprehensive introduction to information retrieval, emphasizing techniques for efficiently accessing and organizing textual data. This work laid a solid foundation for understanding the fundamental principles of retrieving relevant information from large document collections. Jurafsky and Martin [4] (2019) delved into natural language processing (NLP) and speech recognition, showcasing the intersection of ML with linguistic analysis and speech technologies. Sebastiani [5] (2002) contributed significantly to the field of automated text categorization by discussing various ML approaches and their applications in classifying textual data. This work highlighted the evolution of ML algorithms in handling diverse text categorization tasks. Additionally, the Scikit-learn library (Pedregosa et al. [6], 2011) emerged as a prominent tool for ML practitioners, providing a wide range of algorithms and tools for implementing ML models in Python.

3. DATASET:

The dataset consists of two columns: "class" and "text". The "class" column denotes the label assigned to each email instance, with values indicating whether the email is spam (1) or legitimate (0). The "text" column contains the textual content of the emails, including subject lines, message bodies, and sender information. With 5574 email instances, the dataset offers a diverse and representative sample of spam and legitimate emails, encompassing a wide range of textual features and characteristics.

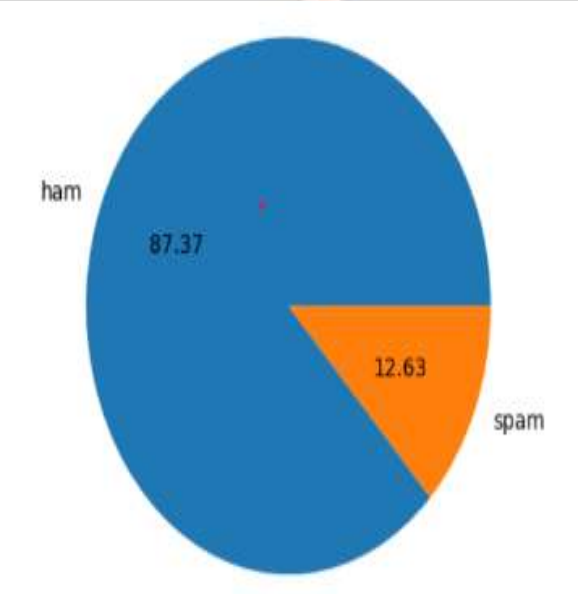

Fig-1: Dataset Partition

The dataset plays a crucial role in facilitating research and development efforts aimed at enhancing spam email detection systems. By providing a labeled corpus of spam and legitimate emails, researchers can train and evaluate machine learning models to accurately classify incoming emails based on their content. Moreover, the dataset enables researchers to analyze the underlying patterns and characteristics of spam emails, thereby informing the design of more robust and effective detection algorithms.

Row number	target	text
3416	ham	But i haf enuff space got like 4 mb...
3493	spam	You are being contacted by our dating service ...

Fig-2: Sample Data
Fig-3: Data Set Segmentation

4. PREPROCESSING

Preprocessing plays a crucial role in enhancing the effectiveness of machine learning models for spam email detection. The preprocessing pipeline typically involves several steps aimed at cleaning and transforming raw email data into a format suitable for model training.

	target	text
4193	ham	Can not use foreign stamps in this country
4631	ham	I'm in a meeting, call me later at
3416	ham	But i haf enuff space got like 4 mb...
3493	spam	You are being contacted by our dating service ...
919	ham	Dont talk to him ever ok its my word

Fig-4: Transforming Data

	target	text
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

Fig-5: Applying Binary Values

Firstly, text data is subjected to tokenization, where emails are divided into individual words or tokens. This step facilitates subsequent analysis by breaking down the text into its constituent elements. Following tokenization, common preprocessing techniques such as stop word removal and stemming or lemmatization are applied to eliminate noise and reduce feature dimensionality. Stop words, which are frequently occurring but semantically insignificant words such as "and", "the", and "is", are removed to prevent them from skewing the

model's predictions. Stemming and lemmatization techniques normalize words to their root forms, reducing redundancy and improving generalization.

	target	text	num_characters	num_words	num_sentences
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	2
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2
3	0	U dun say so early hor... U c already then say...	49	13	1
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1

Fig-6: Word Frequency Count

Additionally, feature engineering is often employed to extract relevant information from the text data. This may involve the creation of features such as word frequency counts, n-grams, and tf-idf (term frequency-inverse document frequency) vectors. These features capture the underlying patterns and characteristics of spam and ham emails, enabling the model to learn discriminative patterns effectively.

5. PROPOSED WORK

The proposed work builds upon the innovative solution outlined in the abstract, aiming to further enhance spam email detection efficiency and accuracy. We intend to delve deeper into the implementation of the Naive Bayes Multinomial algorithm, optimizing its parameters and fine-tuning its performance to achieve even higher accuracy rates in distinguishing between spam and legitimate emails. Additionally, we will explore the integration of advanced machine .

The Naive Bayes Multinomial architecture tackles text classification. It assumes independence between words (features) in a document. Based on Bayes' theorem, it calculates the probability of a document belonging to a class (e.g., spam) by considering the likelihood of each word appearing in that class and the overall class probability. During training, the model learns these probabilities from labeled documents. At classification time, it assigns a new document to the class with the highest probability, effectively performing probabilistic text categorization.

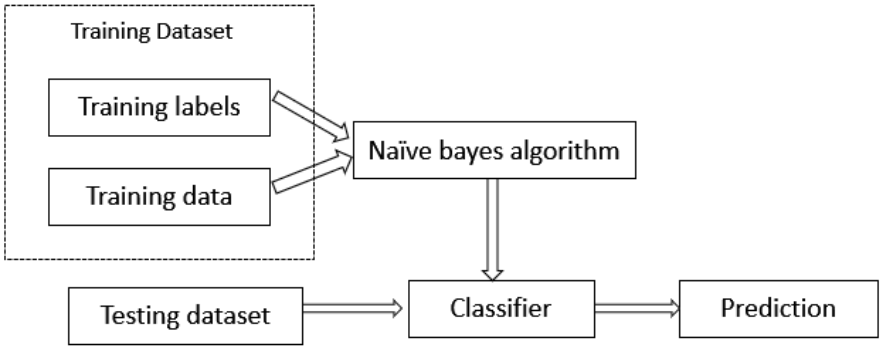

Fig-7:Architecture Of Naïve Bayes Algorithm

5.1 ALGORITHM SELECTION AND EVALUATION:

A comprehensive evaluation of the Multinomial Naive Bayes algorithm will be conducted to assess its suitability for spam email detection. This will involve implementing the algorithm on the prepared dataset and evaluating its performance metrics, including accuracy, precision. In Multinomial Naive Bayes, each document is represented as a bag-of-words, where the frequency of each word in the document is used as a feature. The algorithm assumes that the order of words doesn't matter and only considers the frequency of each word. Multinomial Naive Bayes estimates two types of parameters: class priors and conditional probabilities.

Class priors ($P(C_k)$) represent the probability of each class occurring in the dataset. They can be estimated by counting the frequency of each class in the training data. Conditional probabilities ($P(x_i|C_k)$) represent the probability of observing each word (x_i) given a particular class (C_k). These probabilities are estimated by counting the occurrences of each word in documents belonging to the corresponding class. To classify a new document, Multinomial Naive Bayes calculates the posterior probability of each class given the document's features (word frequencies) using Bayes' theorem. The class with the highest posterior probability is predicted as the label for the document

5.2 USER FEEDBACK INTEGRATION:

In order to improve the accuracy and robustness of the spam detection system, mechanisms for user feedback will be integrated into the classification process. A user feedback form will be designed to allow users to provide input on the classification of their emails, enabling iterative refinement of the algorithm based on user interactions

5.3 WEB INTERFACE DEVELOPMENT:

A user-friendly web interface will be developed to facilitate user interaction with the spam detection system. The interface will enable users to submit emails for classification, view the classification results, and provide feedback on the accuracy of the classification.

5.4 ACCURACY IMPROVEMENT AND PERFORMANCE EVALUATION:

The proposed spam detection system will undergo iterative refinement based on user feedback and performance evaluation results. Techniques such as hyperparameter tuning, feature engineering, and model retraining will be employed to improve the accuracy and efficiency of the system.

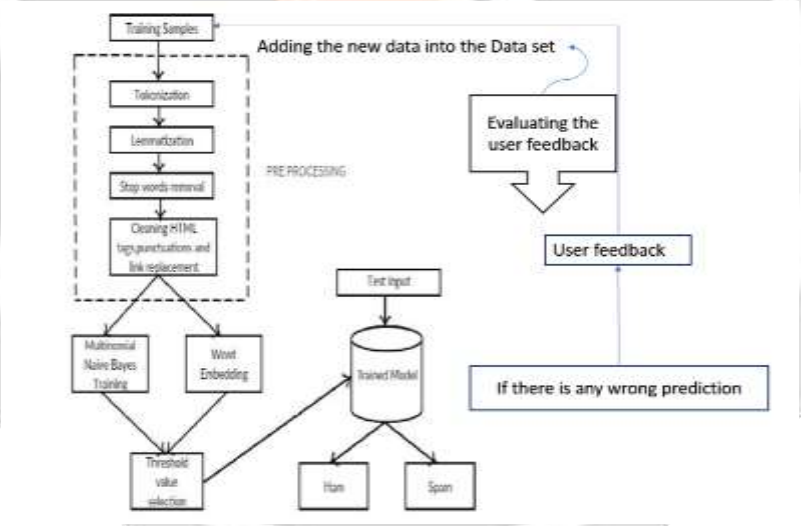

Fig-8: Flowchart Of The Model

5.6 MODEL TRAINING :

The Multinomial Naive Bayes algorithm is trained on the preprocessed and feature-extracted dataset. During training, the algorithm learns the parameters necessary to make predictions, including class priors and conditional probabilities.

Cross-validation techniques such as k-fold cross-validation are applied to assess the performance of the model. The dataset is divided into k subsets, and the model is trained and evaluated k times, each time using a different subset for validation. This helps to ensure that the model's performance is robust and not overly dependent on the particular choice of training and testing data. After training the model, it is evaluated using a separate test dataset that was not used during training. The model's performance metrics, including accuracy, precision, recall, and F1-score, are calculated to assess its effectiveness in classifying emails as spam or not spam.

6. TOOLS & TECHNIQUES USED:

6.1 STREAMLIT:

Streamlit is a Python library used for building interactive web applications for machine learning and data science projects. It simplifies the process of creating user interfaces with minimal code, making it easy to deploy machine learning models and visualizations.

6.2 PICKLE:

Pickle is a Python module used for serializing and deserializing Python objects. In this code, it is used to load the pre-trained machine learning model and CountVectorizer object from files (spam.pkl and vectorizer.pkl, respectively).

6.3 SCIKIT-LEARN (SKLEARN):

scikit-learn is a popular machine learning library in Python that provides a wide range of tools for data preprocessing, model building, and evaluation. In this code, it is used for feature extraction (CountVectorizer), model training (Multinomial Naive Bayes), and performance evaluation (accuracy_score, precision_score).

6.4 WIN32COM.CLIENT:

This module is part of the PyWin32 library and is used for interacting with the Windows COM (Component Object Model) system. In this code, it is used to enable text-to-speech functionality for speaking classification results.

6.5 PANDAS:

Pandas is a powerful data manipulation library in Python. In this code, it is used for loading and manipulating datasets stored in CSV format.

Techniques:

6.6 DATA PREPROCESSING:

Data preprocessing techniques such as tokenization, stop word removal, and stemming are used to clean and normalize the textual data before feeding it into the machine learning model. This ensures uniformity and consistency in the feature representations.

6.7 FEATURE ENGINEERING:

Feature engineering involves selecting and creating relevant features from the raw data to improve the performance of the machine learning model. In the context of spam email detection, features such as word frequencies, n-grams, and TF-IDF (Term Frequency-Inverse Document Frequency) vectors can be extracted from the email text.

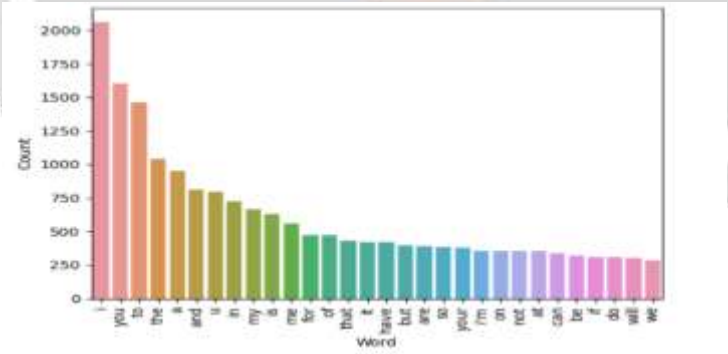

Fig-9: Graph Of Word Freq Count

6.8 CROSS-VALIDATION:

Cross-validation is a technique used to assess the generalization performance of machine learning models.It involves splitting the dataset into multiple subsets, training the model on a subset, and evaluating its performance on the remaining subset. This process is repeated multiple times to obtain robust performance estimates. Hyperparameter Tuning: Hyperparameter tuning involves selecting the optimal values for the hyperparameters of the machine learning algorithm to improve its performance. Techniques such as grid search and random search can be used to systematically explore the hyperparameter space and identify the best combination of parameters.

7. PERFORMANCE METRICS

7.1 CALCULATION OF ACCURACY: Count the number of instances where the predicted label matches the true label and divide it by the total number of instances in the test dataset.
Mathematically, accuracy is calculated as

Accuracy=(Number of Correct Predictions/Total Number of Predictions)×100%

Model accuracy is = 97.84 %

7.2 CALCULATION OF PRECISION: Divide the number of true positives by the sum of true positives and false positives.
Mathematically, precision is calculated as:

Precision= (True Positives + False Positives) / True Positives

Model precision is = 91.44%

8. RESULT

8.1 WEB PAGE IMPLEMENTATION:

A web page was developed to allow users to input email messages and classify them as spam or ham (non-spam) using the trained Multinomial Naive Bayes model. The web page interface provides a user-friendly environment for email classification, with real-time feedback on the classification result and display of the model accuracy.

8.2 EMAIL CLASSIFICATION:

Upon accessing the web page, users are presented with a text input field where they can enter the content of their email messages. After entering the email content, users can initiate the classification process by clicking the designated button.

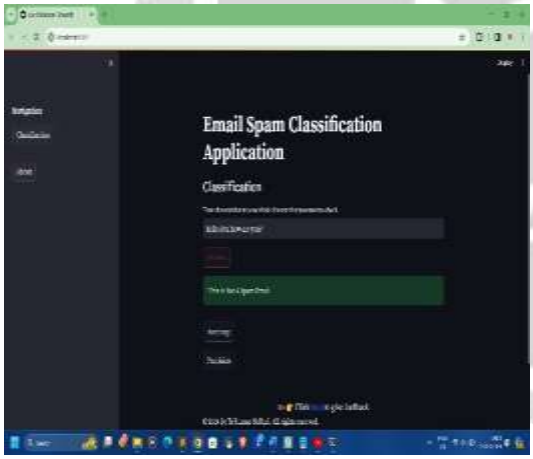

Fig-10: Result Of Ham Mail

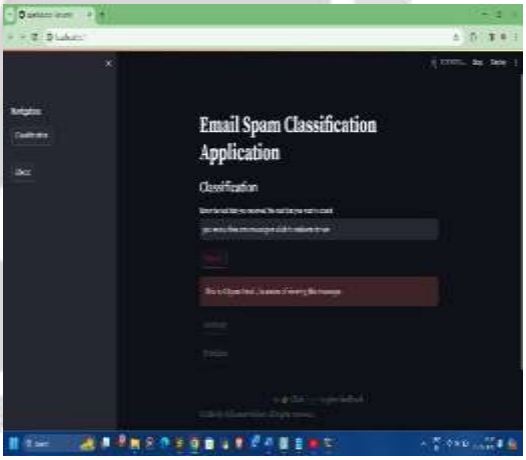

Fig-11: Result Of Spam Mail

8.3 REAL-TIME FEEDBACK:

Upon clicking the classification button, the web page leverages the trained Multinomial Naive Bayes model to classify the input email message as spam or ham. The classification result is displayed to the user in real-time on the web page interface.

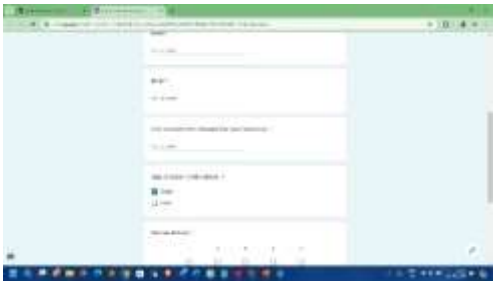

Fig-12: User Feedback Form

8.4 DISPLAY OF MODEL ACCURACY:

In addition to providing classification results, the web page also displays the accuracy of the trained model. The model accuracy, representing the percentage of correctly classified instances out of the total, is prominently featured on the web page interface to provide users with insights into the performance of the classification system.

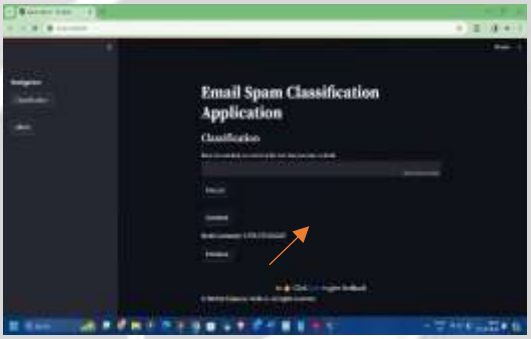

Fig-13: Model Accuracy

8.5 ENHANCED USER EXPERIENCE:

The integration of real-time classification feedback and model accuracy display enhances the user experience by providing users with immediate insights into the classification results and the reliability of the underlying model. This functionality empowers users to make informed decisions regarding the classification of their email messages.

9. CONCLUSION

The development of an enhanced spam email detection system utilizing the Multinomial Naive Bayes algorithm represents a significant advancement in combating unsolicited email messages. Through the utilization of machine learning techniques and real-time classification capabilities, the system provides users with an effective tool to distinguish between spam and legitimate emails.

The implementation of a user-friendly web interface further enhances the accessibility and usability of the system, allowing users to conveniently input email content and receive instant classification results. By integrating real-time feedback and displaying the accuracy of the underlying model, the web interface empowers users to make informed decisions regarding the classification of their email messages, thereby enhancing email security and productivity.

REFFERENCES

[1] Sahami, Mehran, et al. "A Bayesian approach to filtering junk e-mail." Proceedings of the AAAI workshop learning for text categorization. Vol. 62. 1998.

[2] McCallum, Andrew, et al. "Automating the construction of internet portals with machine learning." Information Retrieval 3.2 (2000): 127-163.
[3] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. Introduction to Information Retrieval. Cambridge University Press, 2008.
[4] Jurafsky, Dan, and James H. Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2019.
[5] Sebastiani, Fabrizio. "Machine learning in automated text categorization." ACM computing surveys (CSUR) 34.1 (2002): 1-47.
[6] Scikit-learn: Machine Learning in Python. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Journal of Machine Learning Research, 12(Oct), 2825-2830.