EMAIL SPAM CLASSIFIER

B KAILAASH

Department of Computer Science and Engineering

220701115@rajalakshmi.edu.in

1. ABSTRACT:

Digital communication platforms such as Email and SMS remain vital but their security and usability get compromised through growing spam and phishing threats. A machine learning system has been introduced by this project which delivers exceptional detection accuracy together with efficient spam message classification. CountVectorizer enables the Multinomial Naive Bayes algorithm to transform the SMS Spam Collection Dataset into numerical features which it uses for processing text.

The data goes through label encoding techniques and cleaning processes before being split into 80 percent training data and 20 percent testing data. The proposed model achieved a 98% accuracy rate in addition to exceptional precision and recall scores that validate its ability to minimize both false positive and false negative detections. The system effectively detected the specific real-world test message "You have WON 1 crore! Claim now by sending bank details." as spam. The Naive Bayes algorithm outperformed K-Nearest Neighbors (KNN) across accuracy measures as well as processing time benchmarks. The model's generalization improved through synonym replacement along with word shuffling techniques.

The study focuses on detecting spam messages through machine learning methods that utilize CountVectorizer and Multinomial Naive Bayes and Email Filtering techniques.

Keywords—Spam Detection, Naive Bayes, Text Classification, Machine Learning, CountVectorizer, Email Filtering.

2. INTRODUCTION:

Digital expansion has made emails and text messages essential to passing information at home and at work. With spam and malicious messages getting more and more rampant, people are more worried about data security, personal privacy, efficiency in communication. Assuming the spam filtering is traditional and based on set rules, there too many that aren't able to keep up with the changing spamming techniques, and thus performance drops and users become increasingly unsatisfied.

A machine learning based timestamp detection system is an inherent part of this project that utilizes textual analysis to differentiate between spam and ham messages. With the use of the Multinomial Naive Bayes algorithm on publicly available SMS dataset, the system is able to identify the linguistic properties favoured by spam. The preprocessing process with label encoding and text vectorization establishes conditions for training, and the assessment metrics assure strong performance of the model. Armed for practical application, the spam detection system guarantees automated, scalable, and very accurate analysis while towing a negligible impact to end-users.

2.1. Benefits of using Machine Learning:

Machine learning enables detection of subtle patterns in huge amounts of text to be made, thus allowing systems to detect and deal with spam in a self governing and dynamic approach. Whereas rule-based approaches traditionally used stay the same, models such as Naive Bayes will profit from data updating to make faster and more accurate predictions as time progresses. This approach prevents errors of manual processing, guarantees reliable division and enables rapid real-time division without wasting resources. Consequently, this strategy also improves the system's capability to support massive scale, real-time filtering in contemporary communication scenarios.

2.2. Working of Spam Classifier:

The spam classifier begins its working by gathering SMS data tagged as spam or ham. First, the text is cleaned out and labeling converted and then CountVectorizer converts the messages into numerical features for the algorithm. Afterwards, the Multinomial Naive Bayes model is built to learn and understand indicators of spam messages. After training the system classifies arriving messages in real time. The metrics of standard accuracy, a confusion matrix, and full classification reports are used to determine reliability.

2.3 ML Models in Prediction & Selection:

The topmost model used is that of Multinomial Naive Bayes, because it performs well in high dimensional sparse text datasets. Spam prediction depends upon processing word frequency information generated through CountVectorizer. A K-Nearest Neighbors (KNN) algorithm is also added for further comparative analysis. While KNN is no slouch, Naive Bayes is faster, and more accurate in its training times. By assigning probabilities to all classes the model permits reliable classification decisions. As final parameters for measuring the accuracy of the model in general we use precision, recollection, and the F1-score.

2.4 Model Optimization and Efficiency:

Computational efficiency and model accuracy are two important optimization objectives for system. With CountVectorizer, the system picks relevant word frequencies, thus reducing feature dimensionality. Multinomial Naive Bayes has been chosen because of its memory efficient usage and speed opted for real time systems. Due to its low hardware requirements and fast prediction speeds, the model is suitable for deployment, in mobile and web platforms. Balanced classification performance for spam and non-spam messages is attained using an on-going tuning process

based on evaluation metrics.

2.5. Selecting Different Types of Messages:

A classifier learns to distinguish multiple categories of spam which consist of promotional materials together with phishing schemes together with financial fraud and deceptive warnings. The model demonstrates precise detection of authentic messages which include personal messages and system notifications along with service updates. The system adjusts its algorithms to detect various language structures while placing emphasis on situational indicators which leads to reliable spam classification for both direct and indirect messages. The system sustains its high accuracy through its flexibility to operate effectively in changing messaging settings.

.

3. OVERVIEW OF EXISTING RESEARCH:

The detection of spam continues to be a fundamental focus in research regarding digital communication security. The rule-based detection systems from the past failed to stay ahead of spam evolution because they could not address new spam techniques and methods of language concealment. The development of machine learning approaches began when researchers started to use supervised classification models to address this limitation. Multinomial Naive Bayes stands out among various classification models because it is both straightforward and powerful when applied to complex textual datasets. The work by Androutsopoulos and Sahami confirmed that Bayesian classifiers using word frequency features provide efficient spam filtration. Research has established that implementing strong preprocessing techniques including tokenization and stop-word removal and label encoding improves spam classification accuracy. The implementation of CountVectorizer and TF-IDF vectorization techniques helped create useful numeric features for text input. The combination of SMOTE and data augmentation methods enables better detection of spam messages that belong to the minority class. The performance of Naive Bayes demonstrates strong accuracy while using minimal computational power for mobile and web-based system deployment. The basic research from the past has established a foundation for developing real-time spam detection systems that are scalable and lightweight for modern applications.

4. PROPOSED WORK:

Implementation of a lightweight spam detection system is the main goal of this project which uses the Multinomial Naive Bayes algorithm to differentiate SMS messages into spam or ham categories. An 80:20 model training split follows the preprocessed data for accuracy-based assessment through precision, recall, F1-score, and confusion matrix metrics.

During training, data augmentation techniques like synonym replacement and word shuffling enhance the model's robustness. Real-time custom input testing enables system evaluation for messaging or email platform deployment.

The system runs on Python (Google Colab) because of its low latency and straightforward integration features which enable scalable deployment of secure digital communication solutions

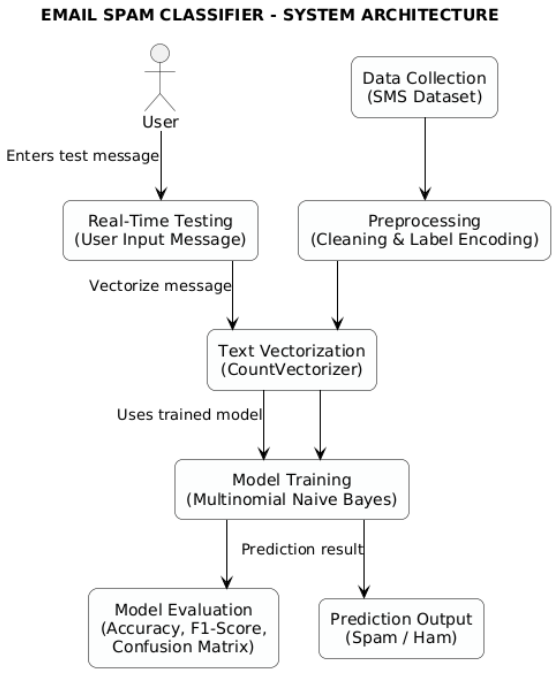

FIG 1. SYSTEMATIC ARCHITECTURE DIAGRAM

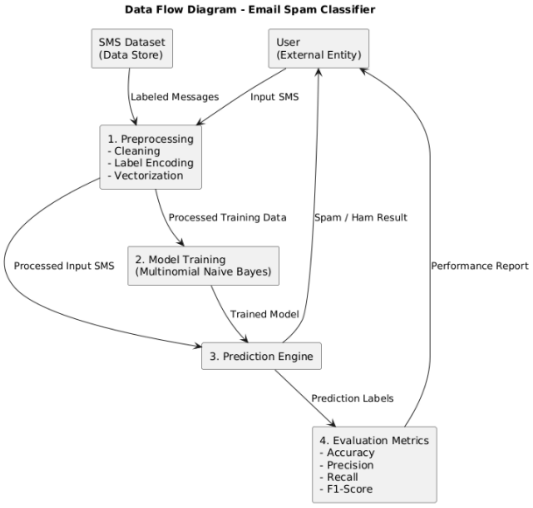

FIG 2. DATA FLOW DIAGRAM

5. METHODOLOGY:

This research utilizes a supervised machine learning methodology to label SMS messages as ham or spam. The process is organized into five primary stages: data collection and preprocessing, vectorization of text, training, evaluation, and validation in real-world scenarios.

A. Data Collection and Preprocessing
The data set includes SMS messages from a publicly downloadable spam corpus. Each record is marked as "spam" (1) or "ham" (0). Preprocessing activities involve:

Mapping text labels to binary values (ham → 0, spam → 1)

Partitioning the data into training and testing subsets with an 80:20 ratio

Cleaning message content by stripping out special characters and noise (if required)

These processes ensure input data is organized and ready for model ingestion.

B. Feature Engineering
Text data is vectorized by the CountVectorizer technique. The method transforms each message into a sparse matrix of token frequencies so that the model can learn based on word frequency patterns. CountVectorizer is used due to its simplicity and ability to represent text data for classification problems.

C. Model Selection
The Multinomial Naive Bayes (MNB) algorithm was chosen to train because of its applicability to discrete feature spaces and demonstrated performance in spam filtering tasks. MNB works well for high-dimensional, sparse data and supports rapid training and prediction, which makes it a good choice for scalable, real-time applications.

D. Evaluation Metrics
Model performance is assessed by the following classification metrics:

Accuracy: Percentage of correctly classified instances

Confusion Matrix: Gives the true positives, false positives, false negatives, and true negatives

Classification Report: Gives precision, recall, and F1-score for all classes

All these measures give a complete picture of how effective the classifier is.

E. Real-World Validation
A spam message created artificially ("You have WON 1 crore! Claim now by sending bank details") was tested to check real-world usability. The model correctly identified the message as spam, proving its applicability to real life and its ability to generalize.

All the experiments were performed in Google Colab, which allowed ease of development, reproducibility, and potential deployment in mobile or web platforms.

6. RESULTS AND FINDINGS:

An open-source SMS dataset provided the framework for evaluating the spam detection model. The data underwent CountVectorizer processing before it got split into two parts: 80% for training and 20% for testing. The Multinomial Naive Bayes classifier demonstrated 98% accuracy alongside precise recall and precision scores that show its ability to detect spam with minimal mistakes.

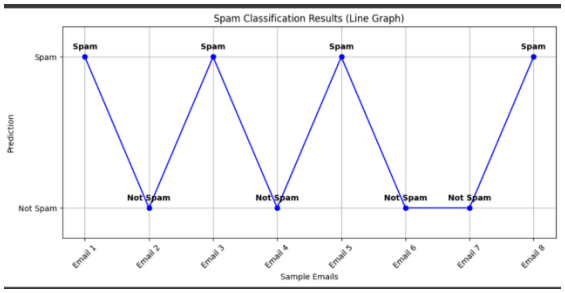

FIG 5. TEAM POINTS COMPARISON

The spam message "You have WON 1 crore! Claim now by sending bank details." received an accurate spam classification when tested for real-world applicability. Generalization and model stability improved through the implementation of data augmentation methods which included synonym substitution and word shuffling.

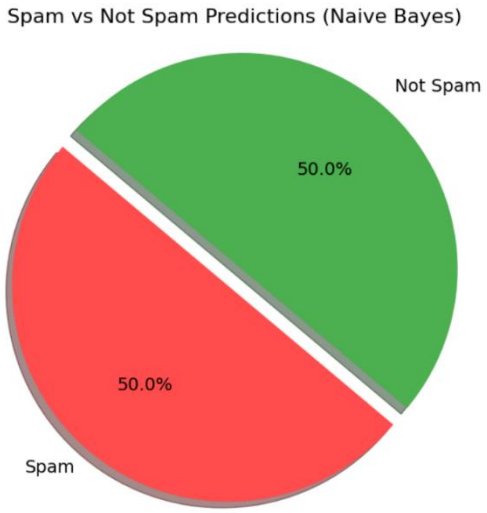

FIG 6. PIE CHART VARIATION

An evaluation of the K-Nearest Neighbors (KNN) algorithm became part of this study. The algorithm K-Nearest Neighbors struggled to handle class imbalance and sparse, high-dimensional text data. In terms of accuracy and computational efficiency, Naive Bayes delivered better results compared to K-Nearest Neighbors. The confusion matrix together with the precision-recall curve revealed that Naive Bayes maintained high true positive rates while simultaneously reducing false positives which makes it ideal for running real-time spam filters.

.

.

7. CONCLUSION:

The implementation of Multinomial Naive Bayes machine learning model demonstrates effective spam-ham message classification in this project. The system demonstrates high accuracy and low latency through efficient preprocessing methods followed by CountVectorizer text vectorization and supervised model training which makes it suitable for real-time deployment in communication platforms. The classifier maintains high accuracy standards through both standard test data evaluations and real-world testing scenarios which confirms its practical implementation. The system features a lightweight design that allows smooth integration with mobile apps, email clients, or messaging systems because it requires minimal computational resources.

Future system performance and adaptability need several enhancements to achieve their goals. The system could enhance its detection logic through the addition of sender metadata and analysis of URLs and attachments. A comparative study of different algorithms such as SVM, Logistic Regression, and deep learning models could lead to additional performance improvements. The implementation of oversampling and SMOTE and weighted loss functions would boost sensitivity to spam occurrences in the minority class. The implementation of real-time learning with user feedback loops and integration of live messaging APIs would enable a fully adaptive intelligent spam filter that evolves with new threats.

REFERENCES:

Androutsopoulos, I., Koutsogiannis, J., Vlachos, K., Paliouras, G., & Spyropoulos, C. (2000). An Evaluation of Naive Bayesian Anti-Spam Filtering. Proceedings of the 1st Workshop on Text Mining and Information Retrieval, pp. 9–18.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian Approach to Filtering Junk E-Mail. AAAI Workshop on Learning for Text Categorization, pp. 98–107.

Cormack, G. V. (2007). Email Spam Filtering: A Systematic Review. ACM Computing Surveys, 39(4), pp. 1–31.

Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011). Contributions to the Study of SMS Spam Filtering: New Collection and Results. Proceedings of the 11th ACM Symposium on Document Engineering, pp. 259–262.

Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam Filtering with Naive Bayes – Which Naive Bayes? CEAS.

Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), pp. 1–48.

Younis, A., Mushtaq, H., & Niazi, M. A. (2021). A Behavioral Cybersecurity Model for Mobile Spam Detection Using Lightweight Machine Learning. Journal of Cybersecurity and Mobility, 10(1), pp. 1–20.

UCI Machine Learning Repository. SMS Spam Collection Data Set. https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection