Email spam classification using tokenization and stemming

Ibrahim
Computer Science
International Islamic University
Malaysia, Kuala Lumpur, Malaysia
i.schigdar9@gmail.com

Ismail Abu Saiid
Computer Science
International Islamic University
Malaysia, Kuala Lumpur, Malaysia
ism.schigdar@gmail.com

Muhammad Afiq Bin Mohd Ara
Computer Science
International Islamic University
Malaysia, Kuala Lumpur, Malaysia
Afiq2000184@gmail.com

Faza Amal Sophian
Computer Science
International Islamic University
Malaysia, Kuala Lumpur, Malaysia
Fazasophian123@gmail.com

ABSTRACT

Email communication has become integral to daily interactions, but the proliferation of email spam presents significant challenges, including security risks and user inconvenience. This study addresses these issues by developing an email spam classification system using tokenization and stemming techniques. Through experimentation with various machine learning algorithms, including Logistic Regression, SVM, Decision Trees, Random Forest, and KNN, we identified models with high accuracy and F1-Score, notably Logistic Regression and SVM. These models effectively distinguished between spam and legitimate emails, contributing to improved email security and user experience. Insights gained from analysis, such as word cloud visualization and top word associations, provided valuable guidance for enhancing spam detection algorithms. Overall, the developed system offers a reliable solution to mitigate the challenges posed by email spam.

KEYWORDS

Email spam, classification, tokenization, stemming, machine learning, Logistic Regression, SVM, Decision Trees, Random Forest, KNN, security, user experience

1 – INTRODUCTION

Email communication has become a vital part of our daily interactions, but the problem of email spam is causing issues for users. With the growing volume of spam email, it has become increasingly important to address this issue, as it brings risks like phishing attacks, malware distribution and overall inconveniences to users. This study aims to create a system for spotting spam emails using tokenization and stemming. Tokenization breaks down the words in emails, while stemming reduces them to their root forms. These methods should help the system better recognize spam by capturing language patterns. In our approach, we'll also explore different machine learning algorithms to find the best results. This way, we aim to show that our system is better at handling the challenges posed by email spam.

Problem Statement

1. Ubiquitous Email Communication: Email has transformed communication, but persistent spam poses security threats and disrupts user experience.
2. Critical Need for Detection System: Security risks, phishing attacks, and malware distribution necessitate an efficient spam detection system.
3. Enhancing User Experience: A reliable spam detection system ensures a streamlined inbox and a secure communication channel.

Importance of the Problem

Email spam jeopardises user security through phishing and malware threats. Our system, employing tokenization and stemming, strives to capture language patterns effectively to identify and mitigate these security risks.

Email spam disrupts user experience by inundating inboxes with unwanted content. Our approach aims to streamline this

experience by efficiently identifying spam, leading to a more organised and user-friendly email environment.

Challenges in Addressing the Problem

Dynamic language patterns in spam pose a challenge, requiring advanced text preprocessing. Tokenization and stemming address this challenge by adapting to evolving tactics, contributing to a robust security solution.

Nuanced language patterns demand sophisticated text preprocessing. Tokenization and stemming overcome this challenge, contributing to an improved email experience while addressing the evolving nature of spam tactics.

The dynamic nature of spam techniques demands a comprehensive approach. Tokenization and stemming contribute to overcoming the challenges posed by evolving language patterns, making our system adept at maintaining trust in email communication.

2 – OBJECTIVES

This study aims to develop an email spam classification system by leveraging advanced text preprocessing techniques and exploring various machine learning models. The objectives for this project are as follows:

1. Efficient Text Preprocessing: Implement tokenization and stemming to enhance the effectiveness of text preprocessing for email spam classification.
2. Count Vectorization Exploration: Investigate the efficacy of count vectorization as a feature extraction technique to represent textual data for spam classification accurately.
3. Model Performance Evaluation: Assess the performance of diverse machine learning models in classifying email spam accurately, considering metrics such as precision, recall, and F1-score.
4. Key Factors in Model Performance: Analyse the significant factors influencing the performance of the classification models, providing insights into effective spam detection. And explore the top-ranked words indicative of spam emails, contributing to a deeper understanding of spam characteristics and detection methods.

3 – RELATED WORKS

This section provides an overview of the various techniques and methods employed by researchers for email spam classification. Authors have utilized various methods, often testing with different datasets and models to address this challenge. In 2020, Neha Sattu conducted a study focusing on the application of seven prominent machine learning algorithms for spam email classification. The research addressed the evaluation of these algorithms, highlighting the split dataset approach as particularly effective in achieving superior classifier performance. Notably, support vector machines (SVM), random forests (RF), k-nearest neighbors (KNN), artificial neural networks (ANN), decision trees (DT), and logistic regression (LR) achieved an impressive accuracy of 99%, while Naïve Bayes (NB) achieved a slightly lower accuracy of 92%. The study employed preprocessing techniques such as tokenization, stop word removal, and n-grams for feature extraction (Neha Sattu, 2020). [1]

In 2010, T. Hamsapriya and D. Karthika Renuka proposed an effective spam detection technique focusing on addressing misspellings in spam emails. Their approach centred around word stemming to enhance content-based spam filters, aiming to improve classification accuracy. The study also introduced an email archiving solution as part of the broader spam control framework. Results from their research demonstrated a high accuracy rate of 96% in spam classification using the word stemming approach, showcasing its effectiveness in outperforming previous techniques (T. Hamsapriya & D. Karthika Renuka, 2010). [2]

In 2018, Kriti Agarwal and Tarun Kumar presented an integrated approach to email spam detection, combining the Naïve Bayes (NB) algorithm with Particle Swarm Optimization (PSO). The study aimed to mitigate the impact of excessive email spam on user experience by filtering spam emails more effectively. By utilizing preprocessing techniques such as tokenization, stop word removal, and n-grams for feature extraction, coupled with the NB algorithm optimized by PSO, the integrated approach yielded promising results. Particularly, PSO showcased superior

performance metrics compared to individual NB approaches, highlighting its potential in enhancing spam detection accuracy (Kriti Agarwal & Tarun Kumar, 2018). [3]

4 – TECHNICAL BACKGROUND

Email spam, also known as unsolicited or unwanted email, has become a pervasive issue in modern communication systems. The sheer volume of spam emails poses a significant challenge for users and organisations alike. Beyond the inconvenience caused by sorting through a cluttered inbox, email spam brings forth more serious risks. Threats such as phishing attacks, where malicious actors attempt to obtain sensitive information, and the distribution of malware through seemingly innocuous emails have become prevalent. Consequently, addressing the problem of email spam has become imperative to ensure the security and efficiency of electronic communication.

Tokenization and stemming are fundamental techniques in natural language processing (NLP) that play a crucial role in the analysis of textual data. Tokenization involves breaking down a piece of text into individual words or tokens. This process enables the system to understand the structure of the text and identify key elements. Stemming, on the other hand, involves reducing words to their base or root forms. By simplifying words to their core components, stemming helps capture essential semantic meaning and eliminates variations that may hinder effective analysis.

5 – METHODOLOGY

Data Collection and Preprocessing

Selection of Labelled Dataset: To conduct our research, we selected a labelled dataset from Kaggle (Retrieved from https://www.kaggle.com/datasets/mfaisalqureshi/spam-email) that includes a diverse range of emails marked as either spam or legitimate. This dataset serves as the foundation for training and evaluating our spam classification system. The diversity in the dataset ensures the robustness of our model against different types of spam and non-spam emails. Though, the author did not exactly give the exact distribution of types of emails, such as formal or informal emails.

Data Exploration and Understanding: Prior to model development, we performed an in-depth exploration of the dataset to understand its characteristics. This involved examining the distribution of spam and non-spam emails, identifying common patterns, and gaining insights into the challenges posed by the data.

Preprocessing Steps: The preparation phase involved cleaning the data by removing unwanted characters, irrelevant information, and any anomalies that could adversely affect the performance of our classification system. This step is critical for ensuring that the model is trained on clean and meaningful data.

Feature Extraction: Our feature extraction process was executed through the application of count vectorization, converting the raw text data of emails into a format conducive to machine learning algorithms. The implementation encompassed the following steps:

Tokenization: Having obtained the email dataset, the initial step involved breaking down the textual content of each email into individual words, a process known as tokenization. This step was crucial in dissecting the linguistic elements within the emails, allowing us to comprehend the structural composition of the textual data.

Stemming: Stemming was applied to reduce words to their root forms. This step aims to overcome variations in language and focuses on capturing the essential meaning of words. By standardising the text, stemming contributes to the effectiveness of our spam classification system.

Counting: With tokenization complete, our algorithm diligently counted the occurrence of each unique word (token) within every email in the dataset. This counting process yielded a comprehensive understanding of the frequency distribution of words, capturing the distinctive language patterns inherent in each document.

Feature Extraction

The culmination of count vectorization resulted in a matrix where each row represented an individual email, and each column denoted a unique word found across the entire collection of emails. The numerical values within this matrix signified the frequency of each word within the respective documents. This

matrix representation transformed the diverse textual content into a structured, numerical format, laying the groundwork for subsequent machine learning analysis.

By adopting count vectorization, we effectively translated the inherent complexity of email content into a quantifiable feature matrix. This matrix serves as a fundamental input for our machine learning algorithms, enabling them to discern patterns, identify relevant features, and make informed predictions based on the frequency distribution of words in the dataset.

Model Deployment

In our pursuit of finding the most effective algorithm for spam email detection, we tested a diverse set of machine learning methods, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees, Random Forest, and Logistic Regression. The selection of these algorithms was driven by distinct characteristics and advantages they offer. SVM's proficiency in high-dimensional spaces, KNN's simplicity and intuitive proximity-based classification, Decision Trees' interpretability, Random Forest's ensemble approach for mitigating overfitting, and Logistic Regression's suitability for binary classification tasks were key considerations. This enables us to identify the most effective solution tailored to the unique challenges of spam email detection.

Model Results Comparison

To evaluate the performance of each algorithm, we utilised key metrics, including precision, recall, F1 score, and support. These metrics provide a comprehensive understanding of how well the models distinguish between spam and non-spam emails.

Precision: Precision measures the accuracy of positive predictions, indicating the proportion of correctly predicted spam emails among all emails predicted as spam. A higher precision score signifies fewer false positives.

Recall: Recall, also known as sensitivity, gauges the ability of the model to capture all actual positive instances. It represents the proportion of correctly predicted spam emails among all actual spam emails. A higher recall score indicates fewer false negatives.

F1 Score: The F1 score is the harmonic mean of precision and recall, providing a balanced evaluation of a model's performance. It is particularly useful when there is an imbalance between positive and negative instances.

Support: Support represents the number of occurrences of each class in the actual data. It helps assess the reliability of the model's predictions by providing context on the distribution of spam and non-spam emails.

In the subsequent sections, we will present the detailed results and analysis of each algorithm, comparing their performance in terms of these metrics and drawing conclusions about the effectiveness of our email spam classification system.

6 – IMPLEMENTATION & RESULTS

Data Split

For the implementation of our Email Spam Classification, the pre-processed dataset was partitioned into training and testing sets using an 80-20 split. Specifically, 80% of the data was allocated for training the models, while the remaining 20% was reserved for evaluating their performance. The split was performed randomly to maintain the representativeness of the data in both the training and testing sets.

Model Results

We then deployed the five machine learning algorithms (Logistic Regression, SVM, Decision Tree, Random Forest, and KNN). The evaluation of these models is based on their accuracy (Figure 1.0) and classification reports (Figure 2.0) which include precision, recall, f1-score and support, where we further focused on two key metrics: Accuracy and F1-Score.

Model	Accuracy
SVM	0.979
KNN	0.978
Decision Tree	0.97
Random Forest	0.969
Logistic Regression	0.935

Figure 1.0 (Model Accuracy)

Model	Class	Precision	Recall	F1-Score	Support
SVM	1 (ham)	0.98	1.00	0.99	975
	0 (spam)	0.98	0.85	0.91	140
	W. Avg	0.98	0.98	0.98	1115
KNN	1 (ham)	0.93	1.00	0.96	975
	0 (spam)	1.00	0.49	0.65	140
	W. Avg	0.94	0.94	0.93	1115
Decision Tree	1 (ham)	0.98	0.99	0.98	975
	0 (spam)	0.91	0.83	0.87	140
	W. Avg	0.97	0.97	0.97	1115
Random Forest	1 (ham)	0.97	1.00	0.98	975
	0 (spam)	0.98	0.80	0.88	140
	W. Avg	0.97	0.97	0.97	1115
Logistic Regression	1 (ham)	0.98	0.99	0.99	975
	0 (spam)	0.96	0.87	0.91	140
	W. Avg	0.98	0.98	0.98	1115

Figure 2.0 (Model Classification Report)

Model Comparison

Logistic Regression and SVM performed exceptionally well with the highest accuracy of approximately 97.94% and F1-Score of 0.98. Random Forest and Decision Tree also demonstrated strong performance, achieving accuracies of 97.13% and 96.95%, with F1-Scores of 0.97. KNN, while having a lower accuracy at 93.54%, still provided a respectable F1-Score of 0.93.

Overall, the model comparison using the accuracy (*Figure 3.0*) and classification reports (*Figure 4.0*) shows that the Logistic Regression and SVM models outperformed the others in terms of both accuracy and F1-Score. These results indicate their effectiveness in classifying email spam. However, the choice of the best model may also depend on other factors, such as computational efficiency and interpretability, which should be considered in practical applications.

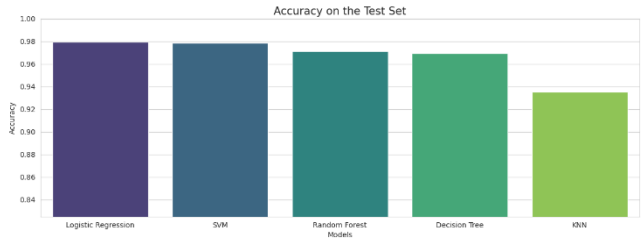

Figure 3.0 (Model Accuracy on the Test Set)

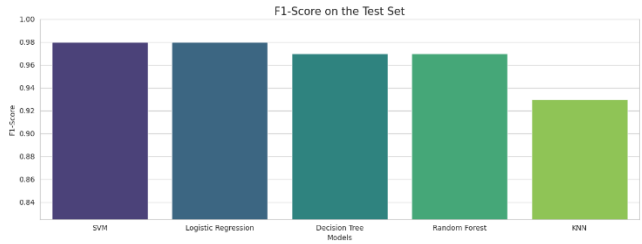

Figure 4.0 (Model F1-Score on the Test Set)

Test Example (Logistic Regression Model)

In the test example using the Logistic Regression model, two sample instances were provided for classification: a ham example and a spam example.

- The ham example: "Hi there, just checking in to see how you're doing. Let's catch up soon for coffee,",
- the spam example: "Get rich quick! Amazing offer, win a million dollars in just one click. Claim your prize now!"

The ham example was correctly identified as legitimate (non-spam) content. Similarly, the spam example was also accurately classified as spam. This successful prediction underscores the robustness of the Logistic Regression model in discerning between legitimate and spam emails, demonstrating its efficacy in making accurate classifications on real-world examples. The model's ability to correctly identify the nature of both ham and spam instances is a promising indication of its reliability in

practical scenarios, reinforcing its suitability for email spam classification.

6 – ANALYSIS & DISCUSSION

Data Observations

As part of our exploratory data analysis, we used Principal Component Analysis (PCA) for dimensionality reduction and created scatter plots to visualise the distribution of both training (*Figure 1.0*) and testing data (*Figure 3.0*).

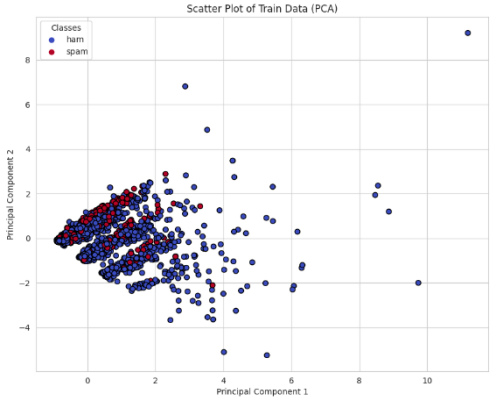

Figure 5.0 (Training Set Scatter Plot)

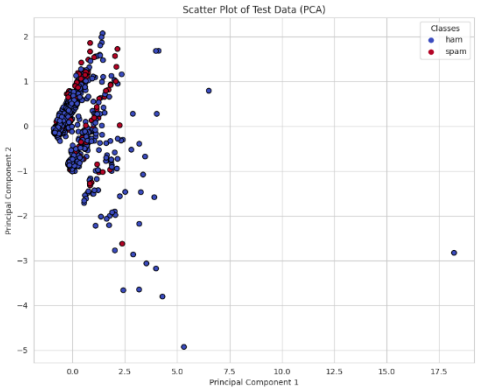

Figure 6.0 (Training Set Scatter Plot)

The scatter plots generated through PCA for both the training and testing data revealed a notable observation – the clustering of ham (non-spam) and spam emails in close proximity. The visual clustering may stem from inherent similarities in the textual features of ham and spam emails, posing a challenge for the classification task. It highlights the importance of feature engineering or exploring more sophisticated techniques and algorithms to capture nuanced distinctions between the two classes.

KNN Decision Boundaries Visualization and Analysis

We focused on examining the decision boundaries of the K-Nearest Neighbours (KNN) model, as KNN was the worst performing model compared to the other 5 models. We made a visualisation through a scatter plot with the test set. This exploration provided insights into how well the KNN algorithm differentiated between ham and spam emails.

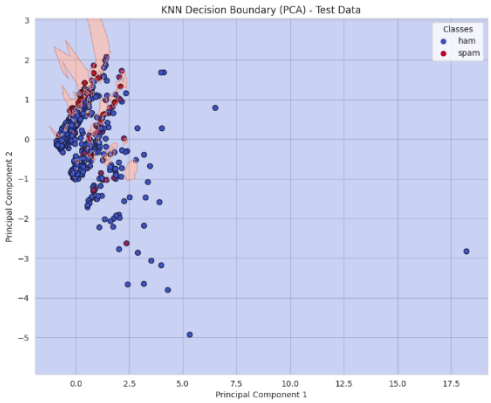

Figure 7.0 (Training Set Scatter Plot)

The observed difficulties align with our previous findings from the scatter plot visualisation using PCA, where ham and spam emails exhibited proximity, making it challenging for the model to establish clear boundaries. The limitations of the KNN algorithm in handling clustered data are evident, and this may impact its performance, especially when dealing with complex patterns. Addressing the clustering challenge may involve exploring additional features or employing advanced techniques to enhance the model's ability to discern subtle differences.

Word Cloud Analysis for Spam Emails (Logistic Regression)

In the pursuit of gaining insights into the distinguishing features associated with spam emails, a word cloud was generated based on the predictions of the Logistic Regression model (*Figure 8.0*), which exhibited high performance.

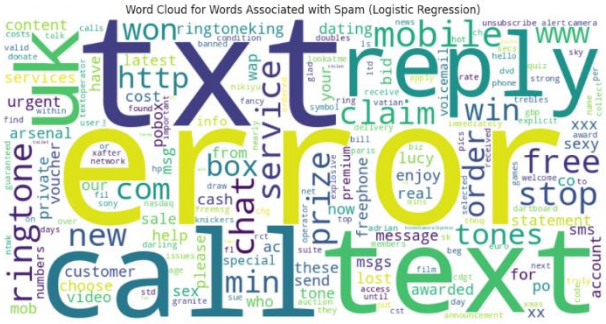

Figure 8.0 (Word Cloud for Words Associated with Spam)

We then did further analysis of the top 20 words associated with spam emails (*Figure 9.0*), as determined by the Logistic Regression model's coefficients, providing valuable insights into the linguistic patterns that strongly contribute to the identification of spam. Here is a brief analysis of the key findings.

Top 10 words			**Top 11-20 words**	
Feature	**Coeff**		**Feature**	**Coeff**
error	2.506		service	1.355
txt	2.279		free	1.282
call	1.862		claim	1.243
text	1.848		new	1.235
reply	1.728		prize	1.174
uk	1.671		win	1.119
mobile	1.573		won	1.108
ringtone	1.393		com	1.089
stop	1.364		min	1.065
chat	1.357		tones	1.026

Figure 9.0 (Top 20 Words Associated with Spam)

The analysis of the top words associated with spam emails reveals key linguistic features that significantly contribute to the identification of spam content. Terms such as "error" and "txt" are strategically employed to evoke attention and urgency, suggesting potential issues that require immediate action and emphasising the prevalence of text-based communication in spam. Additionally, words like "call," "reply," and "chat" underscore the common call-to-action and engagement strategies employed by spammers. Notably, geographic targeting is hinted at with the inclusion of "uk," indicating a potential focus on messages tailored to or originating from the United Kingdom. The presence of terms like "free," "prize," and "win" reinforces the common tactic of enticing recipients with promises of rewards or benefits.

Overall, this linguistic analysis provides valuable insights for refining spam detection algorithms, enabling more accurate email classification systems that can effectively distinguish between spam and legitimate messages.

7 – CONCLUSION

In conclusion, this research project has successfully developed an email spam classification system by leveraging tokenization and stemming techniques. Extensive experimentation with a variety of machine learning algorithms, such as Logistic Regression, SVM, Decision Trees, Random Forest, and KNN, allowed us to identify models with notable accuracy and F1-Score, particularly highlighting the effectiveness of Logistic Regression and SVM. These models exhibited robust performance in distinguishing between spam and legitimate emails. Valuable insights gained from analysis, including word cloud visualization and top word associations, have provided significant guidance for enhancing spam detection algorithms. Ultimately, the developed system represents a crucial step towards mitigating the challenges posed by email spam, thus ensuring a safer and more efficient email communication environment for users.

7 – REFERENCE

[1] Sattu, N. 2020. A study of machine learning algorithms on email spam classification. Order No. 27993789. ProQuest Dissertations & Theses Global, 2435855505. Retrieved from http://210.48.222.80/proxy.pac/dissertations-theses/study-machine-learning-algorithms-on-email-spam/docview/2435855505/se-2.

[2] D. Karthika Renuka and Dr.T. Hamsapriya. 2010. Email classification for Spam Detection Using Word Stemming. International Journal of Computer Applications 1, 5 (February 2010), 58–60. DOI:https://dx.doi.org/10.5120/125-241

[3] Agarwal, K. and Kumar, T. 2018. Email Spam Detection Using Integrated Approach of Naïve Bayes and Particle Swarm Optimization. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 685-690. DOI:https://doi.org/10.1109/ICCONS.2018.8662957