Enhanced Naive Bayes Classifier for Email Spam Filtering Using the Product Rule of Logarithm

Cyrille Anne Chuajap [1], Diether Jay Domanog [1], Mark Christopher Blanco [1], Dan Michael Cortez [1]

[1]*Student, Computer Science Department, College of Engineering and Technology, Pamantasan ng Lungsod ng Maynila, University of the City of Manila, Intramuros, Manila 1002, Philippines.*

Corresponding Author: djldomanog2018@plm.edu.ph

Abstract: - The Naive Bayes algorithm is a widely used classification algorithm with applications in various domains. The prevalence of email spam poses a significant challenge to effective email communication every day. The study proposed an enhanced Naïve Bayes classifier for email spam filtering, utilizing the product rule of logarithm to overcome the issue of probability calculations. The methodology involves data preprocessing, application of log probability calculation, and training model classier using the enhanced algorithm. Provided a testing dataset that was used to evaluate results and demonstrate improved classification accuracy in the implemented filtering model. The proposed enhancement contributes to the robustness of the Naive Bayes algorithm in various classification tasks.

Key Words: - ***Naive Bayes, Spam Filtering, Product Rule of Logarithm, Text Classification, Normalization, Likelihood, Probability, Underflow, Limitation.***

I. Introduction

Spam messages have become increasingly prevalent in today's social media era, manifesting in various forms such as text messages, emails, and chat messages. Orred (2022) highlights that spam texts surged to a staggering 28% in March 2022, with the United States alone receiving around 300 million spam messages daily. Inocencio (2022) reports instances where individuals received messages from unknown numbers containing their full names, further underscoring the potential dangers of spam. Detecting spam messages can be challenging, and having a reliable spam filtering system can greatly assist users in identifying potential threats in emails and messages.

Manuscript revised May 27, 2023; accepted May 28, 2023. Date of publication May 30, 2023.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

Among the various approaches used for spam filtering, the Naive Bayes algorithm has gained widespread popularity due to its strong probabilistic foundations. The Naive Bayes algorithm has found significant traction in commercial and open-source spam filters, as emphasized by Rusland et al. (2017).

The issue of precision limitations and the underflow in the multiplication of small probabilities presents a pressing problem that needs to be addressed. An additional concern, as noted by Chauhan (2022). Tokuc (2022) highlighted that "when multiplying two small values between 0 and 1, the result tends to be even smaller. Consequently, the precision limitations of floating-point arithmetic in computer systems can hinder the accurate representation of these extremely small numbers." This limitation poses a significant challenge in the Naïve Bayes approach, potentially impacting the algorithm's overall accuracy and reliability.

This study aims to enhance the Naïve Bayes Algorithm to improve its accuracy in classifying emails as either spam or non-spam messages. The proposed enhancements encompass various aspects, including data pre-processing, incorporating logarithms into the Bayes Formula, model training, and rigorous testing to showcase enhanced text classification accuracy. This study seeks to demonstrate a more effective and accurate approach to spam email classification by implementing these enhancements.

II. Related Studies

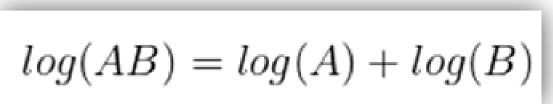
Spam detection's most effective method is utilizing some form of machine learning. One of the machine learning spam filtration methods includes Naive Bayes. The naive Bayes spam detection method is a supervised machine learning probabilistic model for spam classification based on Bayes Theorem. Naive Bayes was chosen for its speed, multi-class prediction ability, and small training set. Also, Naive Bayes is the baseline for most spam filters. Improving the Naive Bayes will inevitably improve most spam filters overall (Peng, 2018). The Naive Bayes method is used to lessen the inaccuracy of the result when there is missing data due to its insensitivity to missing data and efficiency in calculation. Naive Bayes could be improved by using the logarithmic form. "The treatment of missing data using deletion instances, median imputation and mean imputation were conducted to determine the effect on classification accuracy" (Syafie, 2018).

Dada et al. (2019). The naive Bayes classifier is desirable for spam filtering due to its simplicity and ease of implementation compared to other conditional models. It is used in solving problems that involve two or more classes. Naive Bayes are effective when it comes to managing discrete and continuous data. Sumithra et al. (2022), many people abuse emails by sending unwanted and pointless messages for their benefit, like containing virus links that direct users to fraudulent websites. These unwanted emails caused problems for the average user, such as filling the inbox with undesirable emails. As a result, these unwanted emails make it difficult for the user to find valuable emails. Due to this, a powerful email spam detector is required to filter a large number of spam emails with increased accuracy while ensuring that real emails are not screened as spam. According to Tajalizadeh et al. (2019), the behaviour of members can be monitored to identify the source of spam besides investigating the message content. For example, if a member sends messages exceeding his number of connected friends, all of his messages can be categorized as spam. Some spammers deceive the investigators by randomly sending a limited number of spam to members or using fake trending hashtags to make spam more visible in searches. Spammers spread harmful information, participate in disruptive behaviour, and harass others. "In our work, we used machine learning classifiers, for instance, Multinomial-Naive-Bayes algorithm, Support-Vector-Machine models (SVM), Logistic-Regression, and Decision Trees to detect SMS spam from a dataset of nearly six thousand messages, taking into account combinations of Term-Frequency-Inverse-Document-Frequency (TF-IDF) and Count-Vectorization features to investigate the trade-off between F1-score, accuracy -score, and computing time" (Singh, 2022). "Spam filtering is one of the most important applications in email services that have become increasingly sophisticated due to the enormous usage of the Internet. Spam filters have traditionally been implemented on the CPU with a pattern-matching algorithm" (Kalubandi, 2017).

III. Research Methodology

Application of Product Rule of Logarithm to the Bayes Rule

Tokuc (2022), the product of two small values between 0 and 1, tends to result in an even smaller number. This can pose challenges in terms of floating-point precision and the limitations of computer systems when working with very small numbers. To address this issue, the study proposes applying the product rule of logarithm. By leveraging logarithmic calculations, the product of probabilities can be transformed into a sum of logarithms, which mitigates the risk of numerical underflow and allows for more accurate computations. Incorporating the product rule of logarithm in the Naïve Bayes algorithm offers a practical solution to overcome the limitations associated with working with extremely small probabilities.

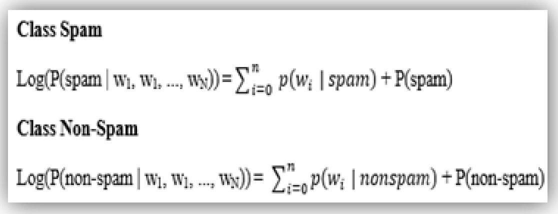


$$log(AB) = log(A) + log(B)$$

Figure 1: Product Rule of Logarithm

Smith (2022) emphasizes that the logarithm of a product can be expressed as the sum of logarithms, as depicted in Figure 1. This mathematical property, known as the product rule of logarithm, becomes particularly relevant when dealing with small probabilities. By applying the product rule of logarithm, the need to multiply probabilities together, which can result in extremely small values, is circumvented.

Instead, the process is simplified by taking the logarithm of each probability and then summing them. Consequently, the logarithm of the product of probabilities, Log(A * B), is equivalent to the sum of the logarithms of A and B, i.e., Log(A) + Log(B). This simplification not only avoids potential issues related to multiplying small numbers but also enhances the computational efficiency of the algorithm.



Class Spam

$\text{Log}(P(\text{spam} \mid w_1, w_1, \ldots, w_N)) = \sum_{i=0}^{n} p(w_i \mid spam) + P(\text{spam})$

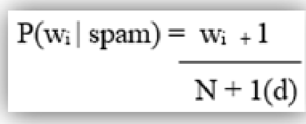
Class Non-Spam

$\text{Log}(P(\text{non-spam} \mid w_1, w_1, \ldots, w_N)) = \sum_{i=0}^{n} p(w_i \mid nonspam) + P(\text{non-spam})$

Figure 2: Application of Product Rule of Logarithm to Bayes' Rule

Figure 2 shows Bayes' formula applying the product rule of logarithm. The new modified formula for computing the probability of spam and non-spam now involves only log function and addition operation without multiplication operation. The computation of likelihood was also simplified using the summation function.

Computing a word's probability outside the dataset results in zero values and leads to inaccurate conclusions. Tokuc (2022) states that "the smoothing parameter ensures that the probability value is never zero. Applying a smoothing technique assigns a very small probability estimate to such zero frequency occurrences, hence, regularising the Naive Bayes classifier".



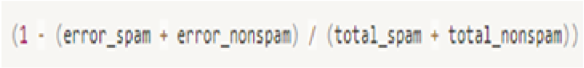
$$P(w_i \mid \text{spam}) = \frac{w_i + 1}{N + 1(d)}$$

Figure 3: Smoothing Formula

The smoothing application ensures that the equation will not produce a probability of zero. The smoothing technique involves regularizing the likelihood of each word (Wi) by adding a non-negative smoothing value which is 1, and dividing it by the sum of the total number of words in a particular class and the product of smoothing value 1 and the total number of unique words in that class (d), as illustrated in figure 3.

The log probability computation utilizes statistical techniques to calculate the likelihood of specific tokens or combinations of tokens occurring in the spam or non-spam emails. Smoothing techniques handle unseen tokens by assigning small probabilities to ensure robustness in classification. The email classification filter employs the calculated probabilities to determine whether an email is a likely spam or non-spam based on comparing the respective probabilities. These processes collectively contribute to establishing a well-trained model for the study.

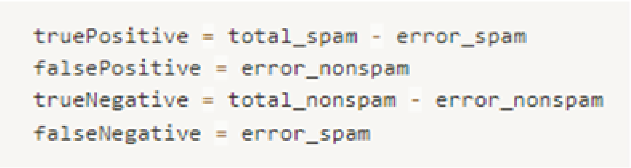
To get the accuracy of the overall accuracy of the spam filter, divide the sum of 'error_spam' and 'error_nonspam' by the sum of the 'total_spam' and 'total_nonspam'. Then, subtract the quotient by error rate value 1 to get the accuracy of the spam filter that represents the correct classified emails in percentage, as shown in the equation in Figure 4.



```
(1 - (error_spam + error_nonspam) / (total_spam + total_nonspam))
```

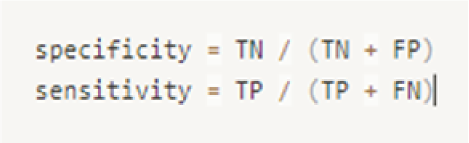
Figure 4: Accuracy Formula

The confusion matrix gives a better picture of the algorithm's performance (Ferreira, 2018). The confusion matrix calculates the following values: true positives (correctly predicted positives), true negatives (correctly predicted negatives), false positives (incorrectly predicted positives), and false negatives (incorrectly predicted negatives). To calculate the test result's values, specificity, and sensitivity, refer to Figure 5 below. Solving sensitivity or the true positive rate and specificity or true negative rate refer to the formula shown in Figure 6.



```
truePositive = total_spam - error_spam
falsePositive = error_nonspam
trueNegative = total_nonspam - error_nonspam
falseNegative = error_spam
```

Figure 5: Confusion Matrix Values



```
specificity = TN / (TN + FP)
sensitivity = TP / (TP + FN)
```

Figure 6: Specificity and Sensitivity

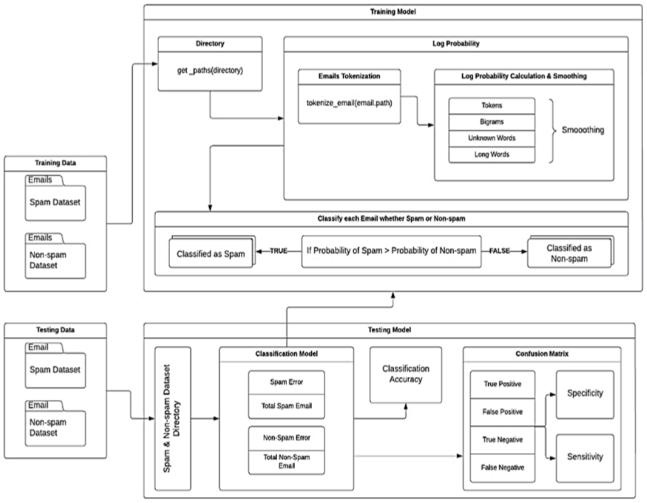


Figure 7: Conceptual Framework

3.1 Conceptual Framework

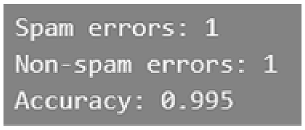
The conceptual framework illustrated in Figure 7 helps to visualize how the data flows, how log probabilities are implemented in the enhancement of the naïve Bayes algorithm, and how the newly enhanced naïve Bayes algorithm performs.

The training data, as well as the testing data, involves the preprocess method called tokenization. According to Chakravarthy (2021), "The tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words". The tokenization in the study implemented converting the text to lowercase, replacing certain symbols and punctuation characters with spaces, and splitting the lines into individual words. This process will help the training model to calculate probabilities and classify spam or non-spam email. The log probability calculation and smoothing class in Figure 7 utilizes the probability calculations during the training model process that will be used to classify a certain email as spam or non-spam.

As Lohner (2019) explains, "The higher the probability number or percentage of an event, the more likely it is that the event will occur." In other words, if the calculated probabilities of an email being classified as spam are higher than the probabilities of it being classified as non-spam, the email is classified as a spam email. This decision is made by comparing the probability values obtained from the trained spam and non-spam models. By leveraging these probabilities, the spam filter can effectively determine the likelihood of an email being spam or non-spam, enabling accurate classification and filtering of incoming messages.

IV. Result And Discussion

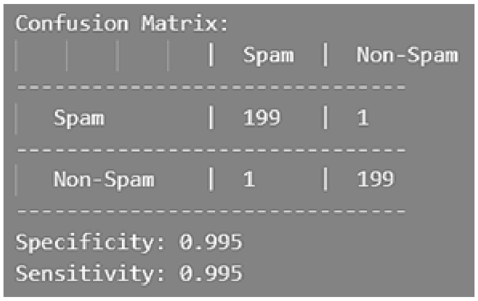
In the study's methodology, establish a training model that will classify a set of testing data. Analyze the results and performance discussion of the spam filter. The performance of the testing data for spam and non-spam emails to the trained model produces an error 1 for both spam and non-spam, shown in Figure 8. The spam filter correctly classified 199 out of 200 spam emails. For non-spam emails, the spam filter correctly classified 199 out of 200 non-spam emails. The overall accuracy of the spam filter, considering both spam and non-spam emails, is 0.995 or 99.5%. This demonstrates the effectiveness of the filter in accurately classifying emails.



```
Spam errors: 1
Non-spam errors: 1
Accuracy: 0.995
```

Figure 8: Performance Rate

Based on the confusion matrix, the filter's specificity (true negative rate) is 0.995 or 99.5%, indicating the proportion of correctly identified non-spam emails. The sensitivity (true positive rate) is 0.99.5 or 99.5%, representing the proportion of correctly identified spam emails.



```
Confusion Matrix:
|   |   |   |   |  Spam  |  Non-Spam
------------------------------
|  Spam        |  199   |  1
------------------------------
|  Non-Spam    |  1     |  199
------------------------------
Specificity: 0.995
Sensitivity: 0.995
```

Figure 9: Confusion Matrix

The spam filter demonstrates several strengths, including achieving a high overall accuracy of 99.5%. It effectively identifies spam and non-spam emails, reducing unwanted emails and minimizing the risk of important emails being classified as spam. However, there are areas where the implemented spam filter might struggle since the newly enhanced Naïve Bayes algorithm only addresses the underflow issue that can occur when calculating probabilities. For instance, unbalanced or limited training data can hinder the filter's ability to accurately classify underrepresented classes. To address this, it is crucial to ensure a diverse and representative training dataset, which suggests that data collection efforts and data augmentation techniques will be necessary to improve generalization and cover a wide range of spam variations techniques.

V. Conclusion

The main research objective of this study was to enhance the Naive Bayes algorithm by implementing the product rule of logarithm and evaluate its performance in classifying spam and non-spam emails. Suresh (2021) notes that a good model is characterized by high True Positive and True Negative rates, along with low False Positive and False Negative rates. To achieve this objective, the research methodology involved preprocessing the data and implementing the product rule of logarithm, which contributed to the effectiveness of the training model development. The study's results demonstrated promising accuracy rates, indicating the effectiveness of the implemented filter model. Overall, the findings highlight the successful enhancement of the Naive Bayes algorithm and its potential for accurately classifying spam and non-spam emails.

In conclusion, implementing the product rule of logarithm in the Naive Bayes algorithm effectively addresses the issue of underflow, which can occur when dealing with extremely small probabilities. The algorithm can handle calculations more accurately by taking the logarithm of probabilities and performing addition instead of multiplication. However, it is important to note that while this implementation resolves one challenge, there may still be other challenges that the algorithm faces. These challenges can be addressed through future improvements and ongoing research. Continuous monitoring, adaptation, and research efforts are essential to overcome these challenges.

However, there are limitations of the spam filter, such as dealing with deceptive or evolving spam techniques, highly targeted or personalized spam, and unbalanced or limited training data is inevitable in the context of spam filtering. These challenges are normal to the nature of spam and the evolving tactics employed by spammers. To address those challenges, it is recommended for continuous monitoring, and research efforts are necessary to stay up to date with new spam patterns and adapt the filter accordingly. This implies that the filter must evolve alongside spam techniques to maintain its effectiveness.

References

[1]. Orred, K. (2022, June 2). 2022 Spam Text Statistics: Are Spam Texts on the Rise?

[2]. Inocencio, S. (20220). PNP bares ways scammers get the personal info of spam text recipients. CNN Philippines.

[3]. Rusland, N. F., Wahid, N., Kasim, S., & Hafit, H. (2017). Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets. IOP Conference Series, 226, 012091.

[4]. Borisov, O. (2021, December 16). Improved Naïve Bayes Classifier to Solve Text Classification Problems. Medium.

[5]. Tokuç, A. A. (2022, November 11). How to Improve Naive Bayes Classification Performance? | Baeldung on Computer Science. Baeldung on Computer Science.

[6]. Chauhan, N. S. (2022, November 15). Naïve Bayes Algorithm — Everything you need to know. AI Planet, Formerly DPhi.

[7]. Sumithra, A., Ashifa, A., Harini, S., Kumaresan, N. (2022). Probability-based Naïve Bayes Algorithm for Email Spam Classification. 2022 International Conference on Computer Communication and Informatics (ICCCI).

[8]. Singh, T., Kumar, T., Shambharkar, P. (2022). Enhancing Spam Detection on SMS performance using several Machine Learning Classification Models. 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI).

[9]. Tajalizadeh, H., Boostani, R. (2019). A Novel Stream Clustering Framework for Spam Detection in Twitter. IEEE Transactions on Computational Social Systems, 6(3).

[10]. Peng, W., Huang, L., Jia, J., Ingram, E. (2018). Enhancing the Naive Bayes Spam Filter Through Intelligent Text Modification Detection. 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications.

[11]. Syafie, L., Umar, F., Mude, A., Darwis, H., Herman, Harlinda. (2018). Missing Data Handling Using the Naive Bayes Logarithm (NBL) Formula. 2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT).

[12]. Dada, E., Bassi, J., Chiroma, H., Abdulhamid, S., Adetunmbi, A., Ajibuwa, O. (2019). Machine learning for

email spam filtering: review, approaches and open research problems. Heliyon 2019 Jun; 5(6).

[13]. Peng, W., Huang, L., Jia, J., Ingram, E. (2018). Enhancing the Naive Bayes Spam Filter Through Intelligent Text Modification Detection. 12th IEEE International Conference on Big Data Science and Engineering (Trust Com/Big Data SE).

[14]. Kalubandi, V., Varalakshmi, M. (2017). Accelerated spam filtering with enhanced KMP algorithm on GPU. 2017 National Conference on Parallel Computing Technologies (PARCOMPTECH).

[15]. Soni, D. (2019, July 16). Introduction to Naive Bayes Classification - Towards Data Science. Medium.

[16]. Bhagat, V. (2021, October). Naive Bayes Algorithm. Topcoder.

[17]. Smith, C. (2022, January 7). The Logarithm of a Sum - Chris Smith - Medium. Medium.

[18]. Chakravarthy, S. (2021, December 14). Tokenization for Natural Language Processing - Towards Data Science. Medium.

[19]. Lohner, S. B. (2019, October 17). What Are the Chances? Scientific American.

[20]. Ferreira, H. (2018, April 4). Confusion matrix and other metrics in machine learning. Medium.

[21]. Suresh, A. (2021, December 16). What is a confusion matrix? - Analytics Vidhya - Medium. Medium.

[22]. Reka. (2020, June). Overflow and underflow. Machine Learn IT.

[23]. Yıldırım, S. (2020, February). Naive Bayes Classifier — Explained. Towards Data Science.