

A New Ensemble Model Based on Machine Learning Algorithms for the Spam-Filtering

Zixuan Lin*

Electric and electronic engineering, University of Leeds, Leeds, United Kingdom

*Corresponding author: el22z2l@leeds.ac.uk

Abstract. Due to the rapid increasement about the information in email, it is necessary to improve the technology of spam-filter. Based on the data on website, this study used python and Scikit-Learn library to process the data. Training the data by using Logistic Regression, Support Vector Machine, Naïve Bayes, Random Forest, and Decision Tree these five models, and calculate the evaluation measures for the rest test. From the results of the evaluation to infer the possible logical formula between these five models and calculate their evaluation measures. Comparing the final evaluation measures from the single model's and the logical formula's to find one possible logical formula that is better than the single model. The experimental results demonstrated that, each models have different efficiencies about the spam-filter, and the logical formula will improve some scores in evaluation measures, which means the logical formula using multiple models will improve the technology of spam-filter.

Keywords: Machine Learning, Scikit-Learn, Spam-filter.

1. Introduction

In contemporary times, electronic mail (email) serves as a widely accepted medium of communication and information exchange, encompassing the transmission of offers, payments, and other such correspondence. The first spam email was in 1978 when one user of ARPANET received an ad unprovoked. From the data showed in Statista, almost 85% of all emails are spam emails. In 2022, there are about 333.2 billion emails on average every day, which means about 283.22 billion spam emails on average have been sent to people every day. The number in 2023 will increase to 295.2 billion, and it will exceed 300 billion in the next year by forecast [1]. The significant volume of spam emails presents a concerning issue as it consumes valuable time, and more significantly, deceives individuals by eliciting money or compromising their privacy.

During the late 1990s, Internet Service Providers (ISPs) take measures to avoid cluttering their customers' mailboxes with spam emails. Subsequently, from the mid to late 2000s, additional measures were introduced to mitigate the reception of spam emails. To date, individuals continue to employ various strategies to enhance their spam filtering capabilities. In research conducted by Kaur et al., they found numerous spam detection techniques, 9 non-machine learning modules and 9 machine learning modules, but all of these techniques can never get 100% accuracy [2]. The field of machine learning, which is currently experiencing rapid growth, has emerged as a significant trend in modern technology, demonstrating the potential for computers to enhance their performance automatically through training [3]. For instance, Nurul Fitriah Rusland et al. had analysis the naïve bayes module performed well when using SPAMBASE dataset [4]. Dave and Harry had found that Random Forest module is good after clustering based on the active learning [5]. Ola and Nizar use SVM for spam filtering and Olusanya Olatunji also use SVM to improve spam detection [6, 7]. Alghoul Ahmed et al had used neural network for email classification [8]. Bilge Kagan et al based on logistic regression module to train an artificial bee colony algorithm for spam filtering [9]. As the machine learning example showed above, most of time the spam emails are cluttered by only one single module or using different modules to find the best module for spam filtering. Only a few numbers of spam filters will use two modules. Like Daisy, S. Jancy Sickory, and A. Rijuvana Begum, who used naïve bayes and MRF module to build a mail spam filtering [10].

And this study will be based on multiple modules to improve spam filtering. The study will consider using Logistic Regression (LR), Support Vector Machines (SVM), random forest, neural network and naïve bayes these five modules to train the dataset, using python to get each prediction and calculate the evaluation measures. Based on the evaluation measures of different modules to find a new prediction using logical function to combine these predictions.

This project is aimed to find one best logical formula between the five modules. Using python to train the data and get the predictions of each module. Based on the evaluation measures, guessing some possible logical formula and getting the new predictions. Then calculate the evaluation measures through these new predictions and compare them with the original measures. There are four possible relationships in this project, but only one has the best measures. Which means it truly has one possible logical formula that has a nice prediction, and maybe in the future, the formula can be related with more than five modules.

2. Method

2.1. Dataset Description and Preprocessing

The UCI website provides the dataset, which is a machine learning repository [11]. In order to use the module for training the data, the data will be first transferred to the mode that can be processed by models. The Bag of Word (BoW), a natural language processing technique of text modelling, will transfer words into the number of the frequencies of each word. The BoW is a technique which is used for classification [12], and it is conducive to transferring a messy, unstructured textual dataset into a structured, well-fixed numerical dataset [13]. In this project, BoW will help to transfer the words into its equivalent vector of the numbers.

The Bow (Bag of Words) is a widely used natural language processing technique that has been incorporated into the Scikit-Learn library. And the next part is to separate the normal email and spam email with 0 and 1, the spam email will be noted as 1 and normal email will be noted as 0. Subsequently, the Bag of Words (BoW) technique is employed to refine the dataset. The parameter "stop_words = 'english'" is implemented to exclude fundamental and ubiquitous words from the dataset. Moreover, the training and test data are separated into x_train and x_test matrices. The visualization presented in graph 4 illustrates the composition of the dataset, which consists of roughly 1,400 spam emails and over 4,100 non-spam emails.

2.2. Proposed Approach

The training modules are also included in the library mentioned above. Based on the predictions, the machine will calculate and print the evaluation measures out. The evaluation measures, including accuracy, precision, recall and F1-score, are very important for the classification models. The training data is like a confusion matrix with four parts: True positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) [14]. The explanation of these metrics can be found as follows: 1) TP: The true positive metric assesses the accuracy with which a model correctly identifies instances belonging to the positive class. 2) FP: False positives arise when a model erroneously labels an instance as belonging to a particular class. 3) TN: True negatives correspond to the cases where a model accurately identifies an instance as belonging to the negative class. 4) FN: A false negative is an error that occurs when a model incorrectly identifies an instance as negative, despite it actually belonging to the positive class. 5) Precision Score: The precision score of a model reflects the proportion of samples that are correctly labeled as positive, out of all samples that the model has predicted as positive. 6) Recall Score: The recall score of a model measures its ability to correctly identify positive samples out of all actual positive samples. 7) Accuracy Score: Model accuracy is a widely used performance metric for machine learning classification models that reflects the proportion of correct predictions out of all predictions made by the model. 8) F1 Score: The F1 score is a popular machine learning performance metric that combines the precision and recall scores of a model into a single value. And next is the formula about how to calculate the scores:

$$\text{Precision Score} = TP / (FP + TP) \quad (1)$$

$$\text{Recall Score} = TP / (FN + TP) \quad (2)$$

$$\text{Accuracy Score} = (TP + TN) / (TP + FN + TN + FP) \quad (3)$$

$$F1 \text{ Score} = 2 * \text{Precision Score} * \text{Recall} \frac{\text{Score}}{\text{Score}} (\text{Precision Score} + \text{Recall Score}) \quad (4)$$

However, the Scikit-Learn library includes the formula to calculate the evaluation measures, and it only needs y_{test} data and the prediction.

Based on the scores of each model, this project gives four possible logical formula: 1) logical and. 2) logical or. 3) complex formula clude logical or, and logical and. (prediction_nb | (prediction_lr & prediction_dt & prediction_svm) | prediction_rf) 4) complex formula clude logical or, logical and, and logical xor. (prediction_nb | (prediction_lr & prediction_dt) | (prediction_svm & prediction_rf)) Then calculate the evaluation measures about these four formulas and compare the result with the single models.

3. Result and Discussion

The results of each single model based on different evaluation metrics are shown in Table 1.

Table 1. The performance of the model in terms of various evaluation metrics.

	Accuracy score	Precision score	Recall score	F1 score
Naïve Bayes	0.9878	0.9615	0.9459	0.9537
Logistic Regression	0.9799	0.9876	0.8595	0.9191
SVM	0.9821	1.0	0.8649	0.9275
Decision Tree	0.9706	0.9045	0.8703	0.8871
Random Forest	0.9821	1.0	0.8649	0.9275

Based on the data presented above, it can be observed that Naïve Bayes' algorithm has produced exemplary results with all scores above 0.95. In contrast, Logistic Regression, Support Vector Machine, and Random Forest algorithms have only managed to achieve a recall score below 0.9. Furthermore, among these algorithms, Decision Tree has exhibited inferior performance with only two scores surpassing the 0.9 threshold when compared with the other four models. The five models employed in the study exhibit high levels of accuracy with almost all spam messages correctly classified, indicating the efficacy of the training process. The Support Vector Machine and Random Forest models have achieved a perfect precision score, which inevitably results in a lower recall score. Similarly, the Logistic Regression model has recorded an impressive precision score but with an unfavorable recall score. In contrast, the Decision Tree model has only attained a precision score of 0.9, with the recall score remaining below the 0.9 threshold, indicating its unsuitability for spam classification. The F1 score, useful in cases where precision and recall scores have a considerable disparity, reveals that the Logistic Regression, Support Vector Machine, and Random Forest models have excellent scores. However, the Decision Tree model's F1 score still falls behind those of the other models, further substantiating its inadequacy for spam classification.

Table 2 provides the result of the logical formula.

Table 2. The performance of the model based on the logical formula.

	Accuracy score	Precision score	Recall score	F1 score
Logical And	0.9734	1.0	0.8	0.8889
Logical Or	0.9777	0.8812	0.9622	0.9199
Complex_formula1	0.9878	0.9615	0.9459	0.9537
Complex_formula2	0.9886	0.9568	0.9568	0.9568

The four logical formulae employed in the study have exhibited consistent high levels of accuracy. However, the Logical And formula has recorded a recall score of only 0.8, falling below the 0.9 threshold. Additionally, the Logical Or formula has achieved a precision score below 0.9. Moreover, the Logical And formula's F1 score is notably lower than that of the other formulae, given its difference between precision and recall scores of 0.2. Conversely, the F1 scores of the last two predicted formulae have been impressive, indicating their suitability for the classification task. The Logical And prediction technique has demonstrated the tendency to overlook significant data by producing consistent identical results across all five single models. Consequently, it records a perfect precision score but a subpar F1 score. Conversely, the Logical Or prediction technique has a propensity to include some erroneous data by predicting all data as "1," similar to the Decision Tree model. In contrast, the first complex formula employed Naïve Bayes' prediction and the same data in three models, namely, Logistic Regression, Decision Tree, and Support Vector Machine, as well as Random Forest's prediction, resulting in similar outcomes to Naïve Bayes' prediction. Here, the inclusion of the Decision Tree model's prediction is to mitigate the impact of incorrect data. A similar approach was employed in the second complex formula, using the Logical And technique to combine the predictions of Logistic Regression and Decision Tree, thus retaining only the "true" data. Furthermore, the Logical Xor technique was used to identify some "missing" data not found in other models, which, upon integration, led to improved performance measures exceeding those of Naïve Bayes' algorithm.

4. Conclusion

This project has identified a possible logical formula that combines multiple models to improve spam-filter performance, which outperforms the Naïve Bayes model. While this logical formula has exhibited favorable evaluation measures, it is plausible that a new formula with even higher scores could be identified, or that the current formula may not perform well with different training data. Nonetheless, this project highlights the potential for improving spam-filter technology through the integration of multiple models. This idea can be applied not only to spam-filter training but also to other AI training models, as single models have their inherent advantages and limitations, and combining multiple models can potentially yield more robust and accurate training data.

References

- [1] Statista. Number of sent and received e-mails per day worldwide from 2017 to 2025 available as: <https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/>
- [2] Kaur G, et al. A Survey on Various Classification Techniques in Email Spamming. *International Journal of Technology and Computing (IJTC)* 5.3, 2016, 589-593.
- [3] Jordan M I., and Tom M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science* 349.6245, 2015, 255-260.
- [4] Rusland N F et al. Analysis of Naïve Bayes algorithm for email spam filtering across multiple datasets. *IOP conference series: materials science and engineering*. Vol. 226. No. 1. IOP Publishing, 2017.
- [5] DeBarr D, and Harry W. Spam detection using clustering, random forests, and active learning. *Sixth Conference on Email and Anti-Spam*. Mountain View, California, 2009.
- [6] Amayri O, and Nizar B. "A study of spam filtering using support vector machines. *Artificial Intelligence Review* 34, 2010, 73-108.
- [7] Olatunji S O. Improved email spam detection model based on support vector machines. *Neural Computing and Applications* 31, 2019, 691-699.
- [8] Alghoul A, et al. Email classification using artificial neural network, 2018.
- [9] Dedetürk B K, and Bahriye A. Spam filtering using a logistic regression model trained by an artificial bee colony algorithm. *Applied Soft Computing* 91, 2020, 106229.

- [10] Daisy S. Jancy S, and A. R B. Smart material to build mail spam filtering technique using Naive Bayes and MRF methodologies. Materials Today: Proceedings 47, 2021, 446-452.
- [11] UCI, machine-learning-databases (unknow) available at: <https://archive.ics.uci.edu/ml/machine-learning-databases/00228/>
- [12] Qader W A., Musa M A, and Bilal I A. An overview of bag of words; importance, implementation, applications, and challenges. 2019 international engineering conference (IEC). IEEE, 2019.
- [13] Great Learning Team An Introduction to Bag of Words (BoW) | What is Bag of Words? available at: <https://www.mygreatlearning.com/blog/bag-of-words/>
- [14] Ajitesh Kumar Accuracy, Precision, Recall & F1-Score – Python Examples available at: <https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/#:~:text=Recall%20score%20is%20used%20to,the%20classes%20are%20very%20imbalanced,2013>