Spam Email Detector

Group 11

Didulani P.K.S
EG/2020/3894
Faculty of Engineering University of Ruhuna, Sri Lanka
Hapugala, Galle

Dinuk P.D
EG/2020/3901
Faculty of Engineering University of Ruhuna, Sri Lanka
Hapugala, Galle

***Abstract*—The surge in email users has led to a significant rise in privacy concerns due to the increasing prevalence of spam emails, which not only pose a risk to personal information but also consume valuable time. Spam emails can range from malicious content to unwanted commercial marketing messages, making it imperative to implement effective detection and filtering mechanisms. This paper focuses on employing machine learning (ML) algorithms to address this issue, utilizing a supervised ML technique on an established email classification dataset.**

This study explores the effectiveness of two Machine Learning algorithms, including Naïve Bayes, Logistic Regression. A dataset consist of two columns have used for this study. After adding some pre-processing techniques, the dataset was ready to use. Then after extracting features from the dataset and vectorizing them by using some algorithms, the data was fed to train the models. Their performance was assessed based on metrics such as accuracy, precision, recall, and F1 score. Final findings demonstrate the efficacy of ML techniques in effectively detecting and filtering spam emails, contributing to enhanced email security and user experience.

I. Introduction

With the increase in emails, the users are facing various difficulties to manage spam emails without looking at them. Spam emails can have various negative effects. Security risks, financial losses, productivity losses, and privacy concerns are some of them. Advanced email filtering systems develop various techniques to identify and filter out spam before it reaches the recipients. This study contains details about such a machine learning model, which can be used to identify spam and ham email. This model contains two machine learning algorithms that analyze email content. This kind of project is important as it addresses critical aspects of online security, user experience, and compliance. Not only that, from this kind of study, a person can gain knowledge about advancements in technology and how to address complex challenges in modern world.

The objective of this project was to develop a machine learning model that has the capability to identify whether an email is spam or not. For that, two machine learning algorithms were used. Naïve Bayes and logistic regression algorithms were the choice. Since spam detection is text classification, the Naïve Bayes is a best suited algorithm [1]. Since spam and ham are two classes, the logistic regression algorithm which is used for categorizing things into two classes is also suitable for this [2]. Apart from positives, according to the gathered data, difficulty with continuous features and sensitivity to feature quality are negatives of naïve bayes. Also, the linearity assumption and susceptibility to overfitting are the negatives of using logistic regression for this model [3].

II. Methodology

To train and evaluate our model the data set was taken from Kaggle website [4]. It contains random emails and its classifies as spam or ham. The 1st column contains spam/ham classification (the output variable) and the other column of the dataset have the mail itself. There were 5572 emails in this dataset, among them 4825 are ham and others are spam. So that the data set was bias towards the ham email. Further there were no null values and there were 403 duplicates emails. As the first step duplicate values were removed by keeping first occurrence of the duplicates. Then the shape of dataset decreases to 5169 data. Then data set was preprocessed by removing the punctuations, splitting the sentence into words, removing stop words and getting the base form of the word (Lemmatization and Stemming). After that the dataset was taken into a vocabulary which contains all the unique words of the whole dataset. That vocabulary contained 8084 words which will give low accuracy because the feature count is more than the dataset. To overcome this, for the vocabulary the unique words appeared more than 10 times were taken. After preprocessing and feature extraction the data was split into train and test containing 80% data from the data set as training data for the two models.

Using a vectorization algorithm, the data was transformed to an array of row arrays which has created with the help of the vocabulary created above. The method was, adding 1s to the array if the word in the sentence is present in the vocabulary otherwise add 0 to the array. Here the array size is fixed and the array is the size of vocabulary. Before training the model,it had to undergo balancing methods as the data set is imbalanced, otherwise model tend to bias towards the majority class. For that oversampling SMOTE method which generates the new values using minority class was used. From that process the imbalances were balanced. As the final process, the dataset using naïve bayes and logistic regression algorithms were trained and observed the results.

III. Algorithm

When talking about two algorithms with this spam email detection, finally the goal was to get a measure of how spammy as incoming email is. Naïve Bayes Algorithm and Logistic Regression Algorithm were used for the study and both these algorithms are categorized under supervised learning algorithms and also both the algorithms are classification algorithms.

A. *Naïve Bayes Algorithm*

Naïve bayes algorithm is best for the text classification. With Bayes' Rule, we want to find the probability an email is spam, given it contains certain words. We do this by finding the probability that each word in the email is spam, and then multiply these probabilities together to get the overall email spam metric to be used in classification [5].

$$p(S/w) = \frac{p(w/S) \times P(S)}{P(w)} \quad (1)$$

The probability of an email being spam S given a certain word W appears is defined by the left-hand side of the above equation. The right-hand side gives the formula to compute this probability.

- The probability the word occurs in the email given it is a spam email P(W|S) multiplied by the probability of an email being spam P(S).
- Divided the probability the word occurs in the email given it is a spam email

When the P(S|W) has been found for each word in the email, they are multiplied together to give the overall probability that the email is spam [5].

B. *Logistic Regression*

Logistic regression algorithm is used for categorizing things into two classes. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function.

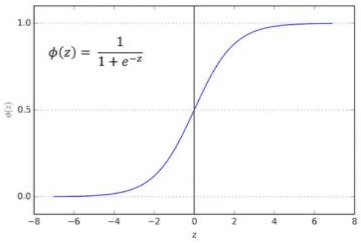

Figure 1 Sigmoid Function

The sigmoid function in Figure 1, maps any real-valued number x to a value between 0 and 1. As x approaches positive infinity, $\phi(z)$ approaches 1, and as x approaches negative infinity, $\phi(z)$approaches 0. The midpoint of the curve, where $\phi(z)$=0.5, occurs at x=0.

Logistic regression establishes a decision boundary based on the calculated probabilities. The standard threshold is 0.5. if the predicted probability is greater than 0.5, the email is classified as spam. otherwise, it's classified as non-spam.

IV. Implementation

As the above algorithms were chosen for the model implementation. A research was carried out to gather information about these algorithms. The base code idea was sourced from the YouTube [6].Optimizations are carried out with the help of further studies. Different data visualization methods were used. Such as count plots, pair plots and image plots. Duplicate values were checked and removed the duplicates keeping the first occurrence. Then preprocess and feature extraction were done. After the model was trained using two algorithms the evaluation was carried out and obtained the results. Additionally hyperparameters were tuned using grid search method and chosen the best parameters which optimized the process to the best.

Alpha, force alpha ,fit prior and class prior parameters were tested under naïve bayes. For the alpha the default value was 1 but e^{-08} was the best parameter. For the class prior the default parameter was none. But for search we gives [0.3,0.7] among them received the result [0.3,0.7] as the best parameter. Then for the fit prior and force alpha the default value was true for the best value also grid search has proven True is suitable.

Four parameters were given to test logistic regression to find the best parameters among them. For Inverse of regularization strength(C) the default value was 1 and among the list given ([0.001, 0.01, 0.1, 1, 10, 100]), grid search choose 100 as the best parameter. For the penalty the default parameter was l2 also the best parameter the grid search given was l2. For the solver the default parameter was 'lbfgs' but from gird search has given 'liblinear'. For the multi class parameter the default parameter was auto and also the best parameter was the same.

V. Results

When evaluating the model, the results received from the Logistic Regression are tabularized below,

Table 1 Different score for logistic regression

Different scores	Training data Set	Test data Set
Accuracy score	0.9745	0.9661
Precision score	0.9637	0.9864
F1 score	0.9748	0.9804
Recall score	0.9861	0.9744

Confusion metric for test set,

$$\begin{bmatrix} 123 & 12 \\ 23 & 876 \end{bmatrix}$$

When evaluating the model the results received from the Naïve Bayes are tabularized below,

Table 2 Different scores for Naive Bayes

Different scores	Training data Set	Test data Set
Accuracy score	0.9415	0.9381
Precision score	0.9409	0.9815
F1 score	0.9415	0.9637
Recall score	0.9422	0.9466

Confusion metric for test set,

$$\begin{bmatrix} 119 & 16 \\ 48 & 851 \end{bmatrix}$$

The score values were changed after tuning hyperparameters,

Table 3 After tuning hyperparameters

Logistic Regression	Naïve Bayes
0.9419	0.9642

Cross-validation is a statistical technique used to assess the performance of a machine learning model by dividing a dataset into subsets, training the model on some of these subsets, and evaluating it on the remaining data. The primary goal is to obtain a more robust and reliable estimate of the model's performance [3].K Fold cross validation method was used with the help of sklearn library.
The dataset is divided into k folds, and the model is trained and evaluated k times, each time using a different fold as the validation set. The results are tabularized below.

Table 4 Cross Validation Scores

Logistic Regression	Naïve Bayes
0.9791	0.9742

10 Folds were used and selected one part by one as test data and other 9 parts as train and get the accuracy for it. Then the mean value was taken as the cross validation score.

VI. Discussion

From the Table *1* a slightly higher training accuracy score and recall score was observed when compared to the test accuracy score and recall score. However, the marginal difference between the two accuracies suggests that the model generalizes well to unseen data, and overfitting is not a significant concern. Also, the precision score and F1 score were higher on the test data compared to the training data. Balancing data in the train set might be the reason for that. It may result to produce duplicate values in the train set. So that this can be happened. Also, there can be some several reason. Such as feature drift , randomness and variability and data quality [3].

The observation from Table 2 all the scores except the accuracy score in the test scores are higher than in the training scores. The reason for that can be the data set used is too small and when spliting the test and train set the dataset size decrease further more. Data leakage, evaluation metric and randomness and variability can be some other reason for that. But for accuracy score the train accuracy score is higher than test accuracy score. However, the marginal difference between the two model generalizes well to unseen data, and overfitting is not a significant concern.

After tuning the model for the best hyperparameter for the logistic regression the accuracy was lower than the previous accuracy. It decreases from 0.9745 to 0.9419. There may be the several reason. Such as data drift between validation and test sets, small test set size and randomness in model training. For the naïve bayes after tuning the hyperparameters the accuracy increases from 0.9381 to 0.9642.

From Table 4 we can see the cross-validation score of the logistic regression is 0.005 higher that the cross-score of the naïve bayes. So that for the spam detection the better model is logistic regression than the naïve bayes.

Acknowledgment

This would not have been possible without the guidance and assistance of several individuals who contributed and extended their valuable assistance in the preparation and completion of this project in various ways. First and foremost, we would like to express our heartfelt gratitude to our Dean of the faculty Dr.Chithral Ambawatte for providing us this background to work for this kind of projects. Then we would like to thank the Department of computer Engineering for offering this kind of module to learn the future trends. Then we would like to express our heartfelt gratitude to Dr.Rajitha Udawalpola ,Dr. Noelin Prins and Mr. Charuka who were the great characters helping us from the beginning to the end to succeed in this project.

References

[1] L. Tadjpour, "Naive Bayes and Spam Detection," 8 June 2016. [Online]. Available: https://opendatascience.com/naive-bayes-and-spam-detection/. [Accessed 21 January 2024].

[2] N. Sharma, "Spam Detection with Logistic Regression," Towards Data Science, 5 May 2018. [Online]. Available: https://towardsdatascience.com/spam-detection-with-logistic-regression-23e3709e522. [Accessed 21 January 2024].

[3] "ChatGPT," OpenAI, January 2022. [Online]. Available: https://chat.openai.com/. [Accessed 19 January 2024].

[4] D. SHANTANU, "Email Spam Detection Dataset (classification)," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/shantanudhakadd/email-spam-detection-dataset-classification. [Accessed 16 November 202].

[5] M. Garvey, "Naïve Bayes Spam Filter—From Scratch," Towards Data Science, 30 November 2020. [Online]. Available: https://towardsdatascience.com/na%C3%AFve-bayes-spam-filter-from-scratch-12970ad3dae7. [Accessed 20 January 2024].

[6] [Online]. Available: https://www.youtube.com/@codeprolk.