An email spam prediction system using logistic regression.

R.L. Thomas
IT19095936
Machine learning for cyber security
Cyber security
SLIIT Y4S1

Abstract

Due to spam-based phishing assaults across emails, sensitive information such as bank data, medical information, and credit/debit card details (username password) has been leaked. Spam email is one of the most serious risks to cyber security [1]. It was a simple chore for someone to create a bogus email account and begin spamming. Since a result, an email spam detection system is required, as the danger has grown to be a major concern for email users [4]. This study provides a python-based machine learning-based spam detection technique. The dataset is separated into various sets and supplied as input to each algorithm in this procedure [6].

Introduction

One of the most successful and widely utilized forms of communication is email [1] [7]. A typical individual might receive a considerable number of emails every day from a variety of sources, depending on their daily activities, which include social networking, online banking, online buying, and ecommerce [2]. It should be feasible to tell the difference between important and valuable communications and spam or junk mail. When a person is exposed to spam and other potentially hazardous sources, he will get a flood of emails from unknown senders [4]. As a result, choosing and evaluating all of the received emails, which may include important facts or information, becomes a difficult and time-consuming task for an email user [6]. The security and privacy of the system are threatened when an email client is fooled into doing a malicious behavior [8]. The email user may become a victim of a phishing scam perpetrated by cyber thieves. It's tough to recover from such situations, and most email users are lured to and react to spam emails. Because the senders' locations change all the time, blocking and reporting these spam email sources is usually futile [4]. The technique of detecting spam emails may be divided into two categories: Machine learning and knowledge engineering Knowledge engineering is a network-based approach for categorizing emails that considers IP (internet protocol) addresses, network addresses, and a set of specified criteria [3] [4]. Although the procedure has shown to be effective, it is time consuming. The labor of maintaining and modifying rules is inconvenient for certain users. Machine learning, on the other hand, is more efficient than knowledge engineering since it does not require any rules [5]. The classification system classifies the email based on its content and other factors [6]. Support vector machines, artificial neural networks, logistic regression, and naive Bayesian classifier are some autonomous algorithms that can be used to classify received emails using appropriate and approximate machine learning approaches and some autonomous algorithms like support vector machines, artificial neural networks, logistic regression, and naive Bayesian classifier [8]. In this work, I will use a logistic regression model to solve the spam message problem [3]. The rest of the paper is structured in the following manner: **Section II - Related work** shows the work of authors related to email filtering. **Section III - Data set** includes the details of data set that will be used in this model to train. **Section IV - Methodology** shows the workflow and steps that will be used to build the model. **Section V - Results & Evaluation**, the modules of the proposed Spam Mail Detection are explained along with working example and illustrates the calculated experimentation results and **Section VI - Conclusion** concludes the paper.

Related work

Email is one of the most common and commonly used communication platforms. Efforts are being made by organizations all around the world to identify spam emails [1]. The authors' efforts to identify ham and spam emails are addressed in this article.

Nikihila et. al. [1] Examines and analyzes the outcomes of numerous ways for decreasing the logistic loss function in the spam filtering problem. The goal of this study is to figure out if an email is spam or not, and logistic regression is widely regarded as one of the best strategies for determining whether an email is spam or not. Three alternative types of algorithms for minimizing of logistic regression are researched and implemented: the Stochastic Gradient Descent Algorithm, Regular Batch Gradient Descent Algorithm, and Regularized Gradient Descent Algorithm. The paper concludes that in the Stochastic Gradient Descent algorithm, which employs the simulated annealing technique, it is unclear how to control the weight vector optimally, whereas performance in normal gradient descent was improved on the test set due to the prevention of overfitting in the training data.

Qingha et. al. [3] conducts an assessment of commonly used methods for preventing e-cheating and demonstrates how biometrics might be utilized for this purpose. The author proposes a novel way for watching student actions by leveraging their IP addresses and timestamps to aid in the detection of possible cheating. The results suggest that the proposed strategy is successful in detecting student collision during exams.

Shadi khawandi et. al. [6] share their concerns about picture spam detection, which has been a severe issue for years and for which many companies have offered a variety of remedies. This document explains the methods for preventing spam and the options available for dealing with spam and image-based spam. The article states that current anti-spam solutions are insufficient because most mail servers rely on blacklists, while others rely on filters with a high rate of false positives.

Kamoru et. al. [9] the goal was to look into existing research on spam detection technologies, the mechanism that these methods use, and other mitigation systems. This study examines a variety of anti-spam solutions for email and social media. For the welfare of the planet, the author highlights the need of focusing on spam identification. This research uncovers new concerns and challenges that must be addressed, posing a significant research challenge.

Kamoru et, al [11]. For the goal of spam identification, sights approach does research on several algorithms. [8] Content-based filtering and rule-based filtering are the two types of algorithms investigated. This study calculates and investigates a number of content-based filtering algorithms. It has been determined that rule-based filtering is the most efficient approach of creating a spam filter since it minimizes filtering time.

M. E. Tipping and C. M. Bishop et.al. [12] One of the common principles of machine learning approaches is the use of decision trees as a probabilistic classifier.

S. Nasser, R. Alkhaldi, and G. Vert et.al. and F. [15] Using fuzzy sets and fuzzy systems, they've applied their scholarly ideas and investigations to a variety of datasets..

Dataset

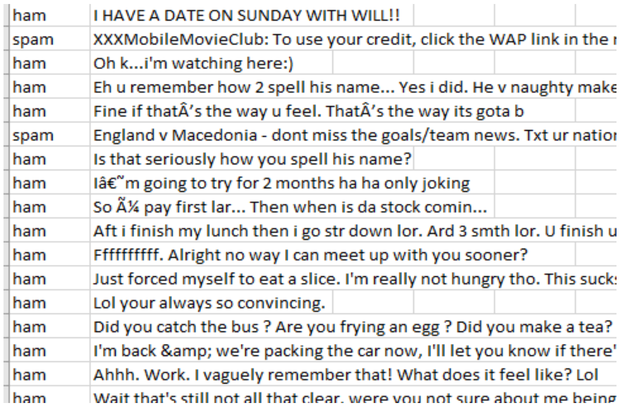
For the Spam mail detection system, an email dataset is created. Various emails are selected at random from the internet. For categorization purposes, the dataset contains a total of 5000+ emails, including both ham and spam emails. *Table 2* and *Table 3* shows the statistics of dataset.

Email dataset	**value**
No. of Ham emails	4870
No. of Spam emails	751
Total	5571

Table 2. Email dataset statistics

Email dataset	**percentage**
Ham emails for training	40%
Spam mails for training	40%
Ham emails for testing	10%
Spam mails for testing	10%

Table 3. Email dataset statistics II



ham	I HAVE A DATE ON SUNDAY WITH WILL!!
spam	XXXMobileMovieClub: To use your credit, click the WAP link in the
ham	Oh k...i'm watching here:)
ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make
ham	Fine if thatÂ’s the way u feel. ThatÂ’s the way its gota b
spam	England v Macedonia - dont miss the goals/team news. Txt ur natio
ham	Is that seriously how you spell his name?
ham	Iâ€˜m going to try for 2 months ha ha only joking
ham	So Ã¼ pay first lar... Then when is da stock comin...
ham	Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish u
ham	Ffffffffff. Alright no way I can meet up with you sooner?
ham	Just forced myself to eat a slice. I'm really not hungry tho. This suck
ham	Lol your always so convincing.
ham	Did you catch the bus ? Are you frying an egg ? Did you make a tea?
ham	I'm back &amp; we're packing the car now, I'll let you know if there'
ham	Ahhh. Work. I vaguely remember that! What does it feel like? Lol
ham	Wait that's still not all that clear, were you not sure about me being

Fig 1. Examples of Messages in the Dataset

Figure 1 is a snapshot of Spam research-related email that has been labeled. It comprises a single collection of English communications totaling 5,571 emails that have been classified as authentic (ham) or spam. It's worth noting that each token in a message has some sort of discriminating power. For example, if the token 'congratulations!!!!' appears frequently in spam email and seldom in non-spam letters, it is a desirable trait of a spam, allowing a mail including this phrase (with triple exclamation) to be identified as spam [3]. Similarly, in the text, a series of capital letters (upper case) is another characteristic that will have its own categorization power [3] [4].

Methodology

The efficacy of the machine learning technique is used to develop the proposed spam mail detection system. Email data is initially acquired via a spam mail detection system. The acquired email data is disorganized and unfiltered [5]. To reduce computations and produce accurate results, email data must be pre-processed. To obtain relevant information, the data is pre-processed by eliminating stop words, capitalizing word tokenization, and minimum document frequency [15] [14]. The pre-processing stage decreases the dimensionality of the input before extracting features in the form of a bag of words [16]. The data set will then be divided into two sections: train data and test data. The train data will be used to insert into the logistic regression model to train the model, while the other data will be used to test the trained model. Following the training of the module, a spam detection system will be constructed to recognize and filter spam and ham emails. *Figure 2* shows the process overview of building the module.

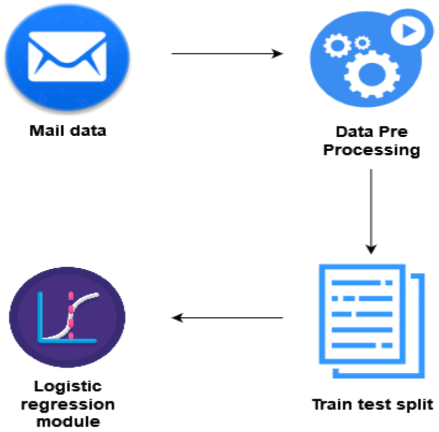


Fig 2. Email spam detection system

Logistic Regression

For the categorization of datasets, logistic regression is one of the most likely and appropriate algorithms [14]. The Logistic Regression approach may be used to classify data into distinct categories of categorical response variables using input factors since it models categorical response variables. One example is the Emails are classified as spam or non-spam. Logistic regression is the most flexible decision-based approach for detecting spam emails in a dataset when it comes to recognizing a dataset as spam [17]. The basic tests performed by logistic regression on a given data distribution include detecting and computing statistical domains like mean and standard deviation, as well as creating results from operations like word and character count, max and min operations [14]. The results of statistical and count tests are obtained and provided by the logistic regression approach, which tends to inter-relate the findings. The table below shows the definition of one of the basic functions used in logistic regression [14] [17]. Logistic regression is a fundamental statistical approach that aims to predict a data value based on past distributions and observations. A logistic regression approach establishes a link between one dependent variable and one or more dependent variables [17]. Machine learning and deep learning algorithms for spam and non-spam email classification are demonstrated and compared. Different machine learning and deep learning methods for classifying a dataset into a number of categories or variables may be compared and distinguished based on their definition and accuracy [14].

Algorithm used for classification	**Accuracy of classifying objects**
Chi square function	92%
Naïve Bayes classifier	75-92%
Fuzzy systems	92%
Decision tree C4.5	60%

SVM	76%
KNN	89%
Logistic regression	80%

Table 3.1.2. Comparison of different algorithms

The suggested system architecture (Fig. 2) displays multiple phases of email spam detection throughout both the training and testing phases. The following are the phases in detail:

1. Tokenization, stop word removal, stemming, lemmatization, and dataset cleansing are all procedures in the pre-processing phase.
2. The feature selection step is implemented to decrease the dataset and amount of features; consistency-based approaches were used to accomplish this.
3. The dataset was then split into two parts: training and testing.
4. The training dataset was exposed to an ensemble of algorithms for classification.
5. The classification findings were then verified using evaluation metric accuracy.
6. The output of Email spam detection is the categorization findings.

Preprocessing

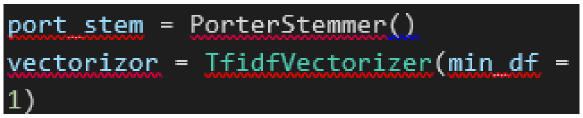
To capture the problem being addressed, the following preprocessing procedures were used to eliminate irregularities and inconsistencies from the dataset gathered. A feature engineering strategy was applied to the dataset during the preprocessing step to enable the machine learning algorithm to deliver correct findings [11]. Before entering the algorithm, the dataset was cleaned and processed using techniques such as tokenization and stop word removal. As a result, the job included data analysis, iterations, and evaluation of the classifier's performance [8]. After processing, the data was converted to text, and a CSV file format dataset was utilized for this study.

The dataset was preprocessed using the python natural language processing program [12]. The feature extraction is the major emphasis of the preprocessing part. The consistency-based feature selection strategy was used in this study to reduce redundant characteristics in order to categorize and identify email spam more succinctly and precisely. Consistency metrics were used to evaluate the value of feature subsets in consistency-based feature selection. This metric is intuitively stated as a way of determining how far a feature subset is from the consistent state [11].

Packages

We have imported many packages in order to work on our project. The "pandas" package was used to read the dataset and use the "get dummies" function to transform categorical data into indicator variables like 0 and 1 [2]. To get functions like "stopwords" and "tfidvectorizer" to operate on the test processing, the "nltk" package was utilized. Text data was additionally processed using the "re" package (Regular Expression Operations). Importing the "sklearn" package provided the "train test split" and "logisticRegression" methods [3] [7]. The data was split into training and testing datasets using the "train test split" function, and the prediction was modelled using the "logisticRegression" function [11]. The confusion matrix of our final result was plotted using Sklearn metrics tools. The "joblib" package was imported to preserve the model and reuse it without having to redo the entire prediction procedure [18].

The total amount of characters in each column of the dataset, as well as the number of rows and punctuation in the dataset, were taken into account. The approach for extracting features was referred to as features engineering [19]. Feature engineering was performed firstly by loading the email dataset, stemming and performing feature engineering on the email data by using a nltk and tfidfvectorizers. Figure 3 shows the both function names in the code



```
port_stem = PorterStemmer()
vectorizor = TfidfVectorizer(min_df =
1)
```

Fig 3. Stemming process of the model

Removal of Stopwords

Stopword elimination is a frequent data processing step in NLP. The goal was to exclude all common terms that appeared in the dataset more often [17].

Stopwords are words that are commonly used in text messages but have no or limited significance, such as you, me, a, him, and so on. While conducting a text analysis experiment, some words have characteristics and are significant in the text message; these words are referred to as features, whilst others do not contribute at all and are referred to as stopwords [17]. Articles and pronouns are usually categorized as stop words. The messages (emails in the dataset) must be understandable for the Machine to interpret, analyze, and perform Natural Language Processing on the data. Because machines cannot comprehend human language, we must preprocess the data to make it machine-readable [11]. We must remove any unnecessary data from the dataset in order for it to be clean. Stopwords are words that are completely worthless. Stopwords include words like 'is,' 'are,' 'a,' 'as,' and so on. Stopwords are frequently used in NLP and even text mining to filter out irrelevant data [18].

Min df

Min df stands for minimum document frequency; unlike term frequency, which counts the number of times a word appears in the whole dataset, document frequency counts the number of documents (aka rows or entries) that include the word [17]. Min df excludes terms with a document frequency that is strictly lower than the provided threshold while creating the vocabulary. For example, some emails in the dataset contain terms that appear in just one or two emails; these might be omitted since they do not give enough information about the whole dataset as a whole, but only a few specific emails [14]. min df is capable of accepting absolute values (1,2,3..) In this scenario, the Min df value is set to 1.

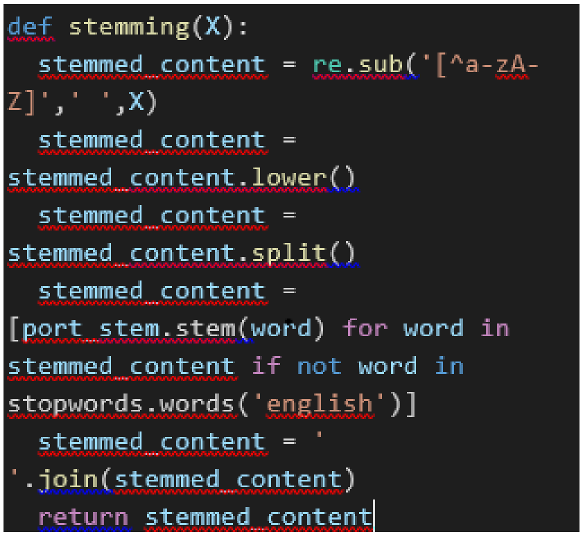
Lowercase

Convert all characters to lowercase before tokenizing. Default is set to true and takes Boolean value. This makes the pre-processing process easier for the model to train [13].

Stemming

For the model I have applied NLTK's stemming for the message labels, which improved the prediction accuracy of spam or ham emails. The process of stemming reduces the word to its base word [12]. Stop word removal and stemming are important steps in the pre-processing phase as they help to reduce the search space for efficient feature extraction and selection [6].

Figure 4 shows the code of stemming process of the model.



```
def stemming(X):
  stemmed_content = re.sub('[^a-zA-
Z]',' ',X)
  stemmed_content =
stemmed_content.lower()
  stemmed_content =
stemmed_content.split()
  stemmed_content =
[port_stem.stem(word) for word in
stemmed_content if not word in
stopwords.words('english')]
  stemmed_content = '
'.join(stemmed_content)
  return stemmed_content
```

Fig 4. Stemming process of the model

Steps followed by the model to classify the spam and non-spam emails;

- Importing the dependencies
- Dataset collection
- Replacing the null values of data
- Label encoding of dataset
- Separating the data as label
- Stemming with nltk
- Feature extraction
- Splitting the dataset into training data & test data
- Training the model
- Testing the model
- Testing the accuracy
- The final model with the capability of detecting spam and non-spam emails.

After building the model I had to extract the model using joblib. Saving your learned machine learning models is a vital step in the machine learning process since it allows you to reuse them later. For example, you'll almost certainly need to compare models to select which one to use in production – storing the models once they've been trained makes this process easy. Otherwise, it will have to learn how to utilize the model anew every time we use it.

Results and evaluation

The model's experimental findings are reported in this section. The method was evaluated using an email dataset of 5571 emails, ham and spam, and the algorithm itself. Figure 5 shows the systematical flow or classifying the emails.

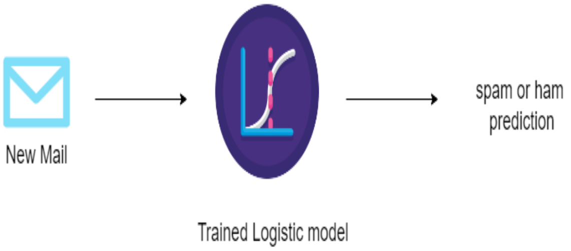
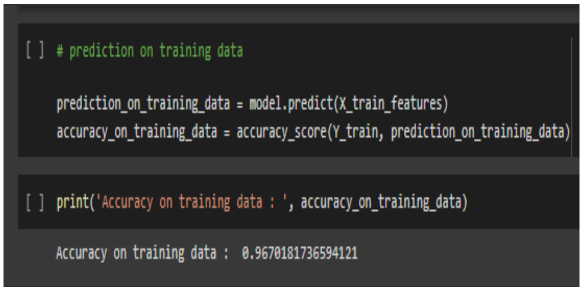


Fig 5. Spam prediction system flow

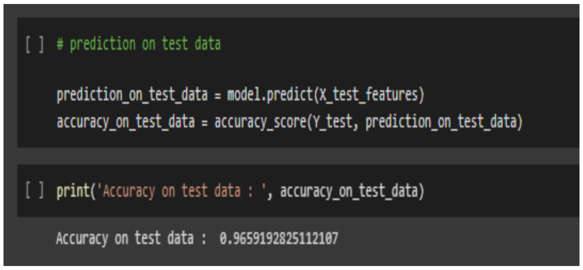
The ultimate result of the model is the aggregate of the algorithm's predictions, resulting in a system that is accurate and dependable.



```
[ ] # prediction on training data

    prediction_on_training_data = model.predict(X_train_features)
    accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)

[ ] print('Accuracy on training data : ', accuracy_on_training_data)

    Accuracy on training data :  0.9670181736594121
```

Fig 6. Accuracy on training data

According to *figure 6* the prediction of training data shows an accuracy of 96.7% for classification on spam and non-spam by accuracy metrics.



```
[ ] # prediction on test data

    prediction_on_test_data = model.predict(X_test_features)
    accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)

[ ] print('Accuracy on test data : ', accuracy_on_test_data)

    Accuracy on test data :  0.9659192825112107
```

Fig 7. Accuracy on test data

According to *figure 7* the prediction on test data shows an accuracy of 96.5% of filtering spam and non-spam by accuracy metrics.

In certain cases, a score of 96 percent may appear to be exceptional [4]. Other ways to increase accuracy with the gathered data include stemming the words and standardizing the length.

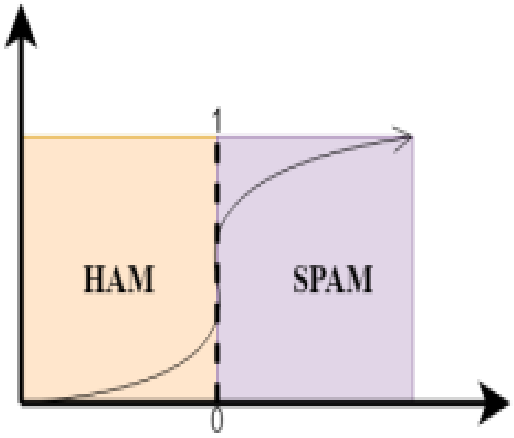
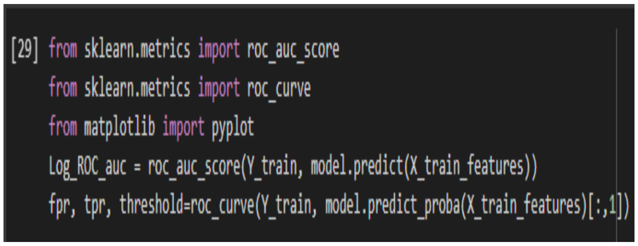


Fig 8. Logistic regression data classifier

ROC AUC

An ROC curve is a graph that shows the relationship between TPR and FPR. The closer the AUC value gets to 1, the better developed the model is [9]. It may be estimated using R and Python tools. The AUC is 0.97, which is closer to one, as seen by the ROC-AUC curve. As a result, we may infer that the model is complete in terms of classifying whether the text in each brief message is ham or spam.



```
[29] from sklearn.metrics import roc_auc_score
     from sklearn.metrics import roc_curve
     from matplotlib import pyplot
     Log_ROC_auc = roc_auc_score(Y_train, model.predict(X_train_features))
     fpr, tpr, threshold=roc_curve(Y_train, model.predict_proba(X_train_features)[:,1])
```

Fig 9. Code for true positive and false positive graph and charts

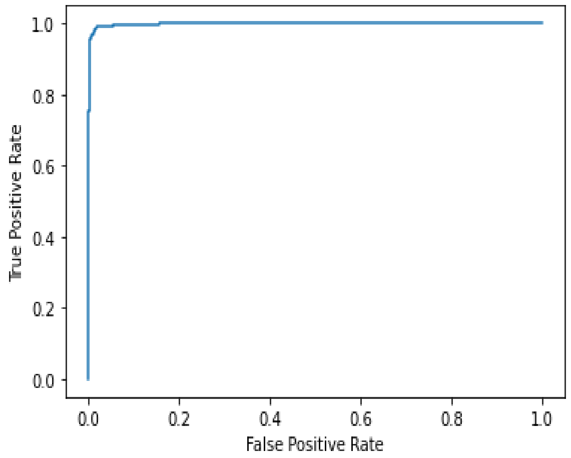


Fig 10. ROC Curve: AUC = 0.9759

The model is built using vscode and colab, and the training data is the whole set of 5571 records from the UCI data set [7]. For the selection of variables into the model, the stepwise forward conditional technique is used. This model is applied to each message in the testing data, and the resulting probability P[Y = 1] is saved. The mail is categorized as ham whenever P[Y = 1]. The LR model's performance is examined in terms of proper classification and the ROC curve.

Figure 3 shows a ROC curve with an Area under Curve (AUC) of 0.9759. It indicates that a randomly picked letter from the testing data with the list attributes of table-3 is 97.59 percent more likely to be spam than a non-spam when evaluated using the LR model. The process for testing this model with a different data set, the Enron data, is developed in the next section.

Precision and recall

Another metric used by machine learning researchers is accuracy and recall, which are two values. These concepts are defined as follows, and they originate from the field of information retrieval [14].

It's entirely accidental that the precision is so close to the previously specified accuracy level. The regularity with which a positive indication proves to be correct is defined as precision [14]. It's important to remember that accuracy is determined by the interactions between the classifier and the dataset. It's meaningless to ask about a classifier's precision in isolation; it's only appropriate to ask about a classifier's precision for a certain dataset [14].

Precision responds to the query, "*What is the chance that this email is spam if the spam filter says it is?"* The ratio of genuine positives to expected positives is known as precision [14]. *Figure 11* shows the heat map of probability.

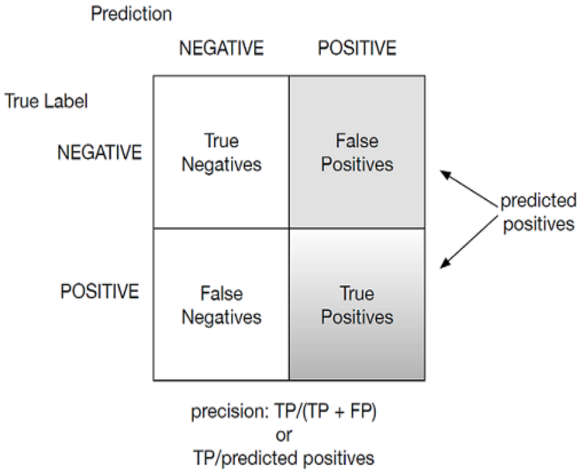


Fig 11. Probability of spam mails according to the model

The companion score to precision is *recall*. Recall answers the question *"Of all the spam in the email set, what fraction did the spam filter detect?" [14]*. Recall is the ratio of true positives over all positives.

When the data contains more spam than the filter was trained on, the filter performs better, resulting in a lower percentage of non-spam email being rejected [14]. This is amazing! When the data contains less spam than the filter was trained on, the accuracy drops, resulting in the filter rejecting a higher percentage of non-spam email. This is a terrible concept [14].

Because a classifier or filter may be used on populations where the prevalence of the positive class (in this case, spam) changes, having performance measures that are independent of the prevalence of the class is useful. Sensitivity and specificity are one such pair of measurements [14]. Because testing for illnesses and other disorders are performed on diverse populations, with varying frequency of a specific disease or condition, this combination of metrics is popular in medical research [14].

Embedding the model to web application

The web application has been successfully embedded with the model to use the platform as an email spam prediction system. *Figure 13* shows the systematical flow of embedding the model to the web application

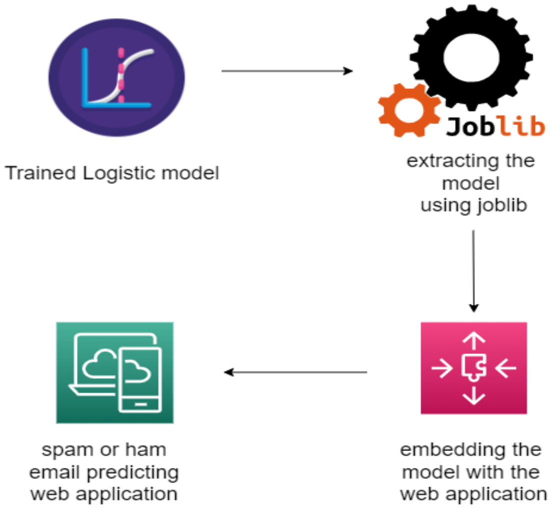
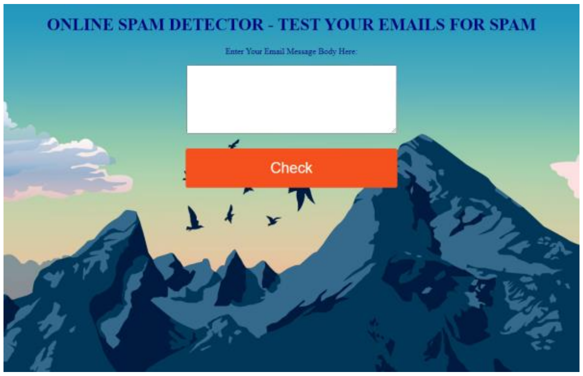

Trained Logistic model
Joblib
extracting the model using joblib
spam or ham email predicting web application
embedding the model with the web application

Fig 13. Embedding the to the web application

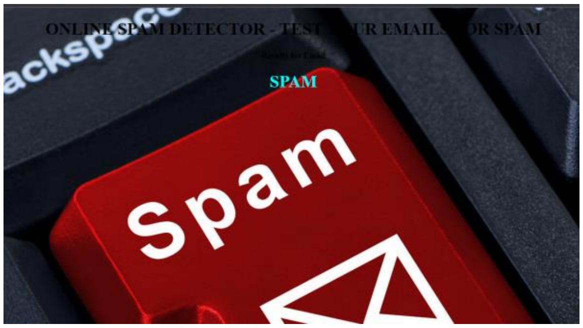
The model was finally deployed using Flask, a python library for web development, a micro framework used for developing small scale websites. Flask is very easy to make Restful API's using python. As of now, the logistic model which can predict emails as ham or spam has to be embedded into a web application. The interface shown in *Figure 14*. The web application reads email messages pasted into the given form and classifies them into spam or ham. The result is displayed in a page as shown in the *Figure 15*.



ONLINE SPAM DETECTOR - TEST YOUR EMAILS FOR SPAM

Enter Your Email Message Body Here:

Fig 14. The web application interface



ONLINE SPAM DETECTOR - TEST YOUR EMAILS FOR SPAM

Results for Email

SPAM

Fig 15. The spam prediction results page

Conclusion

The decision-based support system has been found to be effective in eliminating spam email sources. Criminals attempting to compromise the security of an email client can reveal an email user. Spam emails are sent out in the name of phishing, advertising, and promotions. By obeying the criteria established by the program created for that purpose, the email user can avoid getting spam emails in mass. Anyone may spot a spam email receipt and report or block that particular receipt. Another possibility is to track receipts by IP address, and a good system could tell the firewall to ban the spam source with the reported IP address, which is almost likely malicious. Another need is that the protocol suite be updated with information about receiving emails from a certain IP address and that the address be recommended for blocking. When it comes to corporate emails, it's tough for the administrator to recall the sender's name or the phrase that may be used to find a certain email. The ability of the administration to ban or report frequently recurring email is likewise limited. For this form of categorization, an email classifier can be quite useful. In the future, we may be able to detect whether or not a communication is spam using neural network and deep learning models. Deep learning performs well in natural language processing, but it needs a large quantity of data in order to provide correct findings and beat other machine learning methods. Because Natural Language Processing is a relatively unexplored area of study, the suggested system for spam detection and email filtering in the field of online security can be improved further.

Acknowledgement

I would like to thank Dr. Lakmal Rupasinghe for inspiring me to do this research, and my batch mates who helped me in completing this research.

References

[1] Kamoru, B. A., Jaafar, A. B., Murad, M. A. A., Ernest, E. O., & Jabar, M. B. A. Spam Detection approaches and strategies: A phenomenon.

[2] Medium. 2022. *Let's Learn about the ROC AUC Curve by Predicting Spam*. [online] Available at: <https://towardsdatascience.com/lets-learn-about-the-roc-auc-curve-by-predicting-spam-d8007746a6f9> [Accessed 7 June 2022].

[3] Gao, Q. (2012). Using IP addresses as assisting tools to identify collusions. International Journal of Business, Humanities and Technology, 2(1), 70-75.

[4] Zhong, X. (2014, July). Deobfuscation based on edit distance algorithm for spam filitering. In 2014 International Conference on Machine Learning and Cybernetics (Vol. 1, pp. 109-114). IEEE.

[5] Manning. 2022. *Evaluating a Classification Model with a Spam Filter - Manning*. [online] Available at: <https://freecontent.manning.com/evaluating-a-classification-model-with-a-spam-filter/> [Accessed 7 June 2022].

[6] Khawandi, S., Abdallah, F., & Ismail, A. A Survey ON IMAGE SPAM DETECTION TECHNIQUES. Computer Science & Information Technology, 13. Computer Science & Information Technology, p.13.

[7] Idris, I., Selamat, A., Nguyen, N. T., Omatu, S., Krejcar, O., Kuca, K., & Penhaker, M. (2015). A combined negative selection algorithm–particle swarm optimization for an email spam detection system. Engineering Applications of Artificial Intelligence, 39, 33-44.

[8] Brownlee, J., 2022. *How to Use ROC Curves and Precision-Recall Curves for Classification in Python*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/> [Accessed 7 June 2022].

[9] Kamoru, B. A., Jaafar, A. B., Murad, M. A. A., Ernest, E. O., & Jabar, M. B. A. Spam Detection approaches and strategies: A phenomenon.

[10] Y. Han, M. Yang, H. Qi, X. He and S. Li, "The Improved Logistic Regression Models for Spam Filtering," *2009 International Conference on Asian Language Processing*, 2009, pp. 314-317, doi: 10.1109/IALP.2009.74.

[11] Puri, S., Gosain, D., Ahuja, M., Kathuria, I., & Jatana, N. (2013). Comparison and analysis of spam detection algorithms. International Journal of Application or Innovation in Engineering and Management, 2(4).

[12] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," 1999.

[13] Krasser, S., Tang, Y., Gould, J., Alperovitch, D., & Judge, P. (2007, June). Identifying image spam based on header and file properties using C4. 5 decision trees and support vector machine learning. In 2007 IEEE SMC Information Assurance and Security Workshop (pp. 255-261). IEEE.

[14] Medium. 2022. *Spam Detection with Logistic Regression*. [online] Available at: <https://towardsdatascience.com/spam-detection-with-logistic-regression-23e3709e522> [Accessed 7 June 2022].

[15] S. Nasser, R. Alkhaldi, and G. Vert, "A Modified Fuzzy K-means Clustering using Expectation Maximization," pp. 231–235, 2006.

[16] Kumari, K. V., & Kavitha, C. R. (2019). Spam Detection Using Machine Learning in R. In International Conference on Computer Networks and Communication Technologies (pp. 55-64). Springer, Singapore.

[17] Taooka, Y., Takezawa, G., Ohe, M., Sutani, A., & Isobe, T. (2014). Multiple logistic regression analysis of risk factors in elderly pneumonia patients: QTc interval prolongation as a prognostic factor. Multidisciplinary respiratory medicine, 9(1), 59.

[18] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. IEEE Access, 6, 35365-35381.

[19] S. Journal, R. Statistical, S. Series, and C. A. Statistics, "Algorithm AS 136?: A K-Means Clustering Algorithm Author (s): J . A . Hartigan and M . A . Wong Published by?: Wiley for the Royal Statistical Society Stable URL?: https://www.jstor.org/stable/2346830," vol. 28, no. 1, pp. 100–108, 2019.