# An Effective Spam Classification Filter As A Web Application Using Naïve Bayes Classifier

**Satyam Sagar, Piyush Kumar Shukla, Raju Baraskar**

**Abstract**: Due to the extensive growth internets consumers, the email has become a crucial mode for exchange of information across the globe whether it is for personal or business information as it is an appropriate and low-priced way for exchange of information. However, it is likely subjected to misemploy or abuse. The spam email (non-legitimate email) is one of the examples of this situation, which is a random posting of irrelevant emails to a very large number of recipients. So, spam emails have been a long-standing subject of security in computers. They are very threatening to both the computer user as well as the computer network. As the use of email in business communication increase exponentially, the need of the automatic email management system is also increased, such as email filter which can classify the email into the spam or legitimate mail, phishing email classifier, and grouping into multiple folders, etc. To solve this problem, we proposed a spam filtering method using Naïve Bayes classification Algorithm which can be implemented as a web application to filter incoming messages into spam and ham.

**Index Terms**: Spam, Email, Email Classification, Spam Filter, Spam Detection, Naive Bayes Classification, Filtering Web Application.

## 1. INTRODUCTION

In today's modern era of digitalization, communication plays a vital role in it may it be a formal or informal communication. The use of electronic mail (also known as email), have become increasingly popular. This growth of email communication leads to an unprecedented increase in the number of illegitimate emails (also known as spam), - 49.7% of email sent is spam. This is mainly due to the fact that the present or current spam detection methods lack accurate spam classifier[1]. Some spam is just plain text sometimes with a URL; some is cluttered with images and/or attachment. By analyzing its content spam typically would fall into the following several common categories: gambling, degrees/diploma, diet/weight loss, jobs/money mules, and phishing, scam and so on. These categories are rarely seen in a legitimate message and can be helpful to differentiate legitimate message from a spam message, which makes text-based classifiers to be used to filter out spam emails[2]. In 2017 the number of worldwide email users is estimated at approximately 3.7 billion. By the end of 2021, the number of worldwide email users will be over 4.1 billion, approximately half of the worldwide population uses email in 2017[3].

The spam is irritating in nature but it also consists conveyor malicious code that has intended to do harm to the system but also it has a deep effect on the network bandwidth, storage and also is the major reason for poor work rate and economic losses to various organizations. While the spammers earn more than 200 million per year from spam advertising.

According to[4] “A typical user receives about 40-50 emails per day from others”. An average user in a global organization consumed the majority of its time in the processing of emails for various purposes such as the exchange of information and sharing of resources. Therefore, the need for an efficient automatic email management system plays an important role in increasing the productivity of an organization or individual. Predominantly, the tool pre-owned for email management is an email classification filter. An email filter is a tool used to manage and automatically organized the flow of incoming email message. The classification of this message is based on various criteria such as senders address, subject, and content of a message

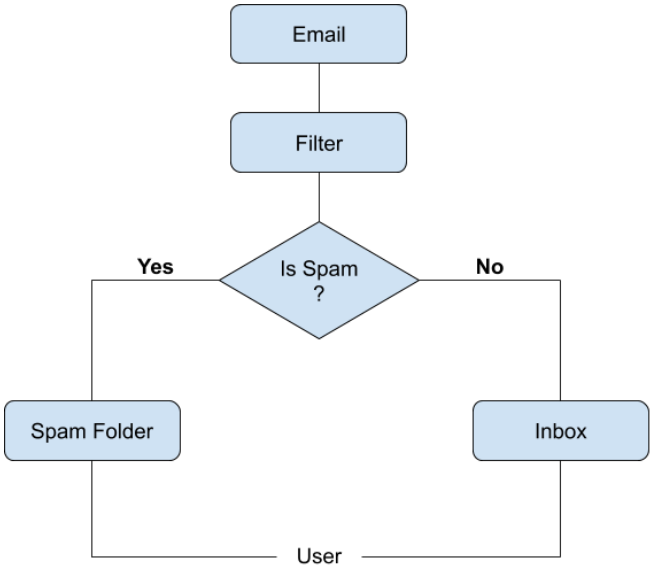


***Figure 1.1***. *Basic Working of a Spam Filter*

There are various classification algorithms/techniques applied in email classification into spam or ham, these techniques have both its advantages and disadvantages. This is due to the dynamic nature of spams. As the spammer is constantly developing new techniques to bypass filters, some of which includes word obfuscation and statistical poisoning[5]. So, the various studies are conducted on the creation of efficient and more accurate email classification filter using the number of classification algorithms over recent years.


- *Satyam Sagar is currently pursuing a post-graduate degree program in Computer Science and Engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, M.P. E-mail: satyamsagar30@gmail.com*
- *Piyush Kumar Shukla is Assistant Professor in the department of CSE, UIT-RGPV, Bhopal having 15 years of experience in the teaching field and also he has published many National and international papers in international Journals & Conferences.*
- *Raju Baraskar is Assistant Professor in Department of CSE, UIT-RGPV, Bhopal having 12 years of experience in the teaching field and also he has published many National and international papers in international Journals & Conferences.*

## 2 LITERATURE REVIEW

There are Various papers have been published showing various methods for email spam classification. Some of them are discus below. The proposed solution by[5] using a python algorithm (Naive Bayes) which joins with semantic & keyword-based, and machine learning algorithms to improve the efficiency of Naive Bayes contrasted with Spamassassin by more than two hundred percent. The calculation was also actualized and tried progressively conditions over the Internet. It additionally is shown that the calculation was reliably diminished the measure of spam messages misclassified as ham email. The calculation expanded the exactness of email arranging as well as demonstrated to be a profitable expansion to the current systems. It also takes care of the issue of content modifications by making an expansion to Naive Bayes, which can improve the multi-class expectation capacity and furthermore discovered that the new expansion improved ham classification because of the high review and accuracy rates. The author[6] uses a Naïve Bayes classification algorithm along with the Hidden Markov model. It proposed a way in which we can categorize email by considering only text part from the body of the message. Because in this paper, the author considers relative words and sentences features. After this, the result is compared and Hidden Markov Model (HMM) is used for classification because it gives better accuracy. The author uses a dataset of 5500 emails which contain 1500 important, 4000 spam email and this dataset from Enron email dataset. The author[2] proposed a spam filtering system using Naive Bayes classification and this system is deployed as a web-service which would consume the emails user uploads and give back the predicted probability that in what degree the given email is spam. This engine was achieved by Rest easy technology and consists of three phases to train prelabelled emails and then apply Naïve Bayes theorem to calculate email's spam. The Hadoop Map/Reduce framework is also integrated to the system to process the large volume of sample email and the pre-processing phase is also added while training, which will kick out some insignificant words, extract some typical features, and help improve the accuracy of email classification. It also focuses on the traits that can be helpful in improving the accuracy of the Naive Bayes classification such as mail header, blank "From" field, and the list a lot of address in the "To" field. Proposed solution by[7] where the author utilizes spam email classifier utilizing context-based email classification model as the fundamental calculation which is supported by the information-gain count to improve the efficiency of spam identification. In this, the procedure of email characterization starts with the pre-processing of email utilizing POS-tagger, then it removes a few email highlights to change email into the graph. So, email is grouped into the envelope(folder) with the agent graph with the most noteworthy match represent. This solution also utilizes the spam channel from the linger to strengthen the general precision of the framework. This examination result demonstrates the 100% precision in the spam classification of the email framework is as yet a neglected need and the handling time between utilizing a spam filter and not utilizing spam filter contrast inconsequential and also it is important to decrease the processing time in the email filtering. The paper by[8] a focus on the impact of the spam email received in the health care sector (such as nursing home, sickbay, and health care centers). For this experiment, the author collects the spam email dataset from the regional hospital and health care centers and some of the emails that are associated with the healthcare are from Tree 2007 corps. This paper focuses on the hybrid solution by combining two different email spam classification algorithms so that if a spam email is escaped from the first algorithm can be detected by the second algorithm. The above-proposed method is used to increase the accuracy of the system and scale down the false positive rate. This analysis demonstrates that the combination of DT (Decision Tree) and NB (Naïve Bayes) classification has the highest accuracy as compared to the other combination. The author[1] concentrates to improve the speed of spam filtering however much as could be expected while guaranteeing the rightness of spam filtering and proposes a quick content-based spam filtering algorithm along with fuzzy-SVM and k-means. In this, the k-means bunching algorithm is utilized to compress the data. This algorithm could bunch (or cluster) the data as indicated by the comparability level of data. At that point, the fuzzy support vector machine (FSVM) is utilized to prepare the classification model. The consequences of this investigation have demonstrated that the model could improve spam filtering algorithm from two parts of diminishing time utilization and expanding precision rate and it additionally demonstrates that with the expansion of compression ratio(γ), recall rate and accuracy rate of mail will increment. This is on the grounds that the fewer sample data, the less valid data contained and the classification precision will decrease. The author[9] proposed an ontology-based approach for spam email classification to provide spam filtering accuracy significantly. The model comprises of two levels of thinking for spam filtering was executed at the primary level a global ontology filter and a second level client tweakable ontology filter. The utilization of the global ontology filter appeared about 91% of spam filtered, which is practically identical without her techniques. The client tweakable ontology filter was made dependent on the particular client's experience just as the filtering system utilized in the global ontology filter creation.

**Table 1.**

| Author | Technique used | Advantages | Disadvantages |
|---|---|---|---|
| S. Peng et al.[5] | Naïve Bayes classification algorithm. | Most simple to implement and less complex. | The speed and accuracy of the system are less than the other system. |
| S. Wang et al.[1] | Fuzzy-SVM and k-means | The precision increases as the compression ratio increases. | Complex to implement and required large sample data is required for training. |
| Sebastian Romy Gomes et al.[6] | Naïve Bayes and Hidden Markov Model | The accuracy of the system improved. | The system is not capable to counter poison attack. |
| Seongwook Youn et al.[9] | Two-level Ontology-based classifier. | The system is suitable when the requirement of the classification model is customized according to the need of the user. | Need focus on the misclassified legitimate emails and also on the overfitting of the filter. |

| | | | |
|---|---|---|---|
| Weiweng Yang et al.[8] | Various combination of classification algorithms. | This experiment by the author gives various methods for classification | Performed poor in some case. |

## 3 PROCESS OF CLASSIFICATION

The classification process works in two phases, in the first phase the classification model or classification rules are builds and in the second phase, the classifier is used for the classification purpose on the testing dataset whereas the first phase uses training dataset for the setting up classification rule or classification model. Figure 3.1. displays the working of the classification system. In the first phase, the various process is applied to the raw data like data pre-processing, implementation of the classification algorithm and features extraction to build the classification model. In the data pre-processing the raw data is pre-processed to remove any kind of noise and other factors which can affect the quality of the dataset then, this data set is split into two parts i.e. training dataset and testing dataset. Then the classification algorithm is applied to the training data set to train the classification model. The classification model is nothing but the set of rules which determine which object goes to which class. Then the testing dataset is used to test the classifier.

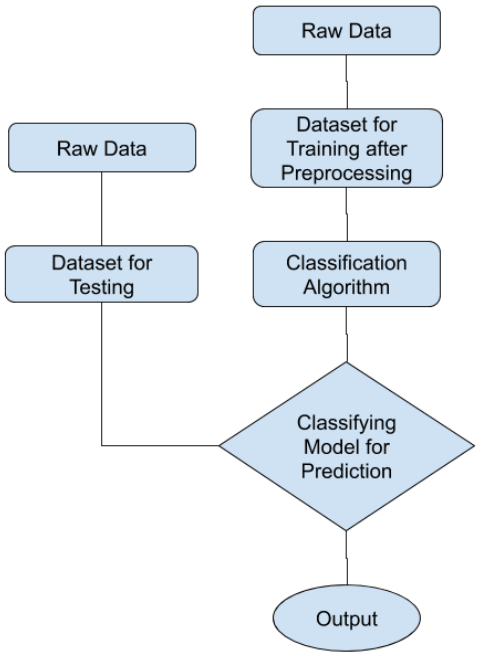


***Figure 3.1**. Basic Working of a Classifier.*

## 4 PROPOSED METHODOLOGY

During the development of the machine learning models, we mostly focused on generating a numerical prediction based on the sample test dataset. But to make the real-world application it is important to deploy these models on the server to use in the form of applications. But in my opinion, both the model generating and deployment part is equally important. As we have discussed the proposed solution in the above section 3.3 now, we will implement this proposed solution as a web application using Python's Micro Flask Framework for web development which takes new email message as input and predicts whether the given input is a spam or a ham as an output. The working of this system is described below.

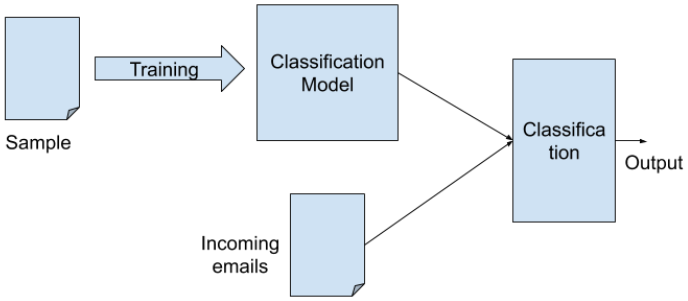


***Figure 4.1**. General Processing flow*

This system consists of two major parts. The first is to train the classifier with a dataset which consists of spam and ham emails and generation of the classification model. The second part is to deploy this model as a web service on a server. In the first part, we generate the classification model which defines the rules or criteria on which the classification of the email takes place. The process starts with importing the dataset in the system then the process of pre-processing on the dataset set takes place which removes various factors which degrades the quality of dataset like removal of punctuation, whitespace and converting upper case words into lower case words and replacing the email addresses, URLs, phone numbers, other numbers with the regular expressions.

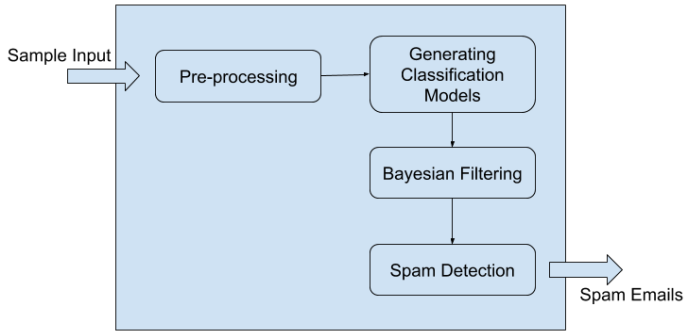


***Figure 4.2**. The architecture of the System.*

Then the features are extracted from the dataset. And after this for the training purpose, the dataset is split into two parts i.e. training part and test part. Now with the help of training dataset, we train our model and after the required amount of training, we can save our model in the .pkl file format so we did not have to train the model again and again. This process of generating a classification model is known as "persist model in a standard format", that is, models are persisted in a certain format specific to the language in development. The whole process in this first part can be done offline. The second is using our classifier model from the first part as a web application for this we have use python's Flask framework for the development of the web application. For this first, we install the Flask in our repository where our previous model is stored and then start building the webpage which takes email as an input and which can be named as "index.html" and another webpage which displays the output and can be named as "result.html". These two webpages are developed with the help

of HTML and CSS in the Flask framework. It is a good practice to begin the development in the virtual environment during the development of the web application which uses various libraries which can collide with the working of different application on your system.

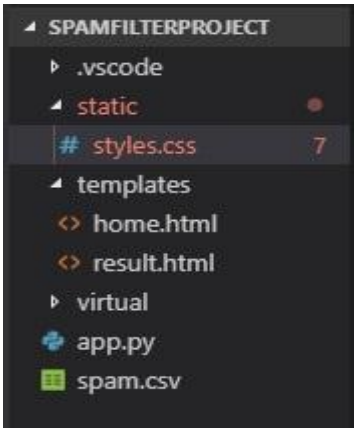


***Figure 4.3**. File Structure of the Project.*

After both, the parts of the application is completed and the application is running well in our local environment or system and we also have tested its working then we upload this complete application on the webserver for the public use.

## 5 RESULT

We conducted an experiment using the dataset from UCI Machine Learning Repository which contains more than 5000 labeled messages which are collected from mobile spam research. The impact of change in the preprocessing of Naïve Bayes classifier is shown in Table 2. The preprocessing plays an important role in deciding the efficiency of the filter.

**Table 2**. changes in accuracy score

| Changes in accuracy score with a change in pre-processing | | | |
|---|---|---|---|
| | With stemmers words | Without stemmers words | With message length as a feature |
| Accuracy Score | 0.984450 | 0.985048 | 0.982656 |

The above result shows the increment in the accuracy score when the stemmers words are removed from the messages. But there is a slight decrement in the accuracy score when the length of the message is also considered as the part of features set. This shows the importance of the pre-processing of datasets before generating the classification model. The confusion matrix shows the performance of the proposed method when used in the training phase. The result of this is represented in the form of a confusion matrix below in Table 3.

**Table 3.** Confusion Matrix

| | Spam | Ham |
|---|---|---|
| Ham | 232 | 1569 |
| Spam | 18 | 20 |

Table 4. Shows the precision and recall score which is achieved using the proposed method. These scores show the performance of the method.

**Table 4.** Precision and Recall Scores.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Spam | 0.93 | 0.92 | 0.92 | 252 |
| Ham | 0.99 | 0.99 | 0.99 | 1587 |

Thus from our following results obtained, the use of Naïve Bayes classifier can consider as a better option as it is simple to use and implement.

## 6 CONCLUSION

In this paper, we discuss the various techniques used for spam classification and also how the classification process works. From the conducted experiment we can say that the use of Naïve Bayes classifier is considered to be the best option for email classification as it has high accuracy and precision score. Due to the easy in implementation of naïve Bayes algorithm, it is suitable to use in creating the web application for classification of emails. There are various modules can be added in the current filter to increase such as optical image recognization for image classification and this application can be also implemented in the form of a mobile application for further use.

## REFERENCES

[1] S. Wang, X. Zhang, Y. Cheng, F. Jiang, W. Yu, and J. Peng, “A Fast Content-Based Spam Filtering Algorithm with Fuzzy-SVM and K-means,” Proc. - 2018 IEEE Int. Conf. Big Data Smart Comput. BigComp 2018, pp. 301–307, 2018.

[2] W. You, K. Qian, D. Lo, P. Bhattacharya, M. Guo, and Y. Qian, “Web service-enabled spam filtering with Naïve Bayes classification,” Proc. - 2015 IEEE 1st Int. Conf. Big Data Comput. Serv. Appl. BigDataService 2015, pp. 99–104, 2015.

[3] S. Radicati and Q. Hoang, “Email Statistics Report, 2017-2021,” 2017.

[4] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, “Email Classification Research Trends: Review and Open Issues,” IEEE Access, vol. 5, pp. 9044–9064, 2017.

[5] W. Peng, L. Huang, J. Jia, and E. Ingram, “Enhancing the Naive Bayes Spam Filter Through Intelligent Text Modification Detection,” Proc. - 17th IEEE Int. Conf. Trust. Secur. Prev. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust. 2018, pp. 849–854, 2018.

[6] S. R. Gomes et al., “A comparative approach to email classification using Naive Bayes classifier and hidden Markov model,” 4th Int. Conf. Adv. Electr. Eng. ICAEE 2017, vol. 2018-Janua, pp. 482–487, 2018.

[7] M. K. Chae, A. Alsadoon, P. W. C. Prasad, and S. Sreedharan, “Spam filtering email classification (SFECM) using gain and graph mining algorithm,” 2017 2nd Int. Conf. Anti-Cyber Crimes, ICACC 2017, pp. 217–222, 2017.

[8] Weiwen Yang and L. Kwok, “Comparison study of email classifications for healthcare organizations,” no. 2005, pp. 468–473, 2012.

[9] S. Youn, “SPONGY (SPam ONtoloGY): Email classification using two-level dynamic ontology,” Sci. World J., vol. 2014, 2014.