CS406 – Monsoon 2019 — Project (Spam Filter)

Arup Mondal

December 1, 2019

1 Introduction

In recent times, unwanted commercial bulk emails called spam has become a huge problem on the internet. Spam prevents the user from making full and good use of time, storage capacity and network bandwidth etc. It is also resulted to untold financial loss to many users who have fallen victim of internet scams and other fraudulent practices of spammers who send emails pretending to be from reputable companies with the intention to persuade individuals to disclose sensitive personal information like passwords, verification number and credit card numbers etc. The menace of spam email is on the increase on yearly basis and is responsible for over 77% of the whole global email traffic [1].

Machine learning methods of recent are being used to successfully detect and filter spam emails. This report presents a systematic review of some of the popular machine learning based email spam filtering approaches and compares the strengths and drawbacks of this machine learning approaches.

Though there are several email spam filtering methods in existence, the state-of-the-art approaches are discussed in this report. We explained below the different categories of spam filtering techniques that have been widely applied to overcome the problem of email spam [1].

- *Content Based Filtering Technique* is usually used to create automatic filtering rules and to classify emails using machine learning approaches, such as Naive Bayesian classification, Support Vector Machine, K Nearest Neighbor, Neural Networks.
- *Case Base Spam Filtering Method* extract the all emails from each user's email using collection model. Then pre-processing steps are carried out to transform the email using client interface, feature extraction, and selection, grouping of email data, and evaluating the process. And then machine learning algorithm is used to train data-sets and test them to decide whether the incoming mails are spam or non-spam.
- *Heuristic or Rule Based Spam Filtering Technique* used rules or heuristics to assess a huge number of patterns which are usually regular expressions against a chosen message to decide whether the incoming mails are spam or non-spam.
- *Previous Likeness Based Spam Filtering Technique* uses memory-based, or instance-based, machine learning methods to classify incoming emails based to their resemblance to stored examples. This approach uses the k-nearest neighbor (kNN) for filtering spam emails.

- *Adaptive Spam Filtering Technique* detects and filters spam by grouping them into different classes. It divides an email corpus into various groups, each group has an emblematic text. A comparison is made between each incoming email and each group, and a percentage of similarity is produced to decide the probable group the email belongs to [1].

The remainder of the report is organized as follows. Section 2 shows the methodology and Section 3 analyses the performance implications of different spam filter. Finally, conclude in Section 4.

2 Methodology

Majority of the email spam filtering methods uses text categorization approaches. Consequently, spam filters perform poorly and cannot efficiently prevent spam mails from getting to the inbox of the users. In this project mainly we focused the Random Forests (RF) algorithm to extract important features from emails, and classify the emails into either spam or non-spam. In this section we will review some of the most popular machine learning methods that have been applied to spam detection.

The necessary stages that must be observed in the mining of data from an email message can be categorised into the following:

- Pre-processing
- Tokenization
- Feature selection

K-Nearest Neighbors (kNN) classifier: [1] K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning, and it belongs to the supervised learning domain. In kNN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer). The best choice of k depends upon the data; generally, larger values of k reduces effect of the noise on the classification, but make boundaries between classes less distinct. If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

Naive Bayes (NB) classifier: Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

The fundamental Naive Bayes assumption is that each feature makes an:

- **independent:** we assume that no pair of features are dependent.
- **equal:** each feature is given the same weight.

where, C is class variable and X is a dependent feature vector (of size n) where $X = (x_1, x_2 ... x_n)$.

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(X|C) = \frac{P(C|X).P(X)}{P(C)} \tag{1}$$

Now, put a naive assumption to the Bayes' theorem, which is, independence among the features.

$$P(C|X) = \frac{P(x_1|C).P(x_2|C).P(x_3|C)...P(x_n|C)}{P(x_1).P(x_2).P(x_3)...P(x_n)} \tag{2}$$

In case of continuous data, we need to make some assumptions regarding the distribution of values of each feature. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_1|C)$ is below,

- **Gaussian Naive Bayes classifier:** In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution.
- **Multinomial Naive Bayes classifier:** Feature vectors represent the frequencies with which certain events have been generated by a multinomial distribution.
- **Bernoulli Naive Bayes classifier:** In the multivariate Bernoulli event model, features are independent Boolean's (binary variables) describing inputs

Support Vector Machine (SVM) classifier: [1] Support Vector Machines (SVM) are supervised learning algorithms that have been proven to perform better than some other attendant learning algorithms. SVM has find application in providing solution to quadratic programming problems that have inequality constraints and linear equality by differentiating different groups by means of a hyper-plane. Though the SVM might not be as fast as other classification methods, the algorithm draws its strength from its high accuracy because of its capacity to model multidimensional borderlines that are not sequential or straightforward. We used the binary C-SVM classifier which was explained in [1].

Decision Tree (DT) classifier: Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

In Decision Tree the major challenge is to identification of the attribute for the root node in each level. This process is known as attribute selection. In this we briefly discussed ***Gain Ratio*** to select the splitting attribute using the fallowing formula are,

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * log_2(\frac{|D_j|}{|D|})$$

$$Gain(A) = Info(D) - Info_A(D)$$

$$Info(D) = -\sum_{i=1}^{m} p_i * log_2 p_i$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} * Info(D_j)$$

Random Forests (RF) classifier: [1] Random forest (RF) is an example of ensemble learning approach and regression technique appropriate for solving problems that pertains to classifying data into groups. The algorithm carry out prediction through the use of decision trees. These decision trees are subsequently utilised for the task of predicting the group; this is accomplished by taking into account the selected groups of every distinct trees and the group that have the highest number of vote is taken as the result. Some of the strength of Random forests is that it usually have lesser classification error and superior f-scores compared to decision trees. RF produces several trees used for classification. The task of classifying a new data from an input vector begins by placing the input vector along each of the trees in the forest. Every tree will perform its classification which is often referred to as the tree ”votes” for that group. The forest decides which of the groups have the overall highest votes in the forest.

Multi-Layer Perceptron (MLP) classifier: A multilayer perceptron (MLP) is a class of feed-forward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called *back-propagation* for training.

In our model we used ReLU activation function. A major drawback of MLP model is that the training is slow as it takes a longer time to build compared to NB.

3 Result and Discussion

In this section, we shows the performance of the all explained spam filters which are described in section 2. In our experiments we shown that *Random Forests* gives higher accuracy i.e., 95.87% which is shown in the Figure 1 with the confusion matrix. Classification accuracy is one of the performance metrics for email spam classification. It is measured as the ratio of number of correctly classified instances in the test data-set and the total number of test cases. In spam filtering, false negatives mean that some spam mails were wrongly classified as non-spam and allowed to enter the user's inbox. False positive mean that non-spam emails were mistakenly classified as spam and moved to spam folder or discarded. For most users, erroneously classifying valid emails as spam can be very costly than receiving spam mails in their inbox. The false positive rate is also one of the performance metrics used in evaluating the effectiveness of email spam filter.

From the Figure 1, RF accurately predicts 563 instances out of 577 instances (563 spam instances that are truly spam and 14 not-spam instance that is really spam), and 320 instances are correctly predicted out of 344 (320 not-spam instances that are truly not-spam and 24 spam instance that is really spam). From our experiments it is clear that RF performed excellently in term of effectiveness and efficiency considering its classification accuracy.

All others spam classifier results are shown in the Appendix(5) section by the confusion matrix with the accuracy.

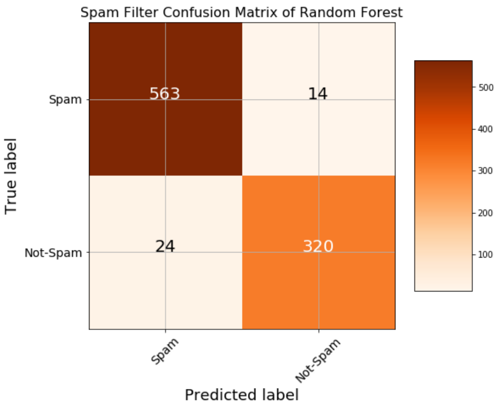

Figure 1: Random Forests Classifier with accuracy 95.87%

4 Conclusion

In this study, we reviewed machine learning approaches and their application to the field of spam filtering. Many of the existing email spam filtering techniques cannot effectively handle some of the spam emails. In this study, we shows Random Forests algorithm for effective and efficient email spam filtering. And evaluated the performance of RF algorithm on *Spambase Data Set* spam data-sets using accuracy.

References

1. Dada, Emmanuel Gbenga, Joseph Stephen Bassi, Haruna Chiroma, Adebayo Olusola Adetunmbi, and Opeyemi Emmanuel Ajibuwa. ”Machine learning for email spam filtering: review, approaches and open research problems.” Heliyon 5, no. 6 (2019): e01802.

5 Appendix

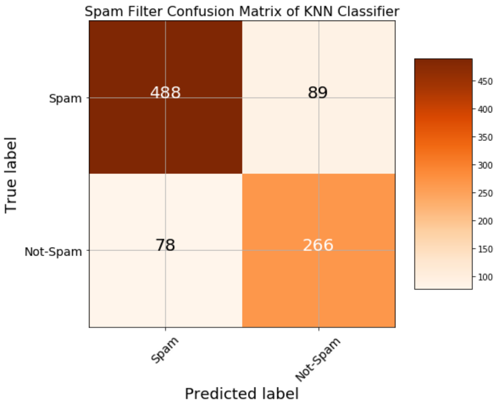

Figure 2: K-Nearest Neighbour Classifier with accuracy 81.86%

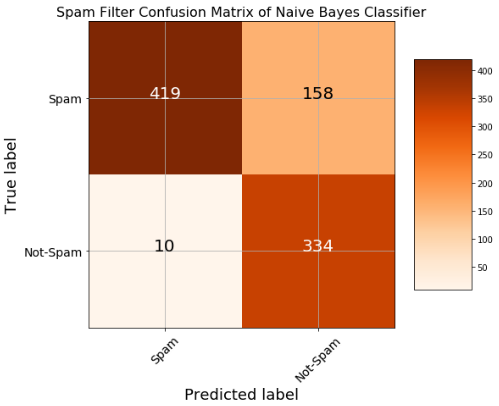

Figure 3: Naive Bayes Classifier with accuracy 81.75%

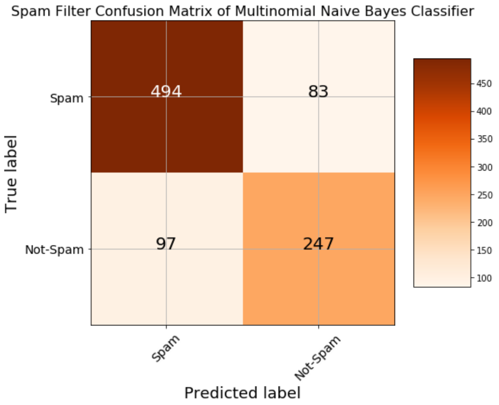

Figure 4: Multinomial Naive Bayes Classifier with accuracy 80.45%

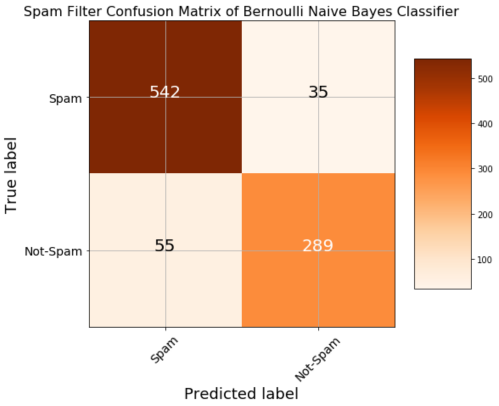

Figure 5: Bernoulli Naive Bayes Classifier with accuracy 90.22%

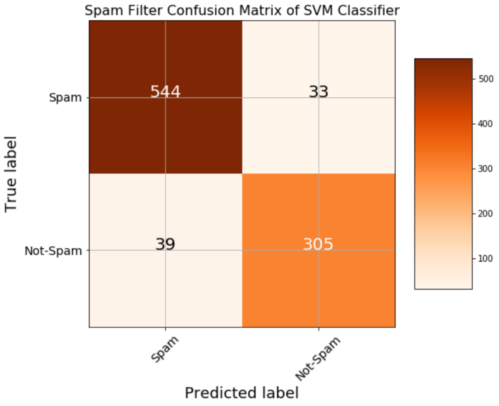

Figure 6: Support Vector Machine Classifier with accuracy 92.18%

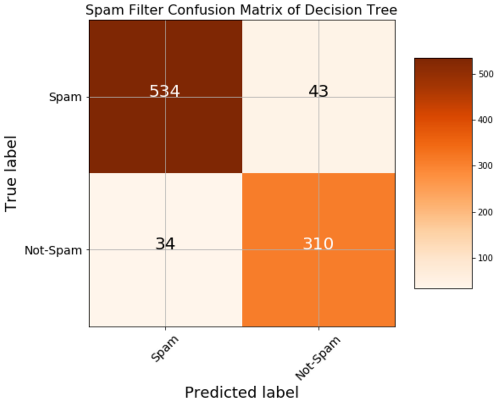

Figure 7: Decision Tree Classifier with accuracy 91.63%

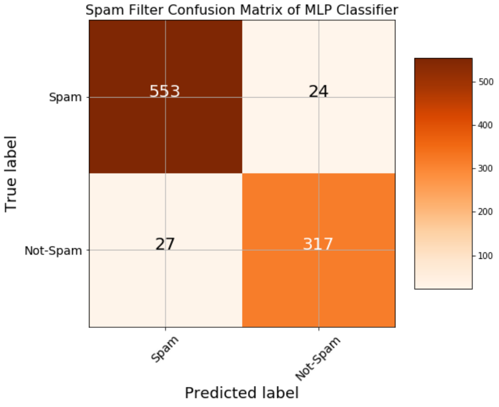

Figure 8: Multi-Layered Perceptron Classifier with accuracy 94.46%