

Multiple Machine Learning Algorithms for Spam Mail Detection

Simou Chen *

Department of Science and Engineering, University of Manchester, Manchester, United Kingdom

* Corresponding author email: simou.chen@student.manchester.ac.uk

Abstract. Due to several issues including clogging mailboxes and obscuring critical personal correspondence caused by a large amount of spam mail, spam filtering application based on machine learning models has become more and more essential. This paper gives a comprehensive evaluation of 4 classic machine learning models including the Naïve Bayes classifier, support vector machine, neural network, and the decision tree in the task of spam email evaluation. In order to make fair comparisons between the algorithms, they were all built by learning from the same training dataset and asked to make predictions on the same testing dataset. More specifically, the Naïve Bayes classifier uses the conditional probability of the Naïve Bayes theorem and consumes the words that appear independently in the emails. The decision tree algorithm keeps recursively splitting the dataset to build a predictive theory. Support vector machine is able to make predictions by transforming the training dataset to a linear equation in n-dimensions. And neural networks can recognize the non-linear relationships between the inputs and outputs by constructing the front, hidden, and output layers. It is observed from the result that all machine learning models have successfully classified spam emails with high testing accuracy which are all above 90%. More specifically, the neural network achieves the highest testing accuracy followed by the support vector machine and decision tree. While the Naïve Bayes classifier does not perform well enough as it must assume the appearance of words happens independently.

Keywords: Spam Mail Detection; Support Vector Machine; Naïve Bayes; Decision Tree; Neural Network.

1. Introduction

A Spam filtering's job is to automatically remove undesirable mail from a user's mail stream according to predetermined criteria. In order to distinguish between incoming emails that are safe for the user to open and those that are spam, spam filtering application uses some of the machine learning models or algorithms within the system that is controlled by a series of protocols. Email spam, also referred to as junk email, spam mail, or simply spam, is unsolicited messages sent in bulk by email (spamming) [1]. These unwanted emails have already created several issues, including clogging mailboxes, obscuring critical personal correspondence, wasting network capacity, and taking users' time and energy to filter through. So, implementing spam filtering is essential for any business. In addition to keeping spam out of email inboxes, spam filtering makes corporate communications more effective and helpful by ensuring that they are only used for the purposes for which they were intended. According to a series of surveys conducted by Sabrina I. Pacifici that email spam costs businesses \$20.5 billion every year and as many as 85% of all organizations have been targeted by phishing scams in 2021 [2]. Therefore, it is critical to apply an intelligent and effective spam filter into practice.

In the early field of spam filtering evaluation, the task of mail filtering has been successfully applied to a few supervised machine-learning algorithms including Naïve Bayes classifier, support vector machines, neural networks, and maximum entropy model. For example, the article reported by Deshapand et al. that they conclude the Naïve Bayes technique is efficient and can make a positive contribution [3]. In addition, Drucker et al. studied the implementation of Support Vector Machines (SVM) in classifying e-mail in 1999 [4]. They found that the SVM had acceptable test performance in terms of accuracy and speed. Moreover, Aman Goel also pointed out that the neural network is powerful enough to be used to learn from the characteristics of the data set in order to solve classification problems [5]. Although all these strategies appear enticing, it is challenging to fairly compare their performances based on published findings. For one thing, cross-classifier evaluation

was challenging because the results presented in earlier publications were based on distinct corpora so different algorithms were provided with totally different training datasets which may lead to inaccuracy when it comes to cross-validation. Another factor is that each algorithm's testing dataset size may vary significantly as a result of the unique dataset. A big testing dataset may favorably affect the performance of some algorithms while hindering the performance of others. With the combination of these two factors, it seems extremely hard to make fair comparisons between different algorithms.

In this regard, this paper will try to bridge this gap, by designing a more scientific experiment for comparisons of various spam filtering algorithms. More specifically, this paper will focus on the four of the commonest machine learning algorithms which are the Naïve Bayes classifier, neural networks, SVM, and the decision tree which has hardly even been considered to compare with other techniques. The training dataset and testing dataset would be obtained from open-source dataset and used for every algorithm. As a result of it, the knowledge they would gain from and the relationship they would identify can vary to an as small extent as possible. Moreover, this study will also seek the unique features of the specialized algorithm and avoid the factors that can affect the classifiers' performance.

The rest of this paper is structured as follows: In section 2, the dataset description and preprocessing will be overviewed. In section 3, the machine learning models will be outlined along with the major characteristics found in each stage. In section 4, a comparison of testing accuracy of different machine learning models would be given. Finally, a conclusion would be formulated in section 5.

2. Dataset Description and Preprocessing

Implementations in this project utilized an email dataset that was made available through open source. The original dataset consisted of 5, 572 rows where every single row represented an individual event which is the email message in this project. In addition, the number of spam messages is 747, and the number of ham messages which means non-spam emails in this case is 4, 825.

Table 1. The sample data from the original dataset

v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN

Due to the obvious evidence of inconsistency, missing data, and the presence of some useless data in the original dataset, the first step of the preprocessing was to apply data cleaning and mining for improving the overall data quality. It is obvious to see that in the Table 1 above that the row name 'V1' and 'V2' needed to be changed to some meaningful names that can specify all values of the column, which would be "label" and "text" respectively. And the as the rest 3 columns did not contain valid data for applying algorithms so they ought to be removed as they may give an incorrect view of the overall figure of data and tend to affect the observation. Then the next step is to modify some contents within the cleaned dataset for some specific algorithms. So that in this step, an additional column called "label_tag" had been added for decision tree and neural network models since they are supervised learning algorithms. More specifically, if the value of the 'label' is 'ham', then that of the 'label_tag' should be 0 for the same row, and 1 otherwise which is illustrated in the Table 2 below. While the Naïve Bayes classifier and SVM models could just utilize the cleaned dataset.

In the final step of the data processing, the cleaned version of the dataset was randomly split into 2 datasets as in machine learning, one for the training and another for the testing. In this project, the size of the training dataset was 75% of the original dataset and 25% for the testing dataset.

Table 2. The sample data from the preprocessed dataset

label	text	label tag
ham	swqrt how u dey,hope ur ok, tot about u 2day.l...	0
ham	Ok da, i already planned. I wil pick you.	0
spam	Urgent! Please call 0906346330. Your ABTA comp...	1
ham	Sorry, I'll call later in meeting	0

3. Machine Learning Models

3.1 Neural Network

Neural networks, also known as Artificial Neural Networks (ANNs) which has been applied in many tasks [6-8], represent the core of deep learning techniques and are a subset of machine learning [9]. Input, one or more hidden layers, and output layers make up a node layer in an artificial neural network. Each node, or artificial neuron, is interconnected with others and comes with a weight and threshold. Any node whose output rises above the specified threshold value is activated and starts sending information to the top layer of the network. Otherwise, no data is sent to the next tier of the network. Moreover, the neural networks can learn from the training data in order to increase the accuracy of the result.

In this project, the first step was to develop a predictive theory so that the network can learn the characteristics of spam emails based on this criterion. Since the neural networks can only read numbers like integers and floats, the content of mails needed to be converted into numbers. During this stage, it was found that there were 12475 unique words which can be used to build up the vocabulary vector. The vocabulary vector would come as a dictionary form which contained a series of {word: count}. When a word appears more than once in a text, the individual word counts in that text are updated from 0 to a number. This implies that when assessing whether a text is spam or not, terms with a larger count can be given more weight.

Afterwards, the input layer function took the vector and mails as input and did the words counting task. The hidden layer functions then would perform nonlinear transformations of the input text. In the end, the output layer function was designed to obtain the desired prediction. More importantly, the epoch number also had to be considered. A neural network is trained for one cycle during an epoch using all of the training data. In this project, the epoch was completed with iterations. Finally, a complete neural network classifier was built with the combination of the input layer, hidden layer, and the output layer. The classifier is expected to make predictions with epoch number from 1 to 9 for the testing dataset.

3.2 Support Vector Machine

In the field of the machine learning, SVM is a classic supervised learning models for classification tasks. Given a set of training dataset, the element within the set has already been marked with a specific type or category. An SVM model is expected to map the training dataset in an n-dimensional space to get larger gaps between each element from these 2 different categories. Afterwards, whenever a new input element is passed to this system, the model just mapped it to the same space. Then the classification can be performed by finding the hyper-plane that differentiates the two categories.

In this project, the SVM algorithms would be performed by Support Vector Classification (SVC) from the sklearn library. Before using the SVC methods in the implementation, the kernel function was used to provide the “window” space to manipulate data via transforming the training dataset to a linear equation in n-dimensions. In this case, Radial Basis Function (RBF) kernel was utilized as shown in the formula below.

$$K(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) \quad (1)$$

Where $||x - x'||$ is the Euclidean distance between x and x' . Finally, the classifier could fit in the training dataset and is expected to make valid predictions.

3.3 Decision Tree

In order to describe options and their potential results, such as utility, resource costs, and chance event outcomes, a decision assistance tool known as a decision tree uses a tree-like architecture [10]. This is one way to demonstrate an algorithm that only employs conditional control statements. Decision Tree can also be considered or used as a supervised learning method that divides space of outcomes into a series of regions and make a prediction for new input.

In this project, the classifier was made to keep recursively splitting the dataset to build a predictive theory. However, the over-fitting had to be avoided since the implementation could be too tight with the dataset. Eventually, a method called “DecisionTreeClassifier” could be used via importing the sklearn library to carry out the prediction.

3.4 Naive Bayes Classifier

Naïve Bayes classifier is a very popular and classic machine learning model for classification problems. Typically, the model used a series of features to categorize the element. The concept of Naïve Bayes Classifier is simple and straightforward which is to categorize element via using the conditional probability of naïve Bayes theorem.

$$P(S|E) = \frac{P(E|S)P(S)}{P(E|S)P(S) + P(E|\neg S)P(\neg S)} \quad (2)$$

Where the $P(S|E)$ is the probability that the message is spam given the event occurred. $P(S)$ is the prior probability that a message is spam. $P(\neg S)$ is the prior probability that a message is not spam. $P(E|S)$ is the probability that event E occurs in a spam email. $P(E|\neg S)$ is the probability that event E occurs in non-spam emails.

The study calculated the probability of a single word to be spam first, and then multiply every single word within an email by consuming those words appear as independent events.

4. Results and Discussion

Table 3. The performance of different models used in this study

Model	Performance (Testing Accuracy)
SVM	0.9740
Decision Tree	0.9605
Naive Bayes Classifier	0.9411
Neural Network	0.9852

After implementing different models using the testing dataset, the performance of these models based on the testing accuracy of the spam classification can be obtained. As it is shown in Table 3, all the models had shown fairly high accuracy which was all above 90%. By comparison, it is evident to notice that the model based on a neural network had made tremendous success in categorizing spam emails from the testing dataset. The accuracy was 0.9852 which was higher than that of other models. However, the Naïve Bayes classifier had the worst performance in this application, with an accuracy of 94.1%.

From the Table 3, it can be concluded that machine-learning models have successfully classified spam emails. Especially when it comes to the neural network, it has achieved extremely high accuracy. The reason behind it could be that the neural network is able to learn and construct non-linear relationships between inputs and outputs. Afterward, it can make predictions on the unknown data to

make the whole model more generalized. As a result of it, the accuracy of the neural network was the highest. Moreover, the experiment also showed that SVM and decision trees are top performers in spam filtering tasks, which are better than Naïve Bayes in this case. These 2 algorithms shared some common features that both of them worked well by separating the margin or gap between classes and SVM is actually more effective when performing in the higher dimensions. The main limitation of the Naïve Bayes classifier in this experiment was that the algorithm had to assume that the appearance of words happened independently which is almost impossible in real life. And this might be the reason that it performed worse than other algorithms in this experiment.

5. Conclusion

In this paper, a comprehensive evaluation of 4 classic machine learning models in the task of spam email classification was provided. They were all built by learning from the same training dataset and made predictions on the testing dataset. Through the experiments conducted, it was found that all models performed fairly good with testing accuracy higher than 90%. In particular, the neural network model is the most successful algorithm via comparison of results, the possible reason could be the ability to learn and build non-linear relationships between inputs and outputs. And Naïve Bayes classifier did not perform as well as others due to the limitations of assumptions of independent feature. In addition, SVM and decision tree both worked well by separating the gaps between categories. In the future, the experiments can be improved by using larger datasets to acquire more diverse spam features and considering some more common spam traits, such as URLs, email addresses, and phone numbers.

References

- [1] Wikipedia, Email Spam, Available at: https://en.wikipedia.org/wiki/Email_spam, 2022.
- [2] Sabrina I. Pacifici. What's On the Other Side of Your Inbox - 20 SPAM Statistics for 2022. Available at: <https://www.bespacific.com/20-spam-statistics-for-2022/>, 2022.
- [3] Vikas P. Deshpande, Robert F. Erbacher, Chris Harris. An Evaluation of Naïve Bayesian Anti-Spam Filtering Techniques. IEEE, 2007.
- [4] H. Drucker, Donghui Wu, V.N. Vapnik. Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 2007.
- [5] Aman Goel. Spam Detection Using Neural Networks in Python. Available at: <https://medium.com/emergent-future/spam-detection-using-neural-networks-in-python-9b2b2a062272>, 2016.
- [6] Meunier, Laurent, et al. A dynamical system perspective for Lipschitz neural networks. International Conference on Machine Learning. PMLR, 2022.
- [7] Q. Yu et al. Semantic segmentation of intracranial hemorrhages in head CT scans. In 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS) (pp. 112-115). IEEE. 2019.
- [8] Guo, Jianyuan, et al. Cmt: Convolutional neural networks meet vision transformers. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
- [9] IBM Cloud Education. Neural Networks. Available at <https://www.ibm.com/cloud/learn/neural-networks>, 2020.
- [10] Wikipedia, Decision Tree, Available at: https://en.wikipedia.org/wiki/Decision_tree, 2022.