”Spam” or ”Ham”: Spam Filtering System using Machine Learning Algorithms

Ephrathah Gebremichael
Computer Science Department
College of Wooster
Wooster, OH
egebremichael24@wooster.edu

Turbat Enkhtur
Computer Science Department
College of Wooster
Wooster, OH
tenkhtur23@wooster.edu

Abstract

Spam filtering is a critical issue in today’s digital communication landscape, where the prevalence of unsolicited electronic messages disrupts not only efficient communication but also poses significant threats to information security and user privacy. To address this challenge, this paper investigates the effectiveness of various machine learning algorithms in spam filtering, focusing on Naive Bayes classifier, k-Nearest Neighbors (kNN), and Support Vector Machine (SVM) classifiers. Using the spam dataset, we employ advanced text preprocessing techniques and feature extraction methods, such as Term Frequency-Inverse Document Frequency (TF-IDF), to enhance the performance of our classifiers. Our results show that the SVM classifier outperforms the other classifiers, achieving an accuracy of 97.6%, followed by the Naive Bayes classifier at 96.5%, and the kNN classifier at 91.9%. These results emphasize the importance of feature engineering and preprocessing in achieving accurate classification and highlight the potential for using advanced machine learning techniques in developing effective spam filtering systems.

Introduction

Over the years, there has been a rapid increase in the number of spam messages due to the growth of digital communication. Spam poses serious threats to information security, user privacy, and communication efficiency. Therefore, effective spam filtering techniques have become a crucial area of research in machine intelligence. This research paper aims to evaluate and compare the effectiveness of existing approaches for filtering spam messages. The study examines different machine learning algorithms and data pre-processing techniques commonly used in spam filtering systems, including supervised and unsupervised learning, to identify the most efficient and accurate approach. The focus of this research is on analyzing the performance of existing techniques and analyzing their results. Furthermore, we investigate the importance of feature extraction and selection in enhancing the performance of our spam filtering model. The choice of appropriate features plays a crucial role in determining the effectiveness of the filtering system. This project analyzes the effectiveness of machine learning-based approaches in spam filtering and achieving accurate classification.

Background

Over the years, various techniques have been proposed to address the unsolicited electronic messages problem, ranging from rule-based approaches to machine learning-based methods (Pugliese, Regondi, and Marini 2021). Machine learning-based approaches have the advantage of automatically learning from data, enabling them to adapt to new spamming strategies without the need for manual intervention. Over time, various machine learning algorithms have been proposed and evaluated for this purpose, including SpamCampaignAssassin (SCA) which is an unsupervised learning method and Naïve Bayes, Support Vector Machine (SVM), and k-Nearest Neighbors (k-NN) which are supervised learning methods.

The SCA system is an unsupervised spam filtering scheme that can identify spam campaigns without requiring any training data. It operates online and generates campaign signatures to detect spam (Qian et al. 2010). Contrary to the SCA system’s unsupervised approach the naïve Bayes, k-NN, and SVM use both training and testing data. The naïve Bayes algorithm works by estimating the conditional probabilities of the features given a class, and then using the Bayes theorem to calculate the probability of the class given the features (Mitchell 1997). Despite the simplicity and the strong independence assumption, the Naïve Bayes algorithm often performs well in practice, particularly in text classification tasks such as spam filtering, where the features are typically words or word frequencies in a document. It can be calculated by:

$$P(C_k|\mathbf{x}) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i|C_k)}{\sum_{j=1}^{m} P(C_j)\prod_{i=1}^{n} P(x_i|C_j)}$$

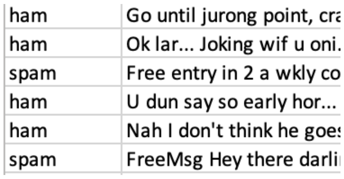
where $P(C_k|\mathbf{x})$ represents the posterior probability of class C_k given the feature vector $\mathbf{x}$, $P(C_k)$ is the prior probability of class C_k, $P(x_i|C_k)$ denotes the conditional probability of feature x_i given class C_k, and the denominator is the evidence term used for normalization (Mitchell 1997). The algorithm selects the class with the highest posterior probability as the predicted class for the given instance.

Another algorithm that can be used is the k-NN algorithm. The kNN algorithm is based on the assumption that similar items are likely to be grouped together. Given a new instance to be classified, the algorithm looks for the K nearest neighbors in the training set and assigns the class of the majority of these neighbors to the new instance (Mitchell 1997). When a new email arrives, the algorithm calculates the distance between the new email and all the emails in the training set. While kNN is one possible algorithm that can be used for spam filtering, another popular option is Support Vector Machines (SVM). Unlike kNN, SVM is a parametric algorithm that tries to find a hyperplane that separates the data into different classes. SVM operates in two main steps: constructing the optimal hyperplane and classifying new instances. The algorithm is particularly effective in high-dimensional feature spaces and is widely used in various applications, including spam filtering, image recognition, and bioinformatics.

These various algorithms can result in different levels of accuracy. While some algorithms may perform well in identifying certain types of spam emails, others may struggle to detect them. According to Qian et. al, SCA employs a text-mining technique on email bodies to generate campaign signatures, which, in one dataset, covered over 80% of spam messages. Remarkably, the detection accuracy of SCA was found to be comparable to supervised anti-spam solutions (Qian et al. 2010). Similarly, studies using different types of supervised learning methods show promising results. A study conducted in 2017 shows that the Naïve Bayes and K-Nearest Neighbor classification methods are able to detect spam and ham contents with 82% and 71% accuracy respectively (Pinandito et al. 2017).

The Spam Data Set

The spam dataset is a widely used collection of emails that has been compiled for the purpose of spam filtering and other text classification tasks. It consists of a set of emails that have been labeled as either ”spam” or ”ham” (non-spam) by human annotators. This paper will be using a version provided by the University of California, Irvine (UCI) Machine Learning Repository. This dataset contains 5,572 emails, of which 1,813 are labeled as spam and 3,759 are labeled as ham. A preview of the dataset is shown in Figure 1 below. The emails were collected from a variety of sources and cover a range of topics, including personal correspondence, business communication, and advertising. For this study we will be labeling spam as 1 and otherwise 0.



ham	Go until jurong point, cra
ham	Ok lar... Joking wif u oni.
spam	Free entry in 2 a wkly co
ham	U dun say so early hor...
ham	Nah I don't think he goes
spam	FreeMsg Hey there darli

Figure 1: Preview of the spam dataset

Methodology

In this study, we aim to develop an efficient spam filtering system by implementing and comparing the performance of three machine learning algorithms—Naive Bayes classifier, k-Nearest Neighbors (kNN), and Support Vector Machine (SVM). We begin by preprocessing email texts, which entails converting characters to lowercase, removing digits and punctuation marks, tokenizing the text into words, eliminating stop words, and applying stemming techniques. We then employ the Term Frequency-Inverse Document Frequency (TF-IDF) method to transform preprocessed email texts into feature vectors, a process that gauges the significance of words in a document by factoring in their frequency within the document and their rarity across a collection of documents (Ramos and others 2003).

With the TF-IDF technique, words are represented as high-dimensional sparse matrices, where each row corresponds to a document and each column represents a unique word in the vocabulary. Matrix elements contain the respective TF-IDF values, indicating the importance of a word in a particular document, thereby facilitating the conversion of text data into numerical features for machine learning algorithms. Next, we divide the dataset into training and testing sets, allocating 80% for training and 20% for testing. For each algorithm—Naive Bayes classifier, kNN, and SVM—we train a model using the training set and evaluate its performance on the test set. Our primary performance metric is the model’s accuracy, defined as the ratio of correctly classified samples to the total number of samples. By comparing the accuracy of these three models, we can determine the most effective algorithm for spam filtering within our dataset, which serves as a valuable resource for selecting the optimal machine learning algorithm for spam filtering applications.

Results

After examining a variety of classifiers, we observed a diverse range of outcomes, highlighting the distinct capabilities of each method. The first method that we looked at was the naive Bayes classifier. The basic implementation of naïve Bayes uses raw word frequency counts as features. In this approach, we preprocessed the email text by converting characters to lowercase, removing digits and punctuation marks, and tokenizing the text into words. We then trained the Naive Bayes classifier by calculating the word frequencies in spam and ham emails and classify an email as spam or ham based on the ratio of spam and ham word frequencies.

This basic implementation served as a baseline for comparison, with an accuracy of 18.8% (this can be seen in Figure 2 below). The confusion matrix shows that the classifier misclassified some of the ham emails (34 were misclassified as spam) and had a higher number of false positives for spam emails (871), resulting in a lower precision score for spam classification. To improve the classifier we incorporated advanced text preprocessing techniques and Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction. In this approach, we preprocessed the email text by removing stop words and applying stemming, in addition to the steps in the basic method. We then transformed the preprocessed email texts into feature vectors using the TF-IDF technique and trained a Multinomial Naive Bayes classifier using these features. This improved method demonstrates how removing stop words, applying stemming, and using TF-IDF features can significantly enhance the classifier's performance in spam filtering tasks, achieving an accuracy of 96.5% which can be seen in Figure 3 below. We can observe from the figure that the improved model only misclassified 39 instances as opposed to the initial model's 905 instance misclassification.

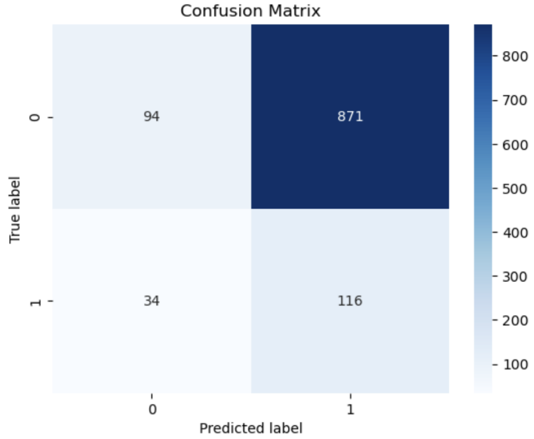


Figure 2: Confusion matrix for initial Bayes model

We also implemented and analyzed the SVM classifier. We used the CountVectorizer class from scikit-learn to transform the email texts into numerical feature vectors based on word frequency counts. This transformation enabled the SVM classifier to work with the textual data. We then trained an SVM model using the Radial Basis Function (RBF) kernel on the transformed training set. Finally, we evaluated the classifier's performance on the test set and obtained an accuracy of 97.6%, which demonstrates the effectiveness of the SVM classifier in spam filtering tasks. From the confusion table, in Figure 4 below, we can see that this method only misclassified 26 instances. The high accuracy

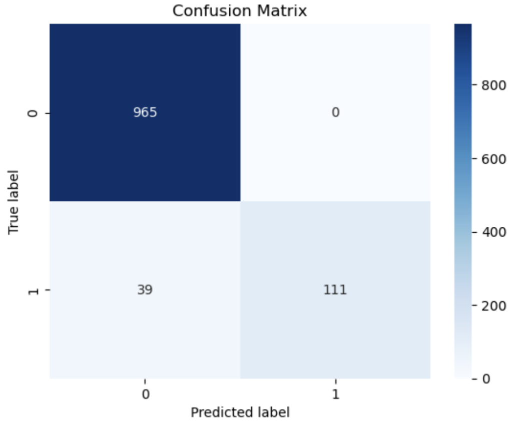


Figure 3: Confusion matrix for improved Bayes model

achieved with the SVM classifier can be attributed to its ability to handle high-dimensional feature spaces and its robustness to the presence of irrelevant features.

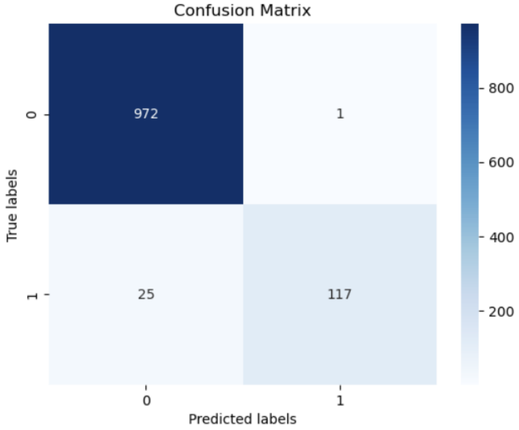


Figure 4: Confusion matrix for SVM

Furthermore, we also explored the k-Nearest Neighbors (kNN) algorithm classifier as a method for spam filtering. We used the CountVectorizer class from scikit-learn to transform the email texts into numerical feature vectors based on word frequency counts. This transformation enabled the kNN classifier to work with the textual data. We then trained the kNN model with a chosen value of k (the number of neighbors) on the transformed training set. Finally, we evaluated the classifier's performance on the test set and obtained an accuracy of 91.9%. This is demonstrated in Figure 5 below with only 90 instances misclassified.

Conclusions and Future Work

Our results showed that the SVM classifier achieved the highest accuracy of 97.6%, followed by the improved Naive

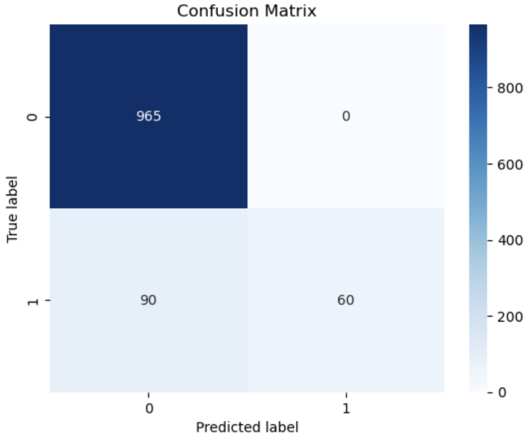


Figure 5: Confusion matrix for kNN

Bayes classifier at 96.5%, and the kNN classifier at 91.9%. These results exhibited superior accuracies compared to prior studies, where the naïve Bayes and kNN algorithms achieved accuracies of 82% and 71% respectively. The comparative analysis of the three classifiers demonstrated that the SVM classifier is the most effective method for spam filtering among the investigated algorithms. Its ability to handle high-dimensional feature spaces and its robustness to the presence of irrelevant features make it particularly well-suited for text classification tasks.

The Naive Bayes classifier's performance improvement from 18.83% to 96.50% after incorporating advanced text preprocessing techniques and using TF-IDF features highlights the importance of proper text preprocessing and feature extraction in text classification tasks. The kNN classifier, although not as accurate as the SVM and Naive Bayes classifiers, still offers competitive performance, making it a viable option for spam filtering when computational resources or training time are limited.

In conclusion, this study demonstrates the effectiveness of machine learning algorithms, particularly the SVM classifier, in spam filtering tasks. Future research could explore other machine learning algorithms, fine-tune hyperparameters, and investigate the use of ensemble methods or deep learning techniques to further improve spam filtering performance. Additionally, examining the impact of different feature extraction methods and incorporating additional features, such as email metadata, could potentially enhance the performance of spam classifiers.

References

Mitchell, T. M. 1997. *Machine learning*, volume 1. McGraw-hill New York.

Pinandito, A.; Perdana, R. S.; Saputra, M. C.; and Az-zahra, H. M. 2017. Spam Detection Framework for Android Twitter Application Using Naïve Bayes and K-Nearest Neighbor Classifiers. In *Proceedings of the 6th International Conference on Software and Computer Applications*, ICSCA '17, 77–82. New York, NY, USA: Association for Computing Machinery.

Pugliese, R.; Regondi, S.; and Marini, R. 2021. Machine learning-based approach: Global trends, research directions, and regulatory standpoints. *Data Science and Management* 4.

Qian, F.; Pathak, A.; Hu, Y. C.; Mao, Z. M.; and Xie, Y. 2010. A Case for Unsupervised-Learning-Based Spam Filtering. *SIGMETRICS Perform. Eval. Rev.* 38(1):367–368.

Ramos, J., et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, 29–48. Citeseer.