Naive Bayes classifiers for spam detection

An ensemble method to efficiently tackle spam

Giulio Lanza, Giaime Paolo Pes

20 maggio 2024

1 Abstract

The detection of spam messages remains a critical challenge in maintaining secure and efficient communication systems. This study evaluates the performance of various Naive Bayes classifiers and ensemble methods in identifying spam within a diverse SMS dataset. Utilizing the SMS Phishing Dataset for Machine Learning and Pattern Recognition, the research highlights the efficacy of Naive Bayes classifiers, specifically the Bernoulli and Multinomial models, in accurately detecting spam messages. The study further demonstrates that ensemble methods, particularly Weighted Voting, significantly enhance spam detection performance, achieving an accuracy of 97.68%, specificity of 90.95%, and sensitivity of 99.07%. These results underscore the potential of combining multiple classifiers to improve robustness and reliability in spam filtering.

2 Introduction

The release of the Enron Corpus in 2004, a database of almost 600,000 emails from the Enron Corporation, allowed a surge in studies on spam email detection. Researchers aimed to develop classification tools to distinguish spam emails from non-spam emails, referred to as “ham” in the literature jargon. Among these studies, "Spam Filtering with Naive Bayes - Which Naive Bayes?" by Metsis et al. (2006) has been particularly influential. It processed, analyzed and made available a subset of 17,171 spam and 16,545 non-spam ("ham") email messages, totaling 33,716 emails. The paper compares the performance of different Naive Bayes classifiers on tokenized words, evaluating them in terms of, among others, accuracy, specificity, and sensitivity. Finally, it simulates a training phase on earlier timeframes to assess predictive effectiveness on later timeframes, aiming to evaluate how well the classifiers perform as the email vocabulary changes over time.

Research on mobile text messages and SMS has historically been rarer and more repetitive. The biggest challenge has been collecting databases of individuals’ private chats, which is why the most widely used database remains the one by Almeida et al. (2011). Generally speaking, Ahmed et al. (2014) showed that algorithms based on Naive Bayes classifiers perform considerably better than those in previous works, such as in email classification. Several reasons can be attributed to this difference in accuracy. For example, the sender’s number often remains the same (possibly because it costs more to obtain a new number than to create a new email address). However, this paper focuses on language recognition rather than these factors.

Firstly, the tendency for text messages to be more direct and intimate means that ham messages frequently contain colloquial stock phrases, minimal sentences, dialectal words, or abbreviations typical in messaging only. Conversely, spam messages, often sent by companies or entities pretending to be companies or government agencies, tend to be more structured and use a persuasive or sales-oriented tone to sell a service or some form of fraud. One significant indicator is the presence of URLs, which are almost never found in ham SMS but can help identify the just mentioned persuasive or sales-oriented tone in spam messages.

In this context, it is important to mention the practice known as "pig butchering," a type of long-term scam where the scammer baits the recipient by pretending to have a personal relationship through a series of pseudo-natural messages. These spam messages often involve introducing themselves or asking for help with something even unrelated to the scammer's real intentions. When executed through text messages, this practice represents a modern and increasingly prevalent form of "smishing" (from SMS and phishing), which is the practice of sending seemingly natural text messages to trick someone into revealing confidential information or making investments. The desire to detect this type of spam messages has led to the choice of the dataset, as explained later.

Naive Bayes classifiers trained on tokenized words or combinations of words have long been highly effective at recognizing spam messages, outperforming methods such as support vector machines (AbdulNabi et Yasseen, 2021). In our study, we show that Naive Bayes classifiers trained on a fixed number of single tokenized words can effectively detect spam messages, even when more nuanced smishing messages are included in the spam category. The development of advanced deep learning models in recent years does not imply that Naive Bayes classifiers will soon become obsolete. In fact, given their performance, it is likely that Naive Bayes classifiers will continue to be used for many years, at least as an auxiliary tool for automatically classifying messages and emails as spam or not spam (AbdulNabi et Yasseen, 2021).

3 Methods

3.1 Dataset Description

The dataset used in this study was published under the name *SMS Phishing Dataset for Machine Learning and Pattern Recognition* by Mishra and Soni (2022). In this dataset, each message was originally labeled as ham, spam, or smishing thanks to receivers who had volunteered for the study. Smishing is a distinct category studied separately in the paper but is generally considered a subset of spam in the literature and in our study.

Table 1: Source: Mishra et Soni (2022)

Dataset Description	**Ham**	**Spam**
Almeida dataset (2011)	4825	739
Mishra et Soni additions (2022)		
Collected from Smartphones	19	
Converted to text from Pinterest website		146
Collected from Internet blogs		103
Collected from SmiDCA research work (2020)		139
Total	**4844**	**1127**

The dataset is intentionally composite to provide a greater variety of spam types. On the other hand, such a varied database might necessarily impose to overlook certain details that can be found in smaller and more contiguous datasets, such as sender number, time of sending, the region or country of the receiver, etc. For the purposes of this study, we prioritized the size and variety of spam messages, which is why we chose to use this dataset.

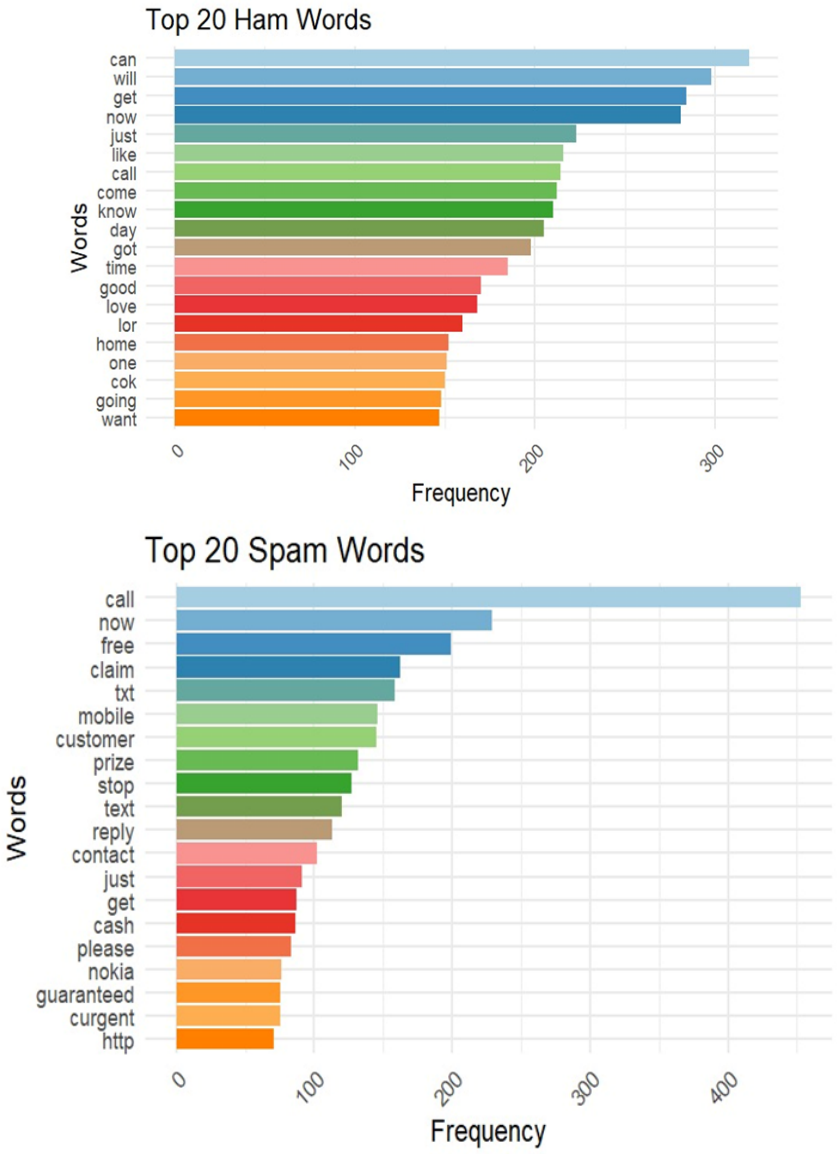
As specified in Table 1, the dataset is composed of three main parts:

- The biggest part comes from the Almeida Dataset (2011), which includes 4825 ham messages and 739 spam messages, with 5564 messages in total.
- Additions from the SMS Phishing Dataset by Mishra and Soni (2022), consisting of 388 spam messages in total.
- Messages collected from the SmiDCA research work by Sonowal and Kuppusami, consisting of 139 spam messages (2020).

The data was largely pre-processed, but the need to compare the SMS dataset with the Enron dataset required some further modifications. These included removing unnecessary columns, combining the spam and smishing columns into a single "spam" column, and aligning the labels with the Enron dataset. The final dataset, which was then used for word tokenization, is as follows:

- The **"text"** column contains the full message text.
- The **"label"** column contains the assigned label (spam or ham).

Figure 1: Frequency of words within class



- The **"label_num"** column contains a number associated to the label (1 for spam, 0 for ham).

Furthermore, as displayed in Figure 1, an exploratory data analysis of word frequency in the two classes of SMS is convincingly compatible with common experience with SMS. Despite the heterogeneity of SMS messages over time and space, this provides evidence that SMS messages are more homogeneous than we might have expected.

3.2 Statistical methods

The initial step in our text analysis procedure involves plain tokenization. A comprehensive dictionary of tokens, comprising all words found in the dataset, is created to vectorize each SMS. Consequently, SMS messages are transformed into points residing in an 8,872-dimensional vector space. Each attribute is assigned a value of 1 if the word is present in the SMS and 0 if it is not.

Following common practice, we exclude uninformative English words such as "at", "any", "who", etc. To reduce computational complexity, we then select 500 tokens based on information gain. However, as demonstrated in Appendix Table 3, even 400 or 300 tokens suffice for classifiers to perform well.

We employ a variety of classifiers including Linear Discriminant Analysis, logistic regression, multinomial and Bernoulli Naive Bayes classifiers, the latter two discussed extensively in Metsis et al. (2006). Training is conducted with 10-fold cross-validation over 80% of the original dataset. However, other than comparing different classifiers, we aim to build a new voting classifier.

Voting classifiers can enhance overall performance by aggregating information from multiple classifiers, as a classifier may detect a spam SMS that others miss. We explore three configurations:

- Simple Majority Voting: Each classifier votes, and the majority decision is selected.
- Soft Voting Mechanism: This approach averages the predictions of each classifier.
- Weighted Average Voting: Optimal weights for each classifier are computed via grid search to maximize the specificity rate.

These setups are designed to improve the robustness and accuracy of our SMS classification system.

4 Results

Figure 2 displays the ROC curves for the four classifiers, while Table 2 contains summary statistics for those four together with the voting classifiers.

The detection of difficult spams by the classifiers is summarized as follows: the logistic model detected 4 instances, the multinomial classifier detected 3 instances, the Bernoulli classifier detected 1 instance, and the LDA classifier detected none. Clearly, the Bernoulli classifier exhibits the highest overall accuracy, closely followed by the multinomial and LDA classifiers. In terms of precision for class spam predictions, the LDA classifier performs the best, indicating it had the fewest false positives for spam. For recall of class 1, the multinomial classifier outperformed others, identifying the highest number of actual spam messages and thus missing fewer spam instances. Regarding the detection of difficult spams, in the end the logistic model demonstrated a relatively strong performance, identifying 4 out of 14 difficult instances where other classifiers typically failed. This information is captured by the optimal weights found through grid search:

$$\text{Optimal weights} = \{w_{\text{LDA}} = 0.05;\ w_{\text{multinomial}} = 0.09;\ w_{\text{logistic}} = 0.41;\ w_{\text{Bernoulli}} = 0.45\}.$$

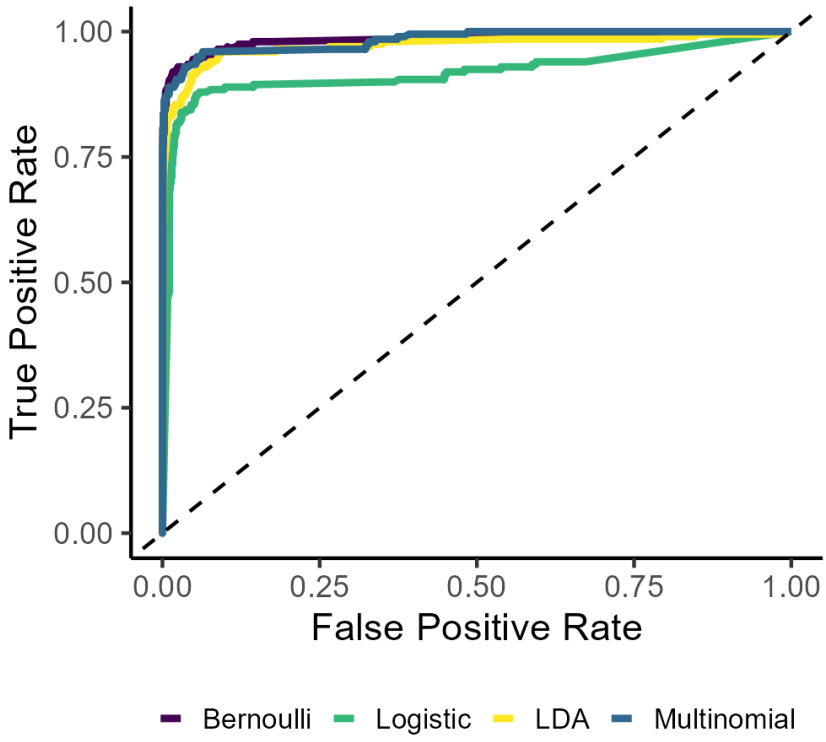
Table 2: Model performance metrics

Model	Sensitivity	Specificity	Accuracy
Multinomial	0.9793	0.8945	0.9648
LDA	0.9938	0.8090	0.9622
Logistic	0.9700	0.8291	0.9459
Bernoulli	0.9896	0.8935	0.9734
Hard Voting	0.9959	0.8643	0.9734
Soft Voting	0.9938	0.8794	0.9742
Weighted Voting	0.9907	0.9095	0.9768

5 Discussion

The primary aim of this study was to evaluate the performance of various classifiers and ensemble methods in detecting spam messages within a diverse SMS dataset. The findings

Figure 2: ROC curves for Different Models



indicate that while individual classifiers, such as the Bernoulli and Multinomial Naive Bayes, perform well, the use of ensemble methods enhances overall performance. Notably, the Weighted Voting method achieved the highest accuracy, and specificity, demonstrating the robustness of combining classifiers. This method leverages the strengths of each individual classifier, optimizing their contributions through a grid search for optimal weights. The improvement in performance underscores the benefit of aggregating multiple models to increase robustness and reliability in spam detection.

Despite the promising results, this study has some limitations. The dataset, although composite, ignores possibly relevant information such as the date of the message. For this reason we were not able to completely replicate Metsis et al. Future research should consider incorporating real-time data and exploring the adaptability of classifiers to new types of spam. As a final observation, our study focused on textual features, but future work could integrate metadata and contextual information to further enhance detection

accuracy.

6 Appendix

Table 3: Accuracy for 300, 400, 500 tokens

Model	300 Tokens	400 Tokens	500 Tokens
LDA	0.961	0.962	0.962
Logistic	0.955	0.946	0.946
Bernoulli	0.971	0.969	0.973
Multinomial	0.945	0.954	0.965
Hard Voting	0.960	0.961	0.963
Soft Voting	0.962	0.962	0.965
Weighted Voting	0.967	0.969	0.970

It is worth noting that, in this particular dataset, a smaller number of tokens produces similar accuracy rates across all models.

References

[AGC14] Ishtiaq Ahmed, Donghai Guan, and Tae Choong Chung. "Sms classification based on naive bayes classifier and apriori algorithm frequent itemset". In: *International Journal of machine Learning and computing* 4.2 (2014), p. 183.

[AHY11] Tiago A Almeida, José María G Hidalgo, and Akebo Yamakami. "Contributions to the study of SMS spam filtering: new collection and results". In: *Proceedings of the 11th ACM symposium on Document engineering*. 2011, pp. 259–262.

[Gra02] Paul Graham. "A plan for spam". In: *http://paulgraham. com/spam. html* (2002).

[MAP06] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. "Spam filtering with naive bayes-which naive bayes?" In: *CEAS*. Vol. 17. Mountain View, CA. 2006, pp. 28–69.

[MS20] Sandhya Mishra and Devpriya Soni. "Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis". In: *Future Generation Computer Systems* 108 (2020), pp. 803–815.

[SK20] Gunikhan Sonowal and KS Kuppusamy. "PhiDMA–A phishing detection model with multi-filter approach". In: *Journal of King Saud University-Computer and Information Sciences* 32.1 (2020), pp. 99–112.

[Yas+21] Qussai Yaseen et al. "Spam email detection using deep learning techniques". In: *Procedia Computer Science* 184 (2021), pp. 853–858.