Spam Filtering Through Supervised Learning Algorithms

By Daniel Ogulnick and Lohith Bollineni

1. Introduction

1.1 Our motivation for this project is to implement machine learning algorithms in a spam filtering program written in Python. We desire to utilize what we have learned in this class, in order to create a program which is both effective in its application and sound in its methodology, and therefore draw a clear relationship between the theory of Machine Learning and its embodiment in code. Additionally, we desire to be able to articulate how the frameworks of Artificial Intelligence can be translated into concrete software solutions, which can be useful and practical in a broader technological framework.

1.2 Our chief aim is to program a spam filtering application, which can learn from a given dataset of known spam/ham messages, and thereafter draw accurate conclusions about the classification of new data points. We intend to execute this program in a way which is widely applicable between datasets, is computationally efficient, and easily understandable to someone examining the code for the first time. To this end, we have implemented some of the well-established Python libraries, such as Numpy and Pandas, which are familiar to nearly all students and researchers in the field of AI and serve to simplify the code to what is conceptually important. We used the “Enron Dataset,” which is a collection of known spam and ham emails received by employees of the now defunct firm Enron, which has been used often in scholarly work in the area (Metsis).

1.3 In this report, we begin by discussing the general topic of spam, and provide information about how and why spam filtering programs are used. We then proceed to examine the theoretical foundation which underlies spam filtering, through which the program itself “learns.” We then go into greater detail about the implementation of these ideas, and then detail our results and considerations for future revisions or study. A list of sources referenced in this study is provided at the end.

2. Background/History of the Study

According to Cisco, one of the largest technology conglomerates in the world today, spam is “unsolicited and unwanted junk email sent out in bulk to an indiscriminate recipient list. Typically, spam is sent for commercial purposes. It can be sent in massive volume by botnets, networks of infected computers.” Oftentimes, spam messages are innocuous attempts to convince the recipient to purchase a product, but they can also contain malware scripts or social engineering attacks. Cisco identifies some of the most common types of spam messages as commercial advertisements, email spoofing, and sweepstakes winnings (Cisco). The term “ham” is often used in contrast to spam, and refers to any form of legitimate email.

Email has become a ubiquitous form of communication in the modern world since its inception in the 1970's, due to its low cost, low maintenance, and ease of use. However, with email's growth and manifold benefits, have also come many cyberthreats, with spam messages being among the most common. Nearly every user of email has complaints about the volume of spam messages received. While the vast majority of these are ignored, enough succeed so that sending these bulk emails remains an economically viable form of misconduct (Karthika.) Some studies suggest that over 85% of all emails are spam, with 421.81 billion of these messages send daily across the globe (Palival). Efforts have been undertaken by governments to mitigate this issue. The Controlling the Assault of Non-Solicited Pornography And Marketing (CAN-SPAM) Act of 2003 was a bill aimed at establishing national standards for the sending of email. While the FTC has enforced the law, and ISP's immediately used the bill to initiate litigation, spam has remained equally prevalent (FTC).

Spam filters are the predominant method of reducing the bulk of spam emails. An alternative to spam filters is to "blacklist" or "greylist" certain email servers, which means that the user's email client does not accept any messages from them. While this method may be effective in lessening the quantity of spam messages, it greatly increases the odds that ham is also filtered out. Given that some emails are of very great importance and urgency, it is essential that spam-reduction methods emphasize the minimization of false-positives (Palival). For this reason, spam filters have become the most common tool to this end. Spam filters are deployed at the level of the mail server, and most often during what is known as the SMTP dialogue. SMTP (Simple Mail Transfer Protocol) is the most modern network protocol used in sending, receiving, and routing email messages. By using the SMTP protocol, mail servers employ software that is known as the mail transfer agent. The mail transfer agent, or simply MTA, adds a “received” field to the message header, thereby leaving each email message with a path describing the route it took from the sender to the recipient until it reaches the server wherein the recipient’s mailbox is hosted. Additionally, mail servers also have programs called mail submission agents (MSA’s) and mail user agents (MUA’s), which are the underlying interface for sending and receiving emails respectively. In general, outgoing emails are only able to be submitted through servers hosted by their current network’s ISP, as a method of holding users accountable for spam messages they might send. The SMTP dialogue is the point at which the client (i.e. the sender) is connecting to an en-route server through the MTA, which can choose how to respond to the message. It is at this point, usually at the host mailbox where the spam filter is applied. There are various methods by which the filter can assess messages, for example with a virus scan. This project is focusing on how a filter can determine spam based on the semantic content of the message, as that is most often the basis upon which the decision is made (Godoy).

3. Approach and Implementation

Spam filters can employ a variety of methods for detecting spam. When they are implemented using a supervised algorithm, they “learn” how to determine spam from a variety of machine learning techniques. An approach used in this project is Naive Bayes Classification. This algorithm is “supervised,” meaning that it learns by mapping inputs to output variables, and does so through a set of “training data.” It then takes the patterns it has “learned” from this data, and applies it to future data points, which are referred to as the “test data.” Moreover, in this particular project, the algorithm would be one of classification rather than regression, since it operates in terms of discrete variables rather than continuous, such as height. Although simple and founded on strong assumptions about the independence of variables, Naive Bayesian Classification is the most commonly used in both open-source and commercial spam filtering programs, and has repeatedly been shown to be very effective, with the added benefit of having linear computational complexity and low storage requirements (Metsis). Particularly in data sets where the assumption of independence holds well (i.e. where spam is non-thematic and from various sources), the Naive Bayes Classifier performs very well, having famously won first and second place in the KDD-CUP 97, against other rigorously implemented algorithms in examining approximately 750,000 data entries (Marcus).

We implemented this project in the Python programming language, as is very common in machine learning. One of Python’s advantages, of which we availed ourselves, is its abundance of libraries which aid in the abstraction, cleaning, and manipulation of data. To this end, we used Pandas, Numpy, and others. SKLearn was used to generate metrics about the accuracy and precision of the program, as well as to vectorize strings in order to be able to effectively utilize them in the algorithm. Additionally, we integrated the filter with a web app, for the purposes of illustrating the program.

4. Experiment Results and Discussion

In the final implementation, we achieved a result of .96 accuracy and .99 precision - accuracy referring to how close the result is to the actual value, and precision meaning how close measurements of the same class are to one another. These results were tested by running some of our own emails through the filter, which correctly categorized the majority of them. Ultimately, the spam filter categorized about 29% of the data as spam, which is very close to the real number contained in the dataset. However, the filter was less effective in categorizing emails from outside the dataset. This likely comes as a result of the ham messages (being those sent among individuals at a financial services firm) pertaining to specific subjects regarding their business operation, and not of a more general nature. Therefore, we believe that this spam filter would scale appropriately if given a larger, more diverse set of training data. In practice, spam filters generally learn from the emails received by the given user, and so therefore become optimized the particular needs of this individual (e.g. learning and identifying their name). Additionally, a feature we did not include was a language filter, as it automatically classifies foreign languages as spam. This could be overcome if given an appropriate set of training data, as the spam filter would still learn in the same way it does in English. However, stopwords and possibly other symbols would need to be separately filtered out in order to maximize efficacy, which would need to be specified according to the semantic structure of a different language.

5. Conclusion

In this project, we used python and its constituent libraries to develop a spam filter. The filter was designed with a Naive Bayes Classifier, which is a supervised learning algorithm, and one which is very commonly used in spam filtering software. The dataset we used was the Enron dataset, which is commonly used in academic work regarding spam filtering, due to being one of the only significantly large datasets of genuine emails which are publicly available. Ultimately, we found that our spam filter was effective in accurately classifying between spam and ham, having learned from the training data and applied it to the test data. The shortcomings of this project are that it is less effective outside of the dataset, and that it lacks certain features that would be desirable in a commercially available spam filter. Additionally, the results of this implementation could be compared with those of other Naive Bayes Classification algorithms, such as the Multivariate Bernoulli or the Multivariate Gauss (Metsis).

6. References

Cisco. "What Is Spam Email?" *Cisco*, Cisco, 29 Oct. 2021, https://www.cisco.com/c/en/us/products/security/email-security/what-is-spam.html.

FTC. "FTC Announces First CAN-SPAM Act Cases." *Federal Trade Commission*, 29 Apr. 2004, https://www.ftc.gov/news-events/press-releases/2004/04/ftc-announces-first-can-spam-act-cases.

Godoy, Jorge. "1.1. Why Filter Mail during the SMTP Transaction?" *Why Filter Mail During the SMTP Transaction?*
, https://tldp.org/HOWTO/Spam-Filteringfor-MX/whysmtptime.html

Karthika, Renuka. "Spam Classification Based on Supervised Learning Using Machine Learning Techniques." *IEEE Xplore*, https://ieeexplore.ieee.org/abstract/document/5979035

Marcus, Mitch. *Building a Spam Filter Using Naïve Bayes*. https://www.seas.upenn.edu/~cis391/Lectures/naive-bayes-spam-2015.pdf.

Metsis, Vangelis. *Spam Filtering with Naive Bayes – Which Naive Bayes?* http://www2.aueb.gr/users/ion/docs/ceas2006_paper.pdf.

Palival, Divesh. *Email Spam Filtering Using Decision Tree Algorithm - IJSER*. https://www.ijser.org/researchpaper/EMAIL-SPAM-FILTERING-USING-DECISION-TREE-ALGORITHM.pdf.