BlueHost, INC

Email Spam Report

## presented by

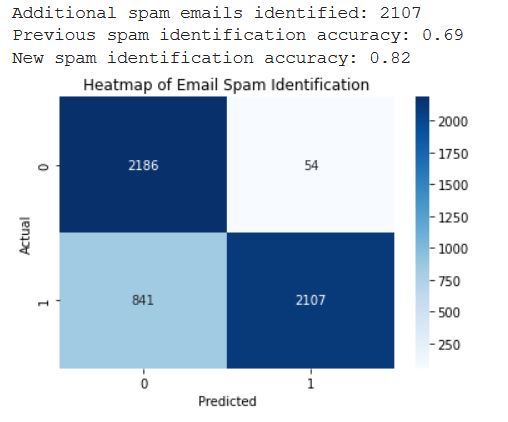
Team Member 1

Team Member 2

Team Member 3

1. Background

BlueHost provides email services to its clients and wants to improve the quality of its email filtering service beyond the naïve bayes filter currently used. While the SpamAssassin servers are able to identify spam with a high probability of accuracy, currently 69%, there are still spam messages that are slipping through as valid emails. Using the data provided by BlueHost, the goal is to predict which emails were spam and which emails were ham (non-spam) to more accurately filter emails for users.

Multiple models were tested and the random forest model performed the best. It achieved an R2 value of 0.78. It successfully identified 2107 of the 2948 spam emails that were missed by SpamAssassin. The model incorrectly identified 54 of the 2240 ham emails as spam.

The new model accurately identifies 82% of spam compared to 69% identified only through SpamAssassin. It incorrectly identified 0.3% of all emails as spam that were actually valid.

The new model was also tested on a holdout dataset and was able to identify an additional 1574 spam email messages and would reduce spam messages by an additional 28.74% beyond what had been blocked by SpamAssassin while only misclassifying 8 emails as spam.

1. Methodology

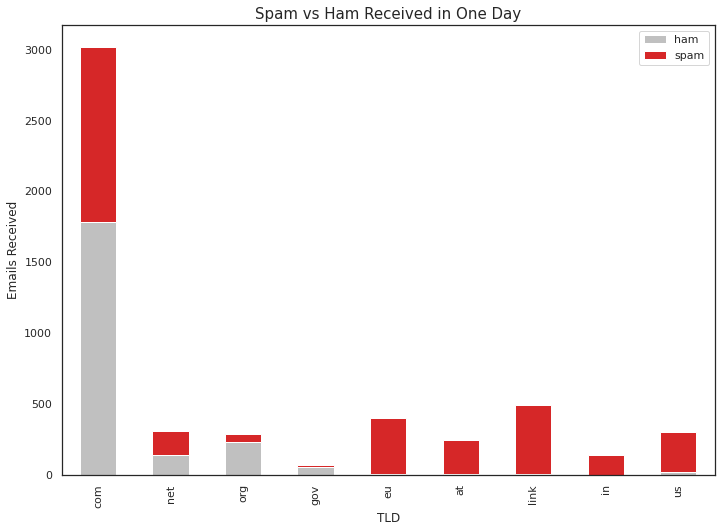
Processing vast amounts of private emails was not an option due to privacy concerns and restrictions. Instead, server logs were used. The log file contained 171,020 server communication records of which there were 16,867 individual emails. These logs required exensive processing to extract useful data to feed into the models. The following information was extracted from the logs for each e-mail:

* Who the message was from (whoever@somewhere.com)
* The domain name of the person it was from (somewhere.com)
* The top-level domain of the domain name (.com)
* "Common" domain name extensions were grouped into a variable (.com, .org, .net, .gov)
* The Spam weight that the SpamAssassin filter reported
* If the message was a local delivery or just getting forwarded to another server (no SpamAssassin spam checks occur on forwards)
* Any user defined content filter that was applied
* The number of recipients that the message was being sent to
* A list of the recipients
* A variable of whether the message appeared to be spam from other checks
* The delivery status of the message (Delivered, Bounced, Deleted)
* The time the message was delivered
* The id of the message

Identifying which emails were spam from logs was one of the biggest challenges of this project. Emails marked as spam by SpamAssassin were removed leaving 5,188 emails. Emails were then checked for cross-domain sending to identify spammers who were sending to multiple disparate industries. This indicated that there were at least 2,948 emails that should have been classified as spam, but were not.

The data was processed through the following models: XGBoost, ANN, KNN, random forest, and decision tree. The best performing model was a random forest. The R2 value for the model is 0.78. The model was able to identify 82% of the spam emails while only misclassifying 54 of the 2240 valid emails.

Top-level Domain (TLD) was one of the largest indicating factors for spam emails. Common TLDs (.com, .net, .org, .gov) had the majority of ham emails. Spammers used many other TLDs, but favorites were .eu, .at, .link, .in, and .us.



TLD length, email length, domain length, and time of email were all investigated as well for trends. When emails arrived had almost no correlation to whether an email as spam or ham, which was contrary to original assumptions of the CEO. Longer email length and longer domain length was indicitive of a spam email with emails longer than 33 and domains longer than 21 having a higher likelihood of being spam.

1. Results, Action Items, and Limitations

It is possible to identify spam messages from data logs without needing to parse individual emails. The random forest model was able to successfully identify additional spam emails and increase the accuracy to 82%.

The CEO’s concern about sending too many valid emails to user’s junk folder is a concern with this model as it identified 54 emails, or 2.4% of valid emails, incorrectly. Tradeoffs between removing an additional 11% of the spam while incorrectly identifying 2.4% of ham is a tough decision. This report recommends allowing businesses the option to turn on the new filtering if they are struggling with too much spam email. This would allow clients to self-select the tighter filtering schema.

Seeing as spammers are constantly changing their techniques, it is recommened to run the model on new data every three months to identify new TLDs that are being used, as well as verifying through a manual process the emails that are being marked as spam through the filter. Feeding the results of this new filter into the SpamAssassin filter would also be beneficial to help it identify spam emails with greater precision.

Assumptions made about cross-domain emailers will need to be verified with actual emails. Combining multiple day logs or increased ranges of logs could help the model become more accurate for future tests.

1. Python Notebooks

https://colab.research.google.com/drive/1DdoXOBO

Gist: https://gist.github.com/sampleperson/058213bb9d9b1885f27e25c003d1d