Spam Identification and Techniques

James Fullwood IV jaf16412@uga.edu

Christopher Gantes cjg49755@uga.edu

Edward Ghioalda ejg90726@uga.edu

Abstract

Spam has been a burden on email users and providers since the dawn of the internet, and almost as quickly as it has risen various techniques have been used to try to sort through them. This paper attempts to compare and contrast the performance of a variety of different models in the classification of spam. These models are trained and tested on two different datasets, the SMS Spam Collection Dataset, a dataset made from a mixture of scraped data from websites as well as a subset of data made from a set of SMS messages gathered for research and the Email Spam Classification dataset. This data is split between spam and ham, i.e wanted and unwanted emails. The models tested in this paper are the Decision Tree, Naive Bayes, Random Forest, Logistic Regression and Support Vector Machine. We found that The Random Forest and Decision Trees was the most accurate at predicting if a message was spam or not. Performing statistical analysis over the models performance compared to the Naive Bayes model showed that the models performance could be due to random chance rather than a difference in the models abilities.

1. Introduction

In the modern world a large amount of communication is done online through the use of sent messages. As a result, many malevolent actors have attempted to leverage the tools used to communicate online to sell products, hack into others' accounts, and to facilitate other malicious actions. These messages, known as spam, have been a problem as old as the internet itself. Despite decades of work done to build automatic filters to detect and filter these unwanted emails they still manage to get through. These messages are typically made automatically and sent through the use of automated bots. The cost to manage this problem has run into the millions and as such there is a great demand for systems to automatically detect and filter these messages.

Most modern online web services deal with spam through the usage of ai models trained to detect and filter these messages. These web service providers use a variety of techniques to create and train these models, each with a variety of outcomes of performance. In this paper we wish to identify and review some of the most commonly used models to classify emails into spam or ham (ham being a recently coined term to

describe non malicious emails). We then compared the models in terms of their accuracy (how well it classifies a message as spam or not) and in terms of its precision (how well it limits the number of false positives and negatives).

For the training and testing of our models we relied upon the SMS Spam collection dataset. This dataset is a collection of english SMS messages gathered together for the explicit purpose of training and testing spam detection models. We preprocess the data using TF-IDF vectorization and then split the model into a training and testing set. The same preprocessing steps is performed upon the second dataset, this one

To give a brief overview of the experiment we did upon the datasets we initialized the data, extracted the text, and preprocessed it using TF-IDF vectorization. Afterwards we initialized five different models, Decision Trees, Logistic Regression, Naive Bayes Classifier, Random Forest, and the Support Vector Machine, and then trained each one over the data. We then tested them and performed a few statistical tests and got a few measurements of their success, such as their accuracy and their precision, as well as their CV scores. Lastly, we then compared the models against the naive bayes classifier by performing a t-test.

After performing our tests we found that the Decision Tree and Random Forests out performed the other models in terms of classification accuracy and precision. This finding is reinforced by the cross validation scores of the different models. For all of the models the t-statistic was positive indicating that the models outperformed the Naive Bayes classifier. However, since the p-value for all of the models was higher than .05 we are unable to reject the null hypothesis.

We will start with a quick review of the current literature on the most up to date machine learning techniques to classify spam. From there we will discuss our data models in more detail and explain the preprocessing that we did to make these data sets usable for our program to train models on. Afterward we will go over the exact methods we employed to produce our models, going into detail over the ML frameworks chosen, the software used to create the models and any problems we encountered. We will end with a discussion of the results of the experiment, why the winning model was chosen and what was the reason it outperformed the other models

2. Related Work

Due to the massive issue that is spam online there has been a large amount of work done on producing and implementing automatic spam detection algorithms into websites and email services. This problem is exacerbated by the fact that the problem of spam is not a static one and could be classified as an adversarial problem, i.e. there is an active attempt to circumvent the measures one creates to solve the problem. The paper “A review of spam email detection: analysis of spammer strategies and the dataset shift problem” which tests the hypothesis of model performance degradation

found that when these models are trained on older datasets there was significant degradation in their performance in classifying spam on newer datasets. So there is an obvious and urgent need to create efficient and robust data classification models, and there have been many approaches to resolving this problem, from blacklisting particularly prominent spam email addresses, to deep learning neural networks to the more humble naive bayes classifier, each one having their own advantages and disadvantages.

Some of the papers focused on the most common implementations of spam detection algorithms. For instance the authors of the paper “The Comparison of Machine Learning Algorithms for Email Spam Detection” created and tested a variety of different machine learning models such as SVMs, Random Forests, Decision Trees, K-Nearest Neighbors, and Naive Bayes and compared and contrasted their performance. While this is a good starting point for the development of more sophisticated models, they stopped short at the identification and refinement of the best performing model out of the bunch. Another paper “Efficient information theoretic strategies for classifier combination, feature extraction and performance evaluation in improving false positives and false negatives for spam e-mail filtering” discussed various methods for improving the precision, reducing the number of false positives and negatives, of the models that they tested. They did not however identify the best model out of the bunch wholesale rather opting to consider the best model for FN and FP reduction.

There has been some work in changing the models used to classify spam “An Integrated Model to Email Spam Classification Using an Enhanced Grasshopper Optimization Algorithm to Train a Multilayer Perceptron Neural Network” used a more exotic algorithm to train a two level perceptron to classify spam and found a remarkable increase in the amount of spam filtered through its model compared to many of the other standard models. “An Intelligent Model for Email Spam Classification” looked at a variety of different models and after seeing that each one has its own advantages in places and disadvantages in other places suggested that a hybrid model was a better way to approach the problem.

A paper that actually implemented a hybrid approach to solving the problem “Spam Email Detection Using Machine Learning Integrated Into the Cloud” found that this resulted in SVM and Decision Trees out competing other models in terms of its performance. A couple of other papers “Classification and Prediction of Spam Emails Based on AI Enabling Models Using Deep and Machine Learning Techniques” and “Spam Email Detection Using Deep Learning Techniques” went into the usage of deep learning to classify and detect spam. Both of the papers found that a deep learning approach results in models that are very accurate, precise and robust.

Research into solving spam is not only limited to text based spam such as emails and SMS messages, but also goes into image spam such as the paper “Image Spam

Classification using Deep Learning" which as the title suggests is an indepth look at making models using deep learning to classify sent images into spam images and ham images. While this paper is useful in showing the versatility of these models in regards to classification there is limited usability considering our models are dealing with text rather than images.

3. Data Preprocessing

To test and train our models we relied on two datasets. The SMS Spam Collection Dataset and the Email Classification Dataset. The SMS Spam Collection Dataset is a dataset composed of scrapped SMS messages, with some classified as spam, unwanted, and others classified as ham, wanted. This is a commonly used database collected for the express purpose of data mining testing. This dataset consists of 5574 english messages separated into two categories, spam and ham. The file contains lines representing each message separated into two columns. One column contains the data the other contains text classifying it as spam or ham. This dataset is heavily weighted in favor of ham with 87% being ham and the other 13% being spam.

Our second dataset is Email Classification dataset. It is much smaller than the other dataset coming in at 179 instances but has a closer amount of spam and ham, coming in at 56% ham and with 44% spam. The data in this dataset was synthetically generated for the purposes of testing and training models. One benefit of having these two different datasets is that robustness of the models can be tested as one is meant to simulate SMS messages and the other emails.

For both of the models we preprocessed the data using TF-IDF vectorization. TF-IDF vectorization allows for textual data to be used for machine learning models. This works by translating the text into numerical vectors that can be read by the models. The words are assigned weights due to their importance and frequency inside of the document. Each of the datasets were split 30% training data 70% testing data for the experiments.

4. Experiments

To perform our experiment we used the pandas and scikit-learn machine learning frameworks to create our models and read and interpret our datasets. Loading the data from the csv files using pandas we extracted the data from the SMS spam data set using the read_csv() method with its encoding parameter set to windows-1252. This data was then preprocessed using the TF-IDF vectorization. This was fit using the fit_transform(0 function. By using TF-IDF vectorization we made the textual data able to be used for machine learning models. From there the data was split in two. One section (70%) was used for the purposes of testing while the other (30%) was used for training.

After the data was loaded in and preprocessed we then created several machine learning models using scikit-learn python library. These models were the Decision Tree,

Logistic Regression, Naive Bayes, Random Forest and the SVM. Each of these were initialized using the default parameters. From here each model was trained using the data gotten from the SMS dataset and then predictions were made using the model.predict() method over that dataset. The models' scores were calculated using the accuray_score() and precision_score() methods.

After this was done five-fold Cross-validation was performed for each model and stored. T-tests were then performed on each model against the Naive Bayes model. From this we derived our t-statistic and p-values for each comparison showing how each model performed compared to the Naive Bayes model. There wasn't any significant issue with creating the models and performing the tests beyond getting the correct parameters set to read the data from the datasets.

5. Analysis

SMS Dataset Scores

Model Type	Accuracy	Precision
Decision Tree	0.9556523968213279	0.8629550321199143
Logistic Regression	0.927710843373494	0.9872881355932204
Naive Bayes	0.9141245834401436	1.0
Random Forests	0.9610356318892591	1.0
Support Vector Machine	0.9551397077672392	0.9912536443148688

Email Spam Classification Dataset

Model Type	Accuracy	Precision
Decision Tree	0.9841269841269841	1.0
Logistic Regression	0.9365079365079365	0.9591836734693877
Naive Bayes	0.9126984126984127	0.8620689655172413
Random Forests	0.9603174603174603	1.0
Support Vector Machine	0.9285714285714286	0.9583333333333334

From our analysis of these five models – Decision Trees, Logistic Regression, Naive Bayes, Random Forests and Support Vector Machines – for the purposes of spam classification we found that the Random Forest and Decision Tree models

performed the best out of the group. Random Forests achieved the greatest accuracy out of the models tested and also had a perfect precision score, and while the Decision Tree did not have the highest precision score it did have the second highest accuracy score. When comparing each model to the Naive Bayes classifier, however, we found that due to each model's p-value being higher than .05 the null hypothesis could not be rejected. Due to the largest dataset being the SMS dataset being weighted heavily in its samples towards non spam targets, the models trained on that dataset may not perform as well if tested on a dataset with a higher proportion of spam.

6. Conclusion

The problem of spam is an ongoing issue and may never be conclusively solved. Despite the massive amount of work done in the field to generate and produce a variety of different models to detect spam the problem is still with us. In this paper we compared and contrasted a variety of different models' ability to classify spam. We found that the Decision Tree and Random Forest Models out competed the other models in terms of their ability to accurately classify data and to avoid false positives and negatives. We also compare each model to the Naive Bayes classifier and receive inconclusive results over their performance. This work could be extended by training and testing the models on larger datasets and performing more statistical analysis on the models.

Bibliography

1. Abayomi-Alli, Olusola, et al. “A deep learning method for automatic sms spam classification: Performance of Learning Algorithms on Indigenous Dataset.” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 17, Apr. 2022, https://doi.org/10.1002/cpe.6989.
2. AbdulNabi, Isra’a, and Qussai Yaseen. “Spam email detection using Deep Learning Techniques.” *Procedia Computer Science*, vol. 184, 2021, pp. 853–858, https://doi.org/10.1016/j.procs.2021.03.107.
3. Aiwan, Fan, and Yang Zhaofeng. “Image spam filtering using convolutional neural networks.” *Personal and Ubiquitous Computing*, vol. 22, no. 5–6, 9 July 2018, pp. 1029–1037, https://doi.org/10.1007/s00779-018-1168-8.
4. Chakravarty, Ajay, and V. Manikandan. “An intelligent model of email spam classification.” *2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, 26 Dec. 2022, https://doi.org/10.1109/icerect56837.2022.10059620.
5. Ghaleb, Sanaa A., et al. “An integrated model to email spam classification using an enhanced grasshopper optimization algorithm to train a multilayer Perceptron Neural

Network." *Communications in Computer and Information Science*, 2021, pp. 402–419, https://doi.org/10.1007/978-981-33-6835-4_27.

6. Jáñez-Martino, Francisco, et al. "A review of Spam Email Detection: Analysis of spammer strategies and the Dataset Shift Problem." *Artificial Intelligence Review*, vol. 56, no. 2, 11 May 2022, pp. 1145–1173, https://doi.org/10.1007/s10462-022-10195-4.
7. Kang, Gwonsik, et al. "The comparison of machine learning methods for email spam detection." *Innovative Mobile and Internet Services in Ubiquitous Computing*, 2023, pp. 86–95, https://doi.org/10.1007/978-3-031-35836-4_10.
8. Karn, Richa Kumari, et al. "Spam email detection using machine learning integrated in cloud." *2023 International Conference on Networking and Communications (ICNWC)*, 5 Apr. 2023, https://doi.org/10.1109/icnwc57852.2023.10127237.
9. Learning, UCI Machine. "SMS Spam Collection Dataset." *Kaggle*, 2 Dec. 2016, www.kaggle.com/datasets/uciml/sms-spam-collection-dataset.
10. Muhammad, Junaid Mazhar, et al. "Classification and prediction of spam emails based on AI enabling models using Deep and machine learning techniques." *2022 International Conference on Emerging Technologies in Electronics, Computing and Communication (ICETECC)*, 7 Dec. 2022, https://doi.org/10.1109/icetecc56662.2022.10069229.
11. Sawhney, Prisha. "Email Classification (HAM-SPAM)." *Kaggle*, 16 Apr. 2024, www.kaggle.com/datasets/prishasawhney/email-classification-ham-spam.
12. Singh, Ajay Pal. *Image Spam Classification Using Deep Learning*, https://doi.org/10.31979/etd.wehw-dq4h.
13. Zorkadis, V., et al. "Efficient information theoretic strategies for classifier combination, feature extraction and performance evaluation in improving false positives and false negatives for spam e-mail filtering." *Neural Networks*, vol. 18, no. 5–6, July 2005, pp. 799–807, https://doi.org/10.1016/j.neunet.2005.06.045.