Internet Security: Using ML to Identify Phishing

Janmejay Buranpuri
Faculty of Engineering University of Western Ontario
London, Canada
jburanpu@uwo.ca

***Abstract*—As the internet continues to grow and evolve, the number of dangerous websites continue to grow in tandem. One of the most common cyber threats found online to this day is the attack known as phishing websites. Phishing attacks use social engineering within emails, messages, and links to appear as a trustworthy and safe website, however underneath the surface, they are used to obtain sensitive information such as passwords or even financial information. Being able to browse the internet has become a necessity for modern day people, however, it may take a keen eye to be able to discern whether or not a website can be considered trustworthy. With this in mind, we had decided to use two machine learning models help us find out if a URL leads to a phishing website or not. In this project, we had decided to compare a logistic regression model against a multinomial naïve bayes model to see which method is more accurate. After evaluating both models, the results show that both models operate with similar efficiency and accuracy, however, logistic regression using pipelining on average performed marginally better with an accuracy of 96.6%.**

I. Introduction

The number of phishing sites has been on the rise since 2009 [1]. Currently there are millions of phishing sites on the internet having malicious intent to get people's information, and money [1]. Some of these sites are getting so good that people have a hard time telling the difference between a real site and a phishing site [1]. Many companies even now have phishing email training to avoid cyber attacks and other issues [1]. It is important to have a machine learning program which can identify common variables in phishing sites and send them directly to spam with high accuracy.

In the program two algorithms are used: logistic regression and multinomial naïve bayes . Both these are tested on the model to compare which one has a higher accuracy. Logistic regression being the better model in this case is put into a sklearn pipeline that can be used to cross validate using different parameters. This report will also show confusion matrices to show the accuracy of each model. The results show a 96.6% accuracy in identifying phishing emails. 96.6% is a very high rate of accuracy for identifying emails, although, for a large sample size it can still mean missing thousands of real links.

II. Related Work

The concept of phishing attacks has been a topic of interest for researchers all over world studying internet security. There are many different research projects revolving around the social engineering involved with phishing attacks as well as a myriad of projects attempting to improve the detection of these harmful websites. The objective of this project is to demonstrate the strength and accuracy of different machine learning models and find out how they can help us identify a phishing website at a moments notice.

While we would like to create a program that can quickly scan and survey a suspicious website for possible phishing activity, it would be very difficult to implement such a thorough analysis. Instead, in this project, we will be training the models to recognize potential phishing websites while only observing the patterns within the attacking page's URL and http requests.

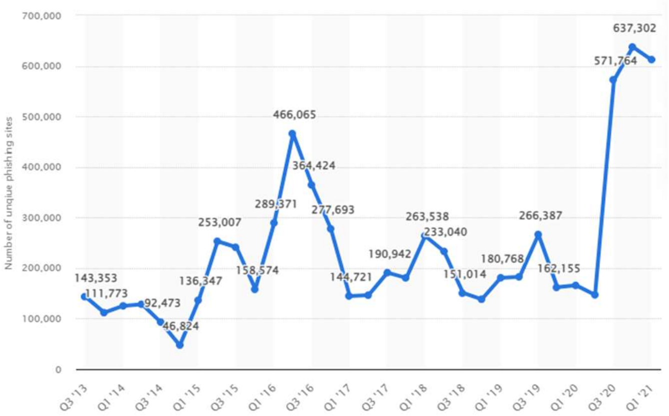


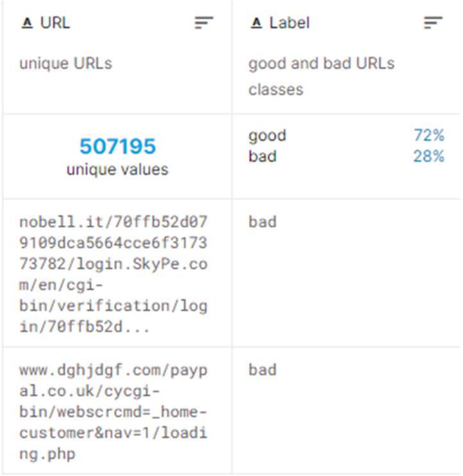
Fig. 1. Number of global phishing sites identified as of Q1 2021 [2]

III. Data

The objective of this project is to implement a machine learning model to be able to correctly identify phishing websites when given a URL or website link. Both the project concept and dataset originate from a user submitted challenge hosted on Kaggle. This dataset contains 507,195 entries of unique URLs along with manually inputted labels indicating whether a website is considered harmful or safe.

TABLE I
DATA DICTIONARY

Variable	Variable Explanation
URL	A Uniform Resource Locator, which is a unique identifier/link for each website entry
Label	Denotes whether a website is considered trustworthy (good) or dangerous (bad)



URL	Label
unique URLs	good and bad URLs classes
507195 unique values	good 72% bad 28%
nobell.it/70ffb52d079109dca5664cce6f317373782/login.SkyPe.com/en/cgi-bin/verification/login/70ffb52d...	bad
www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrcmd=_home-customer&nav=1/loading.php	bad

Fig. 2. Dataset Example Entry

IV. METHODS

This project aims to use RegexpTokenizer and Snowball Stemmer then use data from those to create a logistic regression model and a multinomial naïve bayes model. These models will be compared to check which one gives greater accuracy.

The phishing data consists of two columns. It contains a link, and a datapoint stating if the link is "good" or "bad." Good links are not phishing whereas bad links are phishing. RegexpTokenizer splits a string into substrings using regular expression [3]. In this project it is used using: "*r'[A-Za-z]+',*" meaning every word which is split by anything other than a letter will be put into a string. Then a column is added to the phishing data csv. It tokenizes each link for each row.

Next method used is a snowball stemmer. Snowball stemmer accepts different languages and the language used for this project was English [4]. It also breaks down the link into strings like RegexpTokenizer although this breaks it down into English words. This is also added to a column of the phishing datafile.

Next, two variables are created, one being *"bad_sites"*, and other one being *"good_sites"*. This splits the table up into sites which aren't phishing and other sites which are phishing.

All this data is then put into a model using CountVectorizer and converted into an array. This is then used to train the models. This is visualized using a confusion matrix. This model used logistic regression.

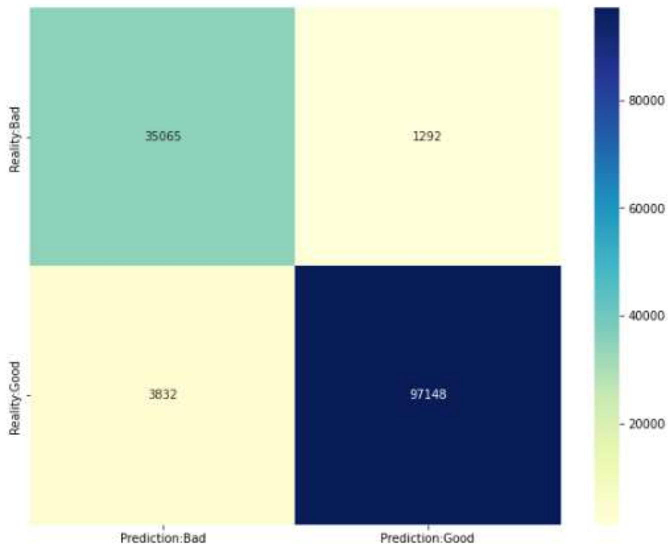


Fig. 3. Confusion Matrix for Logistic Regression.

The accuracies were as follows:
Training Accuracy: 0.977811164319226
Testing Accuracy: 0.9626903165206754

This matrix shows it identified 36065 phishing sites correctly, and 97148 sites good site correctly as well. This shows 3.8% of the sites were done incorrectly in the testing accuracy. Which are thousands of emails since the data file is so large.

Next a multinomial naïve bayes classifier is used. Fig. 2. Shows the confusion matrix for NMB classifier.

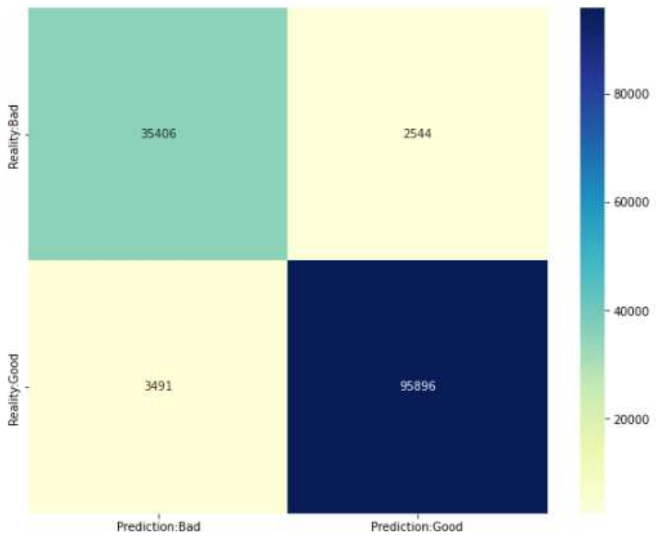


Fig. 4. Confusion Matrix for NMB.

For which the accuracies were as follows:
Training Accuracy: 0.9744010446373744
Testing Accuracy: 0.9560569984781959

The better model then was put into a pipeline to show the final results. The pipeline confusion matrix and accuracies are shown in the results section.

V. Results

As shown in the methodology section, comparing Fig 1 and 2 shows logistic regression is marginally better than NMB. Fig 3 shows the bar graph comparing both the models.

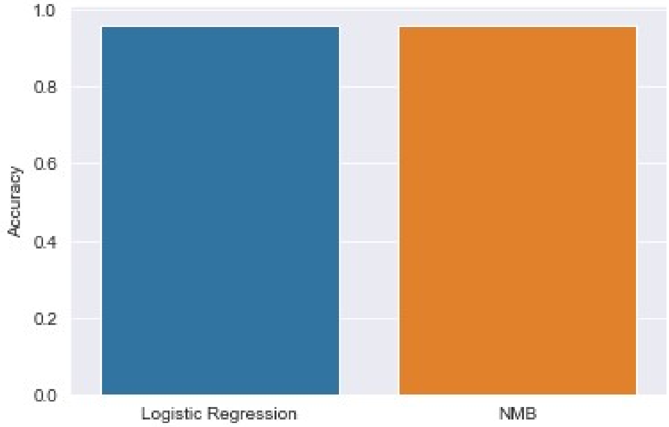


Fig. 5. Logistic Regression vs NMB

Although it is important to realize in this case, if a sample size was in the millions, the NMB would have thousands more inaccurate sites prediction compared to logistic regression.

Then a pipeline is created using logistic regression giving the following results:
Training Accuracy: 0.9816848661072938
Testing Accuracy: 0.9669571928904811

Fig 4. Shows the confusion matrix for logistic regression using pipeline:

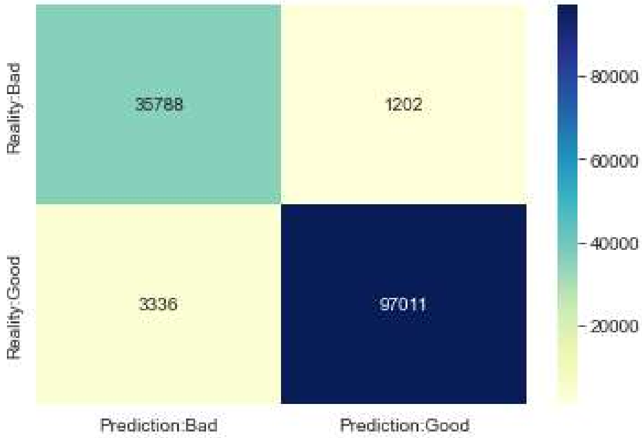


Fig. 6. Confusion Matrix for Logistic Regression in Pipeline.

This gives us a final testing accuracy of 96.6%.

The final results for logistic regression in a pipeline gave 96.6% accuracy. The model was able to identify 35988 phishing sites, 97011 good sites from the data given. Although, it did fail to identify 1202 phishing sites, and predicted 3336 good sites as phishing. This data is visualized in Fig. 4.

The implications of this are massive since it resulted in such a high accuracy for identifying sites which are phishing. Although, if a sample size was 10 million sites, with an error rate of 3.6%, there would be 360,000 falsely identified sites. This number grows massively as the number of sites increase. Meaning although accuracy is high, the model is still not perfect.

VI. Conclusions

After evaluating our two machine learning models, we had obtained the consistent result that the logistic regression model was marginally more accurate at identifying a potential phishing attack than the multinomial naïve bayes model. When both models are trained form the same dataset, both models are able to reach an accuracy rating of at least 95% with the logistic regression model slightly edging out the competition at 96.6%. After analyzing these results, it is clear that a lot of phishing websites follow a common URL pattern that is easily identifiable through machine learning. Using this knowledge, we could possibly create a better filtration system that could be implemented into browsers to protect users from a majority of phishing attempts. While this may prevent the majority of attacks, on a grand scale, thousands of phishing websites may still slip through this filtration system, so it should still be in everyone's best interest to learn how to identify phishing scams to keep your information safe when browsing the internet. If we were to conduct this project again, more machine learning models should be evaluated to find which model performs the most accurately to prevent more scams from going unnoticed.

References

[1] Phishing statistics (updated 2022) - 50+ important phishing stats. Tessian. (2022, March 7). Retrieved April 7, 2022, from https://www.tessian.com/blog/phishing-statistics-2020/#:~:text=Symantec%20research%20suggests%20that%20throughout,as%20the%20primary%20infection%20vector.

[2] Johnson, J. (2021, July 20). Number of global phishing sites 2021. Statista. Retrieved April 7, 2022, from https://www.statista.com/statistics/266155/number-of-phishing-domain-names-worldwide/

[3] NLTK. (n.d.). Retrieved April 8, 2022, from https://www.nltk.org/api/nltk.tokenize.regexp.html#:~:text=A%20RegexpTokenizer%20splits%20a%20string,%3E%3E%3E%20from%20nltk.

[4] NLTK. (n.d.). Retrieved April 8, 2022, from https://www.nltk.org/api/nltk.stem.snowball.html