Detecting Phishing Scams with Machine Learning Report

Carson Trimborn

I. Introduction

Since the creation of the internet, the world has evolved into a digitally driven society. The rapid evolution of technology has been the catalyst for a digital revolution that has characterized today's generation. While society has largely benefitted from these innovations, these tools have also opened the door for bad actors to exploit their capabilities and do harm. One of the most popular methods used by cyber criminals is known as phishing. Phishing is the fraudulent practice of sending messages while posing as reputable sources in an attempt to scam and steal personal information such as passwords or financial information. Phishing scams have also become increasingly successful due to factors such as the takeoff of artificial intelligence technologies which have aided in the ability to disguise phishing scams as legitimate websites. In this project, I will test several machine learning classification models with the aim of developing a strong model for predicting phishing scams. The algorithms I will test are logistic regression, decision trees, random forest, and support vector machines. I will then fine tune the models using grid search for hyperparameter tuning, and select the optimal model for making predictions on the test data for further analysis.

II. Methodology

The dataset I will be using for this project is the Web Page Phishing Detection dataset from kaggle, originally sourced from Mendeley Data. The dataset contains 11,430 URLs with 88 associated features extracted from the structure of the URL, content of the corresponding web pages, and the querying of external services. The dataset also includes labels for each URL in the "status" column, designating each URL as either "legitimate" or "phishing". Conveniently, the data is also balanced, with 50% of the URLs being legitimate and the other 50% being phishing scams.

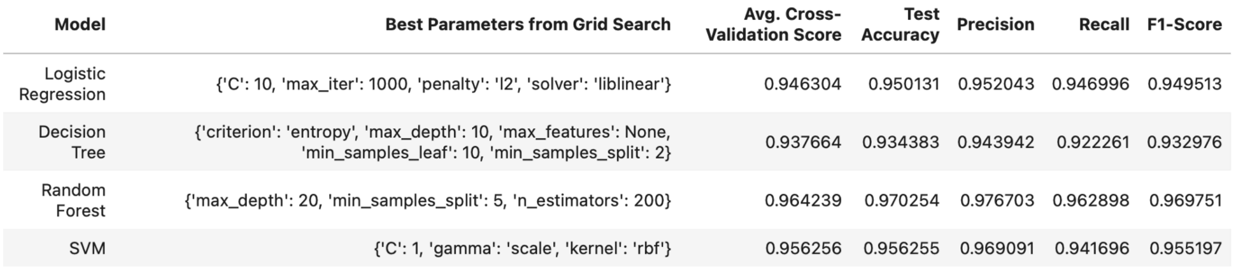
Before the development of the algorithms, the data needed to be preprocessed. First, since all of the information from the structure of the URL is already extracted and detailed throughout the feature columns, the URL column itself was unnecessary for analysis and was thus removed from the dataset. Next, the label column, "status", was encoded as 1's and 0's, with a 1 representing a phishing scam and a 0 representing a legitimate URL. The data was also split into training, validation, and test sets with 80% of the data being used for training, 10% for validation, and 10% for testing. And finally, due to big differences in the scales of the features, the features were standardized to ensure that each feature contributes equally to the model.

The first algorithm that was tested and used as the baseline model was logistic regression. The logistic regression model was then followed by the other models: decision tree, random forest, and support vector machine. Aside from setting random states for reproducibility, and max iterations to ensure convergence, each model was initially built using the default parameters. At this stage in the project, each algorithm was evaluated using their cross validation scores. Additionally, the resulting models were used to make predictions on the validation data for computing their corresponding classification metrics and confusion matrices.

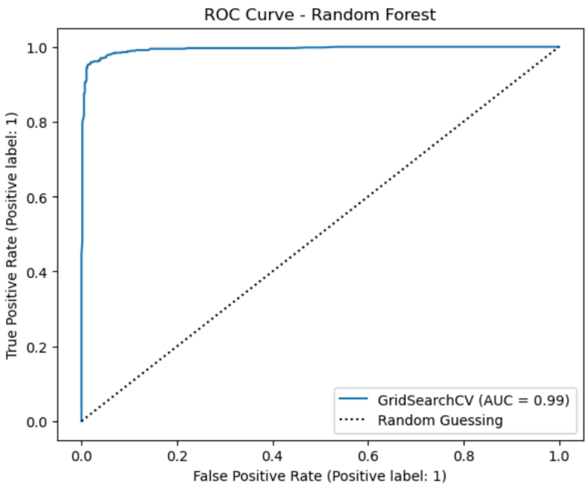
The next stage in the project was dedicated to hyperparameter tuning. This was done with the grid search method, using the cross-validation accuracy as the evaluation metric. Once the grid search for each model was completed, predictions were made on the test data using the best estimators. The accuracy, precision, recall, and F1 score were then recorded and compared.

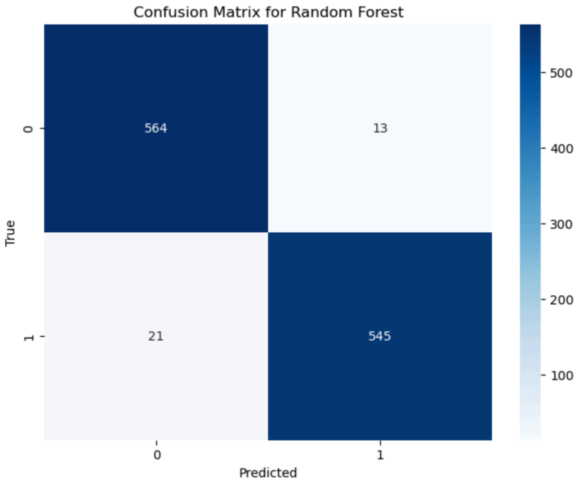
III. Results and Evaluation

From the metrics calculated on the test data it was determined that the random forest classifier with max depth set to 20, min_sample_split of 5, and a n_estimators of 200 was the best model. It outperformed the other models in every metric, achieving an impressive test accuracy of about 97.03%. As seen below, the random forest ROC curve was also strong, with an AUC of 0.99.

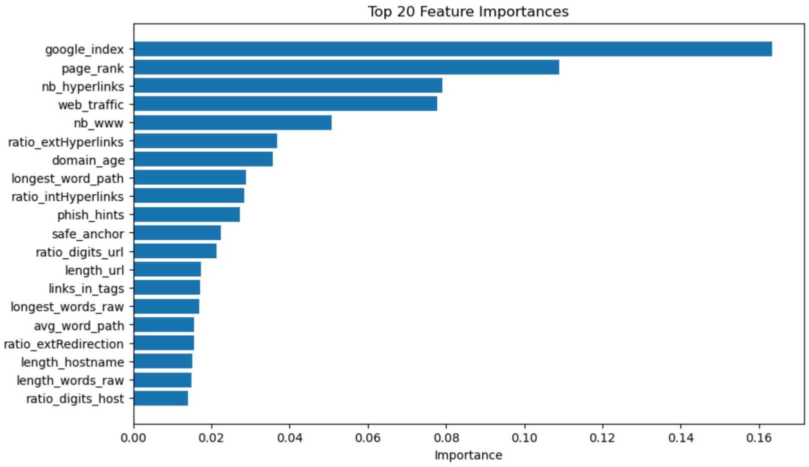


Model	Best Parameters from Grid Search	Avg. Cross-Validation Score	Test Accuracy	Precision	Recall	F1-Score
Logistic Regression	{'C': 10, 'max_iter': 1000, 'penalty': 'l2', 'solver': 'liblinear'}	0.946304	0.950131	0.952043	0.946996	0.949513
Decision Tree	{'criterion': 'entropy', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 10, 'min_samples_split': 2}	0.937664	0.934383	0.943942	0.922261	0.932976
Random Forest	{'max_depth': 20, 'min_samples_split': 5, 'n_estimators': 200}	0.964239	0.970254	0.976703	0.962898	0.969751
SVM	{'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}	0.956256	0.956255	0.969091	0.941696	0.955197





Above is the confusion matrix for the optimal random forest classifier. Among the test set, 545 of the phishing scams were correctly identified as phishing and 564 of the legitimate URLs were correctly identified as legitimate. However, 21 phishing scams were incorrectly predicted to be legitimate and 13 legitimate URLs were incorrectly predicted to be phishing scams.



As seen in the feature performance plot, 4 features stand out from the others in their importance. These include the Google index, page rank, number of hyperlinks on the web page, and the web traffic a web page sees. Google index is a feature google uses to index websites it knows about. Page rank is a metric used by google representing web page importance, with values ranging from 0 to 10.

IV. Discussion

I was very pleased with the results of this project. However, that's not to say there weren’t some challenges along the way. One challenge I came across with this project was algorithm runtime. This was especially difficult when doing a grid search with the more computationally intensive algorithms such as random forest or SVM. Though there were some difficulties, exploring this topic and developing the models revealed some interesting insights about phishing scams. For one, much of the information that can reveal a phishing scam can be found in the structure of the URL. Furthermore, it was interesting to see how metrics recorded by search engines like Google can serve as powerful indicators of a web page’s legitimacy.

An improvement that can be made to this project would be using better software with more computational power to conduct a larger grid search for hyperparameter tuning in order to enhance model performance. Additionally, testing more algorithms such as XGBoost and neural networks could provide valuable insights and improvements. Nonetheless, the use of the detailed web phishing dataset and the machine learning methods outlined in this project enabled me to develop a high-performing algorithm for detecting phishing scams.