Improving Phishing Website Detection with Machine Learning

Adam Eccles

Maryville University

aeccles1@live.maryville.edu

Abstract - Phishing remains one of the most prevalent cybersecurity threats, often leading to significant financial and data loss for the unsuspecting users. This study explores the application of supervised machine learning to detect phishing websites using a feature-engineered dataset containing both text based and content based indicators. I implement a range of preprocessing techniques including SMOTE for handling class imbalance, select top-ranked features from a Random Forest model, Cross-validation, and classification reports are used for evaluation. Results suggest that ensemble methods such as Random Forests and Logistic Regression with L1 regularization yield high performance achieving up to 98% recall; this shows further potential for deployment in real-world phishing detection systems.

I. Introduction

Phishing attacks have long been a threat to cybersecurity and have become significantly more frequent due to advancements in generative AI. Phishing attacks are drastically on the rise with an increase of 4151% since the introduction of Open AI's ChatGPT to the world in 2022 [1]. This surge highlights just how generative AI tools are being weaponized by cybercriminals to craft more convincing and scalable phishing attacks.

In response to these escalating threats, my project focuses on developing a machine learning-based phishing detection system. By leveraging a feature-engineered dataset encompassing both text and content based indicators, I am aiming to enhance the detection accuracy of phishing websites. My approach includes addressing class imbalance through techniques like SMOTE, selecting top-ranked features using Random Forest models, and evaluating classification performance across multiple algorithms.

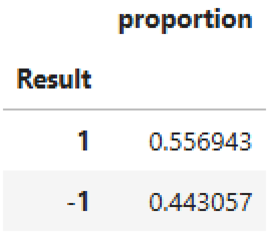
II. Problem Statement

Phishing websites deceive users into disclosing sensitive information by imitating legitimate websites. As these attacks grow in volume and complexity, especially due to the help of AI-generated attacks, traditional security measures are struggling to detect them in time. This project addresses the challenge of accurately identifying phishing websites by building a supervised machine learning model that analyzes URL structure, domain trustworthiness, and page behavior. The goal is to create a robust detection system that improves user safety and mitigates phishing threats in real time.

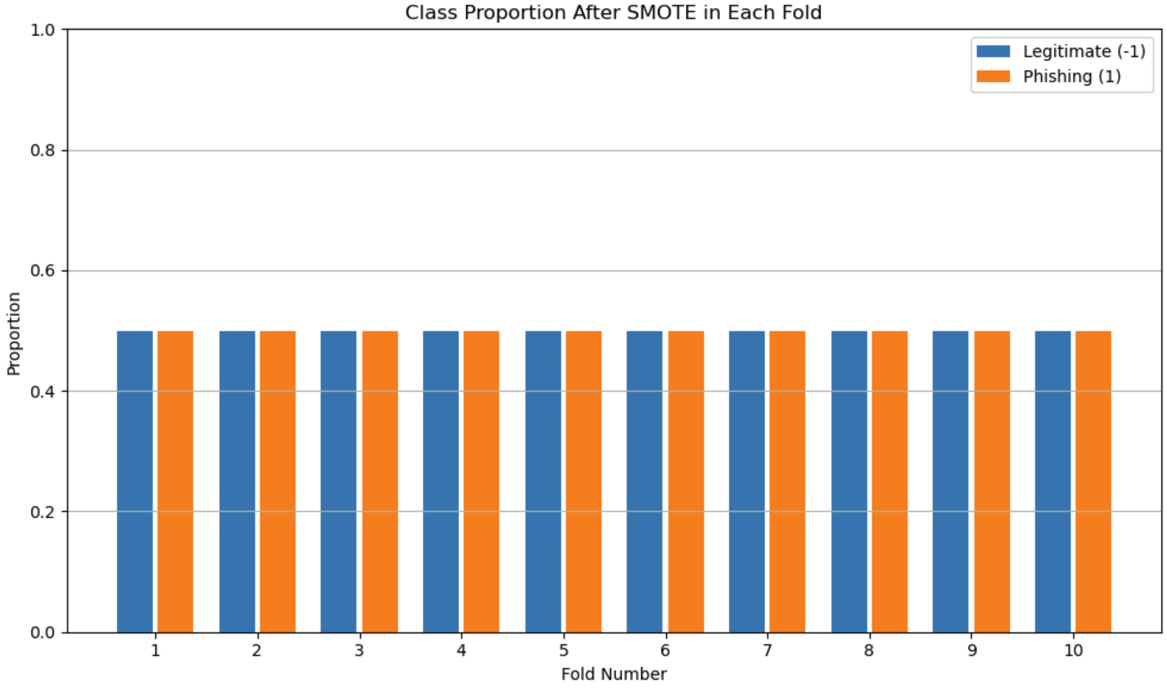
III. Methodology

A. Dataset and Preprocessing

This study uses the 'Phishing Website' dataset that was donated to the UC Irvine Machine Learning Repository that is publicly available for download. The dataset consists of 11055 instances which is made up of 30 engineered features spanning URL structure, DNS records, JavaScript behavior, and domain metadata. Given the imbalance in target class distribution (phishing vs. legitimate), I applied SMOTE (Synthetic Minority Oversampling Technique) to balance the dataset for training whilst not affecting the test set.

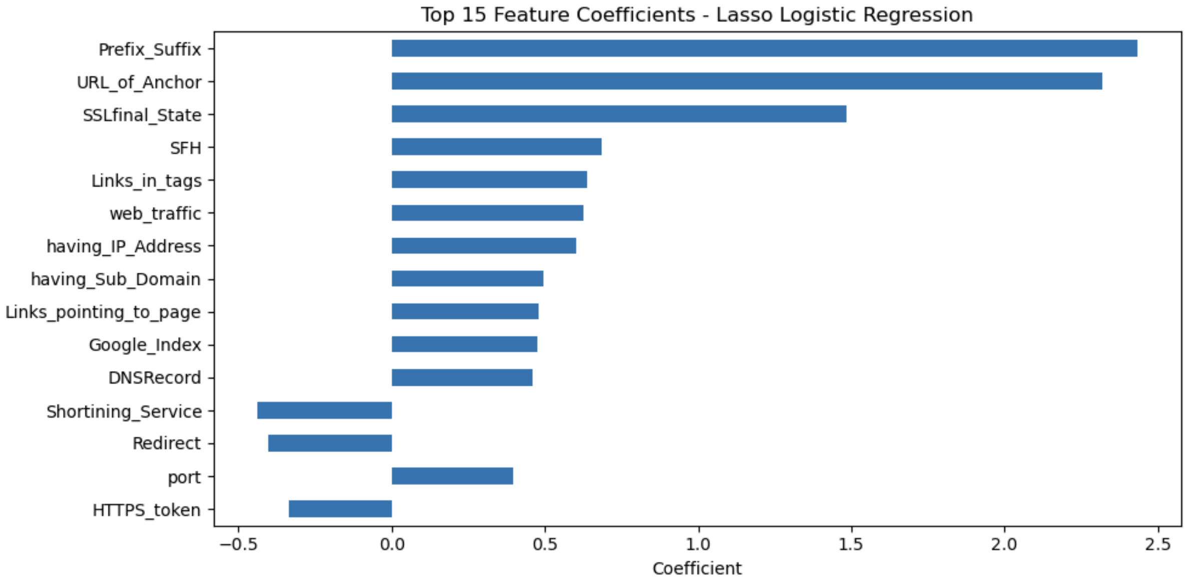


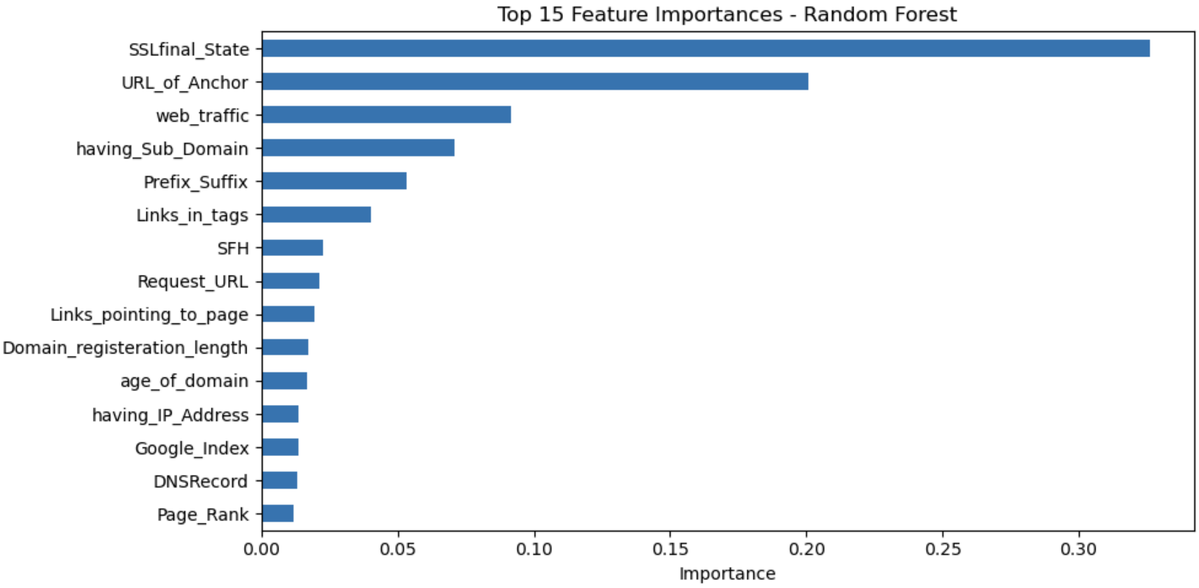
	proportion
Result	
1	0.556943
-1	0.443057



B. Feature Selection

I trained both Random Forest and Logistic (Lasso) Regression models by running them over 10-fold cross-validation. Top features were identified from both models and used to improve performance through further training. I selected both of these models for their strengths in feature selection (RF model) and the ability to shrink the relevant coefficients of features to zero (L1 Model)





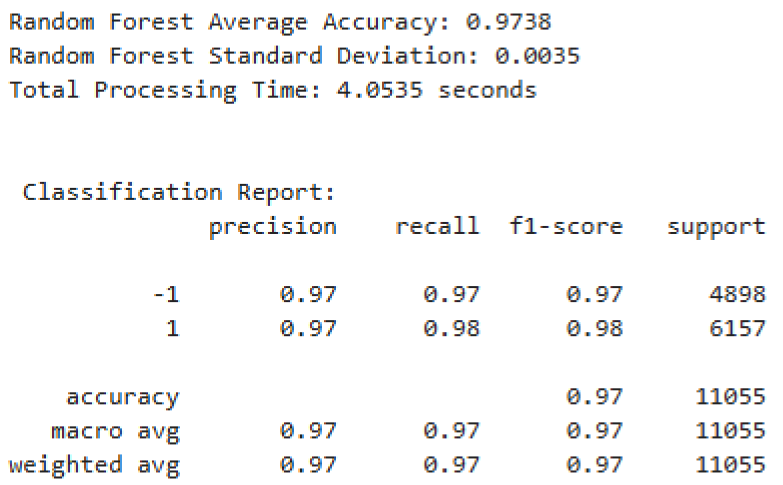
C. Challenges

One of the primary challenges in this project was ensuring real-world applicability. Developing effective phishing detection systems is inherently difficult due to the scarcity of high-quality, up-to-date, and authentic phishing datasets, which limits the model's ability to generalize beyond the training data [2].

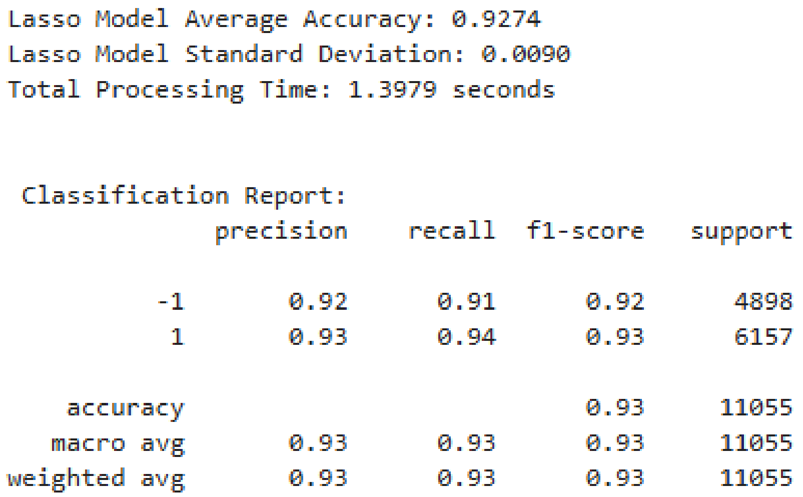
Another significant challenge involved preventing data leakage when using SMOTE (Synthetic Minority Oversampling Technique). As this was my first time working with SMOTE, particular care was taken to ensure oversampling was only applied to the training data. This precaution ensured that the test sets remained untouched by synthetic data, thereby preserving the integrity of the model.

IV. Results

To assess model performance, both Random Forest and Lasso Regression (Logistic Regression with L1 regularization) were evaluated using 10-fold cross-validation along side curated feature sets. The Random Forest model demonstrated superior performance, achieving a recall score of 0.98 compared to 0.94 for the Lasso Regression model. Despite the increased computational demand (approximately three times longer processing time), the Random Forest model was selected as the preferred classifier due to its enhanced detection capability which is essential in the use case of detecting Phishing sites.

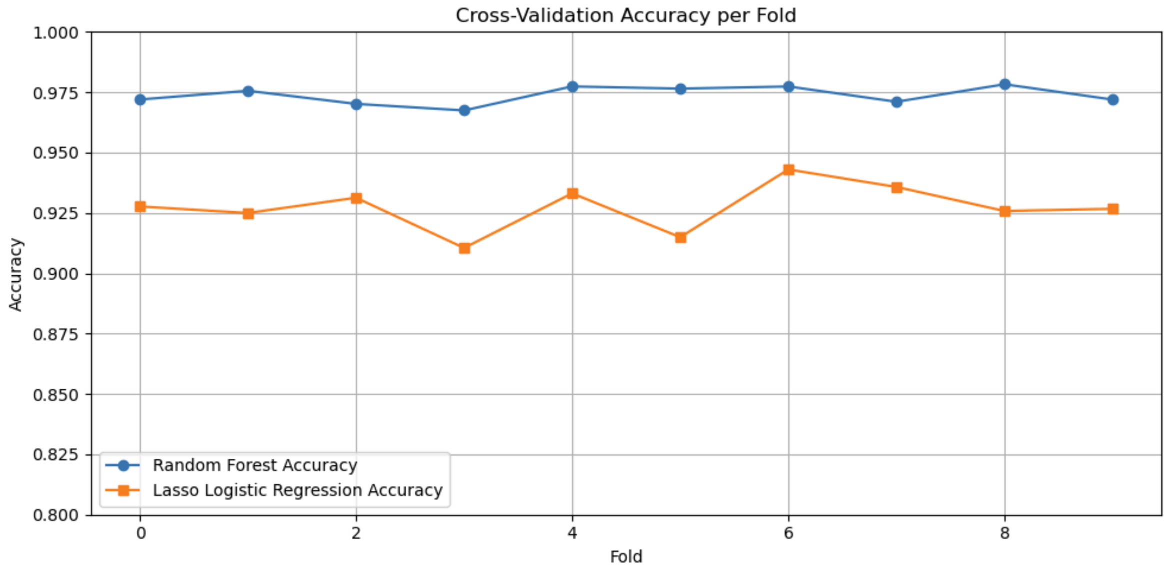


```
Random Forest Average Accuracy: 0.9738
Random Forest Standard Deviation: 0.0035
Total Processing Time: 4.0535 seconds


 Classification Report:
               precision    recall  f1-score   support

          -1       0.97      0.97      0.97      4898
           1       0.97      0.98      0.98      6157

    accuracy                           0.97     11055
   macro avg       0.97      0.97      0.97     11055
weighted avg       0.97      0.97      0.97     11055
```

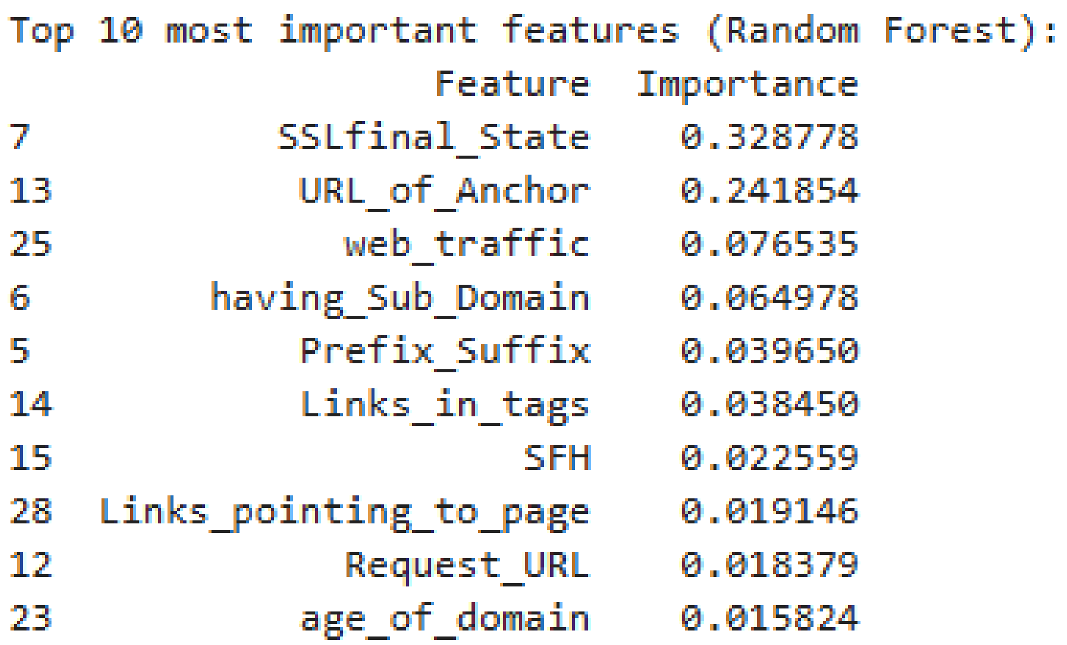


```
Lasso Model Average Accuracy: 0.9274
Lasso Model Standard Deviation: 0.0090
Total Processing Time: 1.3979 seconds


 Classification Report:
               precision    recall  f1-score   support

          -1       0.92      0.91      0.92      4898
           1       0.93      0.94      0.93      6157

    accuracy                           0.93     11055
   macro avg       0.93      0.93      0.93     11055
weighted avg       0.93      0.93      0.93     11055
```

Further analysis of cross-validation accuracy further supported this decision, as the Random Forest model consistently exhibited higher and more stable accuracy scores across folds than the Lasso model.

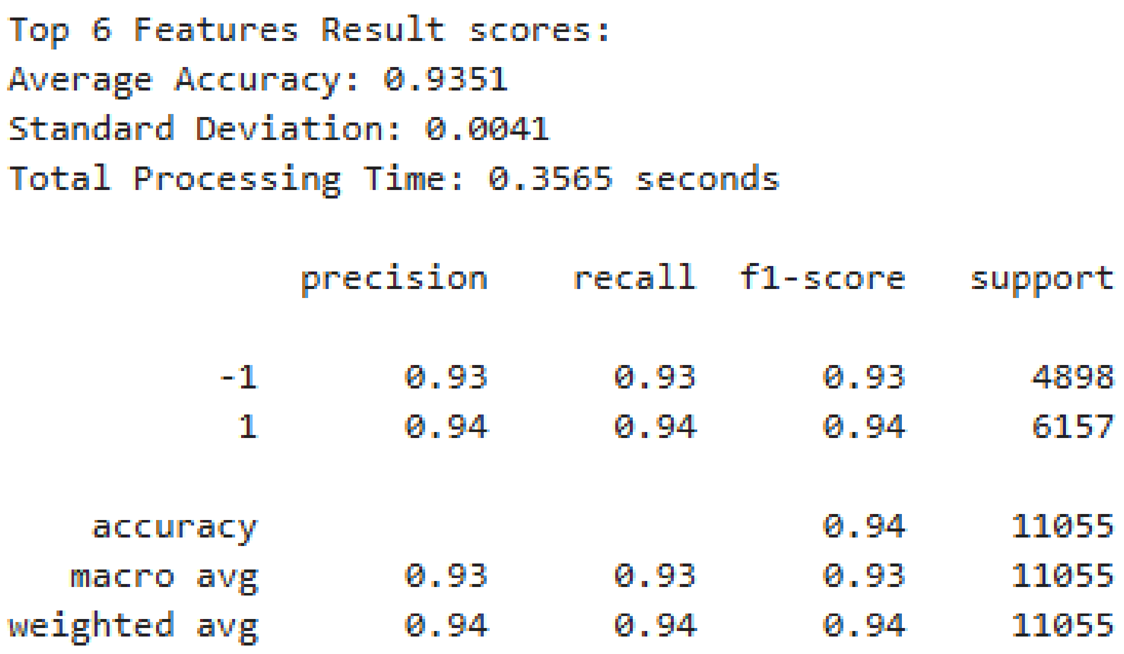


To improve computational efficiency while preserving model performance, feature importance rankings were extracted from the trained Random Forest model. The importance scores revealed a substantial drop after the 6th-ranked feature (Links_in_tags), indicating that many of the remaining features contributed minimally to the model's predictive power. Based on this observation, the six most important features were isolated to create a reduced dataset.



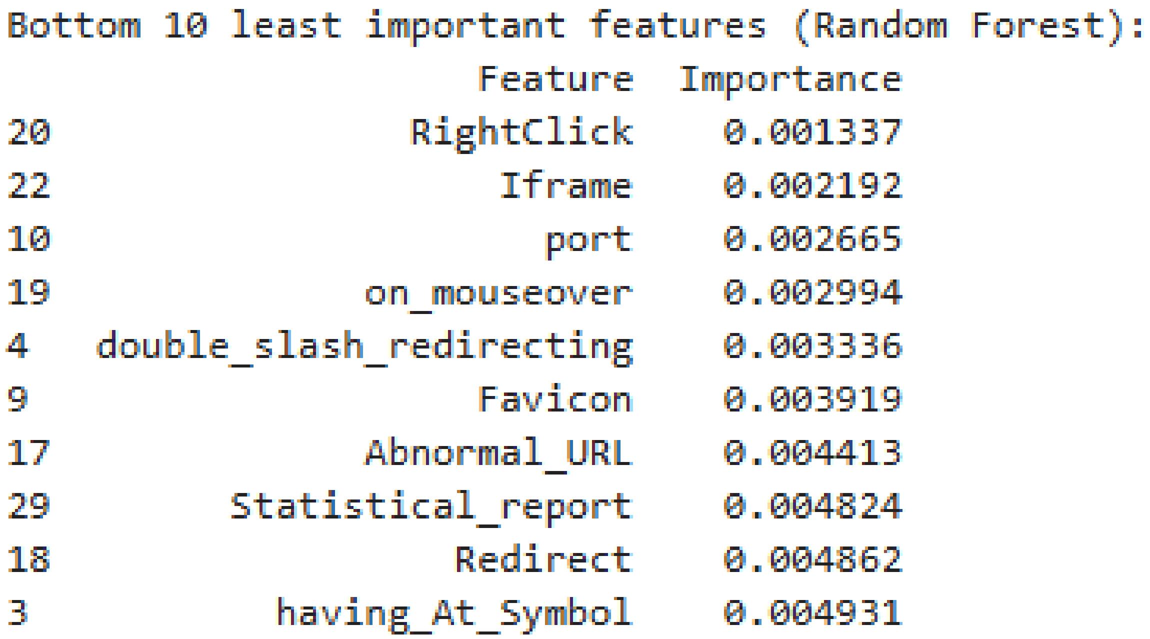
```
Top 10 most important features (Random Forest):
                   Feature  Importance
7             SSLfinal_State    0.328778
13             URL_of_Anchor    0.241854
25               web_traffic    0.076535
6          having_Sub_Domain    0.064978
5              Prefix_Suffix    0.039650
14             Links_in_tags    0.038450
15                       SFH    0.022559
28  Links_pointing_to_page    0.019146
12               Request_URL    0.018379
23             age_of_domain    0.015824
```

A notable decline in importance was observed after feature 15. (SFH) at a feature importance of 0.22559 compared to 14. Links_in_tags at 0.03850. So I decided to isolate the top 6 features into a new dataset to evaluate the scores that the Random Forest model can produce.



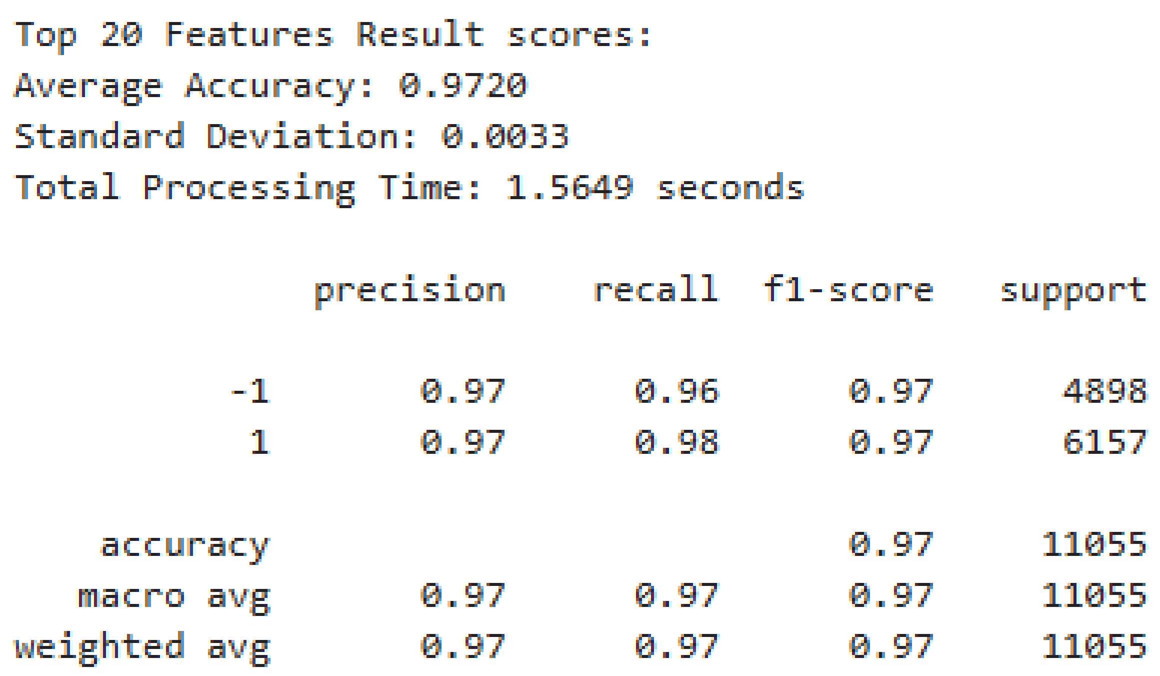
```
Top 6 Features Result scores:
Average Accuracy: 0.9351
Standard Deviation: 0.0041
Total Processing Time: 0.3565 seconds

              precision    recall  f1-score   support

          -1       0.93      0.93      0.93      4898
           1       0.94      0.94      0.94      6157

    accuracy                           0.94     11055
   macro avg       0.93      0.93      0.93     11055
weighted avg       0.94      0.94      0.94     11055
```

When re-evaluated on this subset, the Random Forest model maintained a recall of 0.94, comparable to the full-feature Lasso Regression model, while achieving a significant reduction in processing time (0.3565 seconds versus 1.3979 seconds for the full Lasso model). However, given the critical nature of phishing detection, even a 0.04 reduction in recall is considered a significant amount.



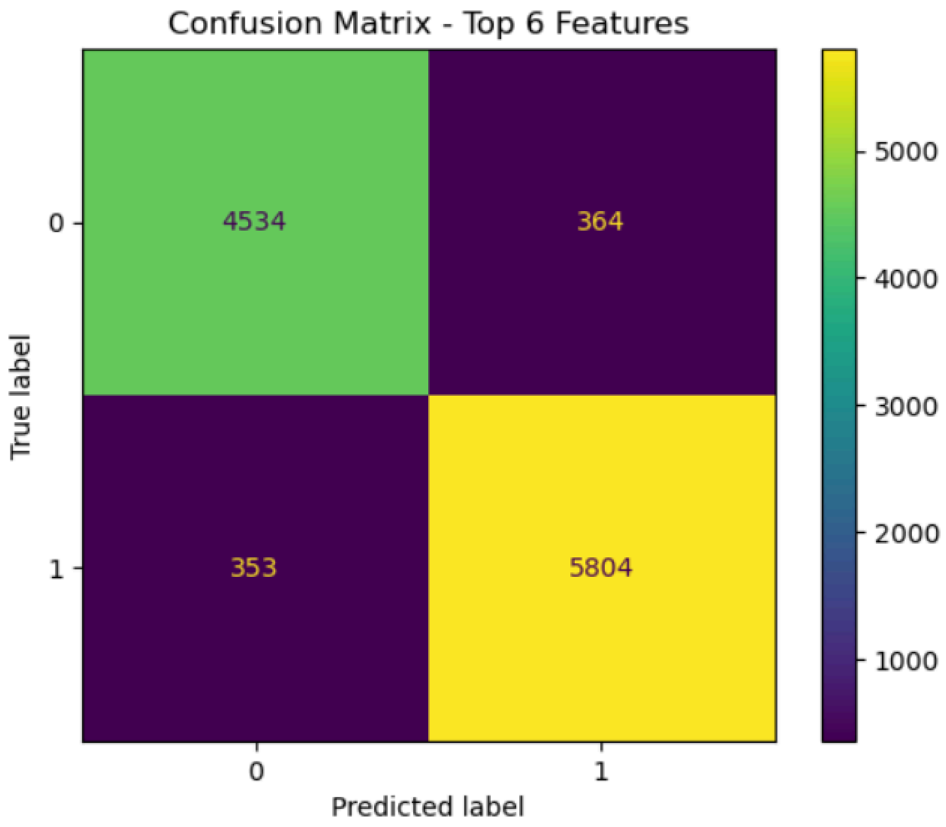
```
Bottom 10 least important features (Random Forest):
                     Feature  Importance
20                RightClick    0.001337
22                    Iframe    0.002192
10                      port    0.002665
19              on_mouseover    0.002994
4   double_slash_redirecting    0.003336
9                    Favicon    0.003919
17              Abnormal_URL    0.004413
29        Statistical_report    0.004824
18                  Redirect    0.004862
3            having_At_Symbol    0.004931
```

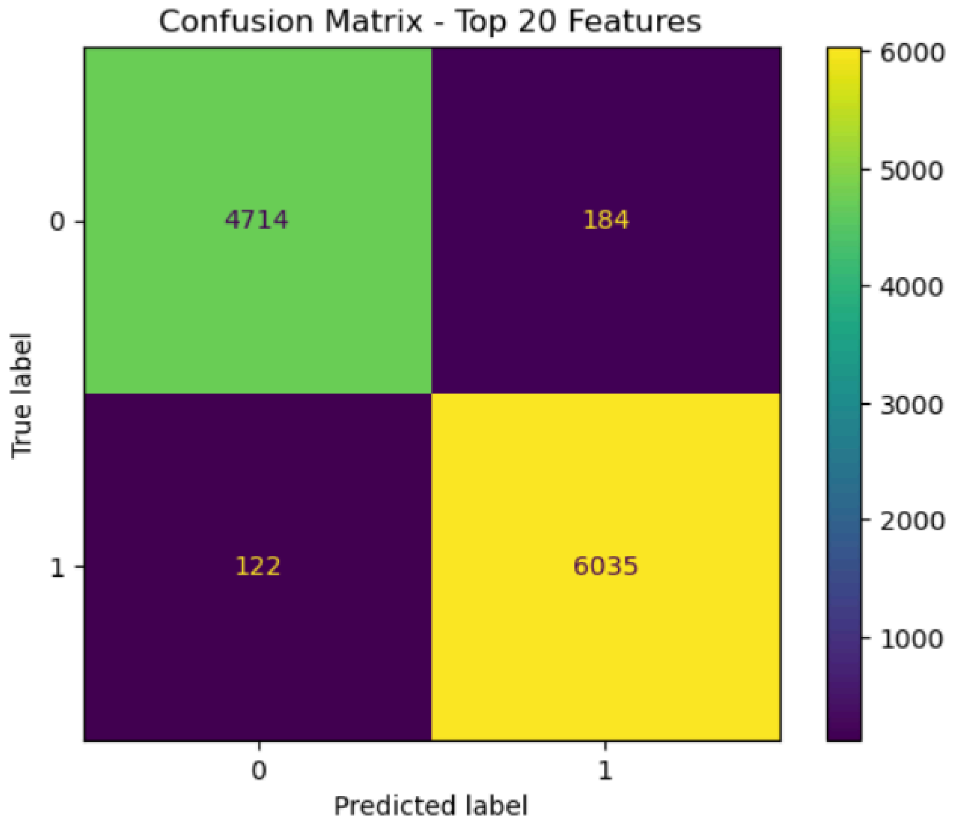
To address this, an alternative feature reduction strategy was implemented. Instead of retaining only the top features, the ten least important features were removed, resulting in a refined dataset with the 20 most influential features. Evaluation on this updated feature set yielded a recall of 0.98, equivalent to the original full-feature Random Forest model, but with a substantially reduced processing time of 1.5649 seconds (down from 4.0535 seconds). Although there was a slight decrease in average accuracy, the critical objective of maintaining high recall was preserved.



```
Top 20 Features Result scores:
Average Accuracy: 0.9720
Standard Deviation: 0.0033
Total Processing Time: 1.5649 seconds

              precision    recall  f1-score   support

          -1       0.97      0.96      0.97      4898
           1       0.97      0.98      0.97      6157

    accuracy                           0.97     11055
   macro avg       0.97      0.97      0.97     11055
weighted avg       0.97      0.97      0.97     11055
```

Comparing the two confusion matrices, the model using the top 20 features outperforms the top 6 feature model in both precision and recall. It reduces false negatives from 353 to 122 and false positives from 364 to 184, while maintaining high true positive and true negative counts. This improvement significantly lowers the false negative rate, once again crucial in phishing detection, while still offering faster processing than the full 30-feature model. As such, the 20-feature model was selected as the optimal balance between performance and efficiency.





These findings suggest that feature selection can meaningfully reduce computational cost while retaining strong detection performance, supporting the feasibility of real-time deployment scenarios.

V. Conclusion & Future Work

The Random Forest-based phishing detection system achieved a recall of 0.98, an overall accuracy of 0.9720, and a low standard deviation of 0.0033, all within a processing time of 1.5649 seconds. These results suggest the model is suitable for real-world deployment. SMOTE and feature selection were instrumental in achieving this performance.

For future work, hyperparameter optimization using GridSearchCV (with recall as the scoring metric) is my intended solution to further minimize false negatives and improve model efficiency.

References

[1]https://slashnext.com/press-release/slashnext-mid-year-state-of-phishing-report-shows-341-increase-in-bec-and-advanced-phishing-attacks/

[2]https://www.researchgate.net/publication/383342666_Challenges_of_Data_Collection_and_Preprocessing_for_Phishing_Email_Detection