BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

Phishing Site URLs Prediction

Shailesh Pratap Singh
SCSET
28/04/2024

Abstract

This project aims to develop a machine learning based model classifier keeping users' privacy and security in mind to classify websites either as safe or phishing based on their features. Using Dataset which Contains various websites with labeled as good or bad websites. it used leveraging algorithms like Logistic Regression, XG-Boost, and Random Forest where Random Forest gave the best accuracy of all the test and trained Model. We used features related to domain age, and SSL certificates were crucial in distinguishing between safe and phishing websites. in last seeing models' accuracy, precision rates and performance metrics during the evaluation separate datasets it concluded that machine learning techniques can be effectively used to identify the malicious websites, this project can help in enhanced measures for users.

Context:

With the Rapid Expansion of the internet in the daily world, cyber threat is also increasing day by day, like phishing attacks have become very popular these days, phishing attacks aim to deceive users' personal data and sensitive data through fraud websites. In today's digital world, when no one is safe, identification and safeguarding of users from phishers is essential.

Objectives:

The objective of this project is to make a model which uses machine learning and can differentiate between phishing and safe URLs with high accuracy. by analyzing various things of websites such as domain age, and SSL certificates, the model aims to identify phishing websites.

Scope:

The significance of this project is to enhance the cyber measures for internet users in daily life. it can automatically detect suspicious websites, this project can help people to fall into honey trap of phishing attacks, and lessen the financial losses and safeguarding the sensitive data.

Overview of the Project:

The project has been built across several stages: data collection, feature engineering, training and selection of models, evaluation, deployment. Each phase of this project includes specific responsibilities to effectively and quickly achieve objective. Additionally, this report will ensure transparency and facilitate all the knowledge transfer.

Literature Review:

Multiple studies and researches have taken on the issue of phishing detection using multiple and various techniques. Some have used machine learning algorithms such as logistic regression, decision trees, neural networks, and etc. while some others have focused on working on feature engineering and to find feature to identify phishing attacks. additionally, research in this field involves the analysis of website, website's content, URL Structure, domain characteristics, and user's behavior.

Existing Solutions and Their Limitations:

Currently present solutions try to catch phishing websites by involving their set of rules or use their list of known good and bad websites. but these methods can struggle to keep up with new tricks that phishers come up with. There's been some kind of success using Machine learning, where computer learn from examples, to spot the phishing websites. Plus, the we pick out important details from the websites to detect the safety might not work well with machines for catching phishing sites.

Gap Identification:

Project aim is to shift the users from the existing solutions to upcoming features of phishing attacks by using the advanced machine learning techniques and new features. by building a strong machine learning model which is capable of effectively differentiating between Phishing and Safe URLs, the project aim is to achieve the goal of developing the model to predict more accurately and be adaptable of cybersecurity measures in this digital world. Additionally, this project seeks to provide best solution so that it can enhance the efficiency of phishing website detection among users.

Approach:

this project uses machine learning approaches for phishing detection. it involves several steps for detection between Phishing and Safe URLs, i.e. data collection, preprocessing of data, feature engineering, model training and model evaluation.

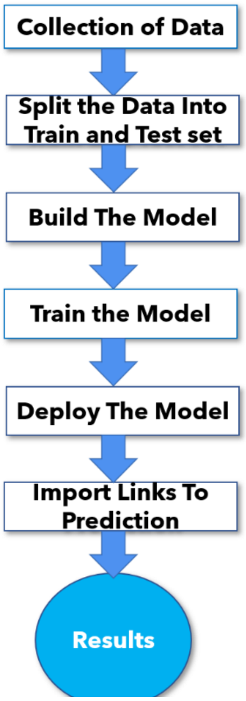
Tools & Technologies:

Programming Language: Python
Web Development: Flask
Machine Learning Libraries: Scikit-learn, pandas, NumPy and matplotlib
Model Deployment: Flask's development server

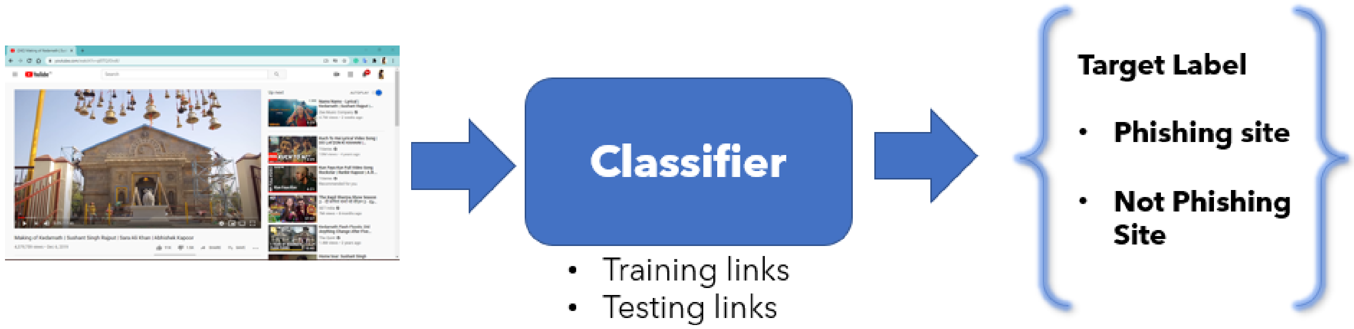
Implementation:

Data Collection: Phishing and Safe websites dataset is taken from Kaggle.
Feature Engineering: Features such as domain age, SSL certificate validity, and HTML content are engineered to capture the difference between Phishing and Safe websites.
Model Training: Machine learning features and libraries are used for feature selection, and model training (e.g., Random Forest), it is constructed and trained on the dataset.
Evaluation: The trained model's performance is evaluated using metrics.
Deployment: The trained model is deployed as a RESTful API using Flask, allowing users to input URLs and receive predictions regarding their input.

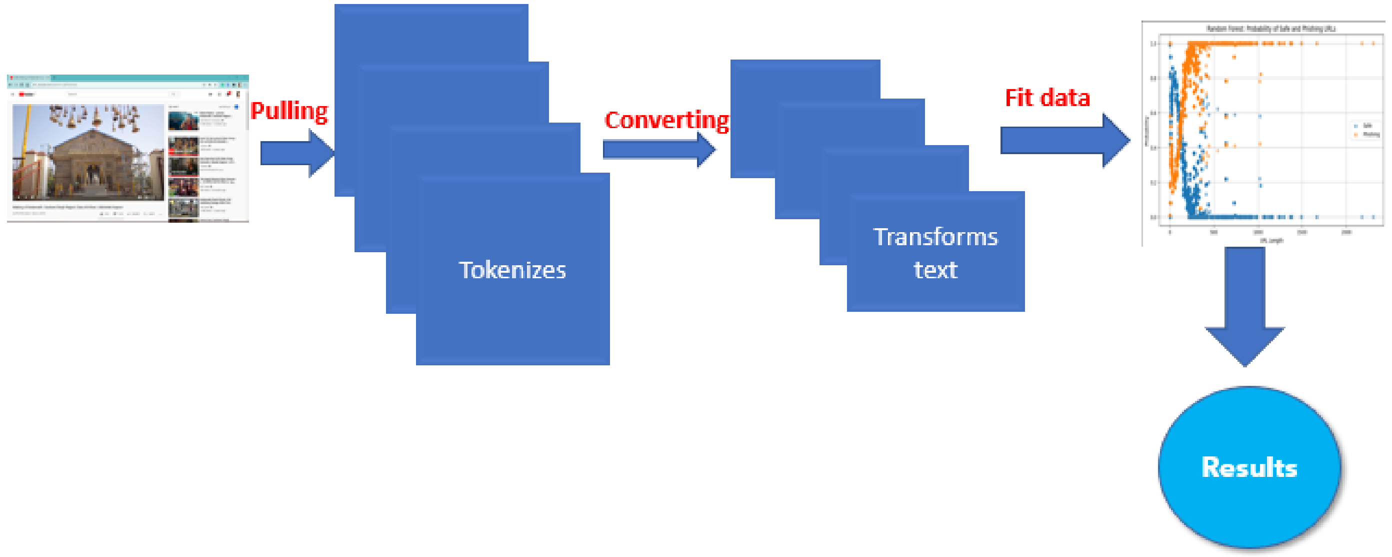
Implementation:



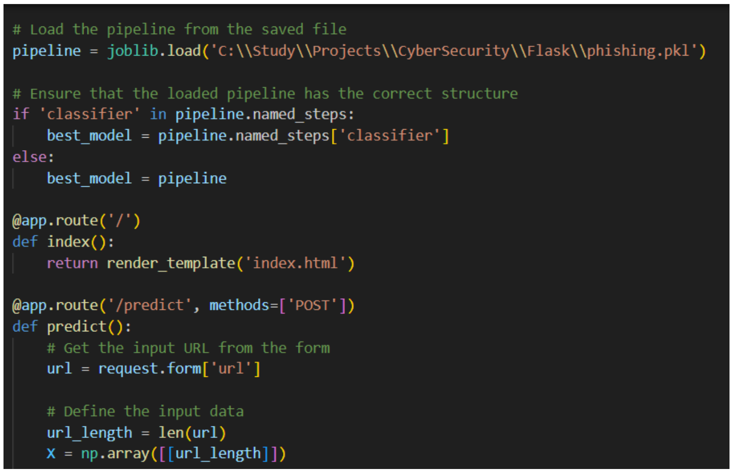
Overview:



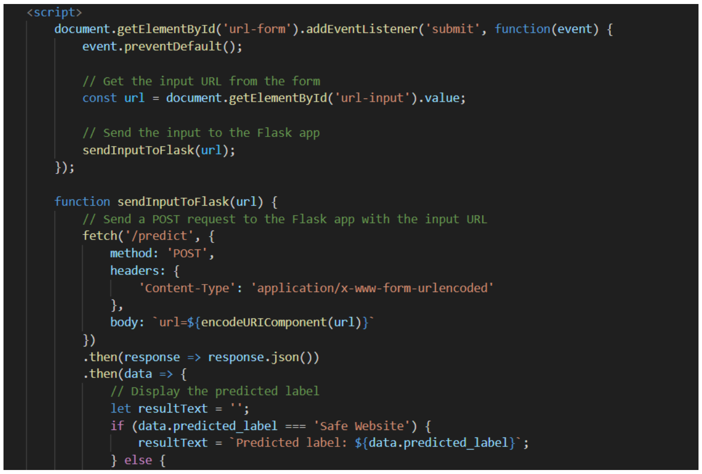
Working:



Code Snippets:

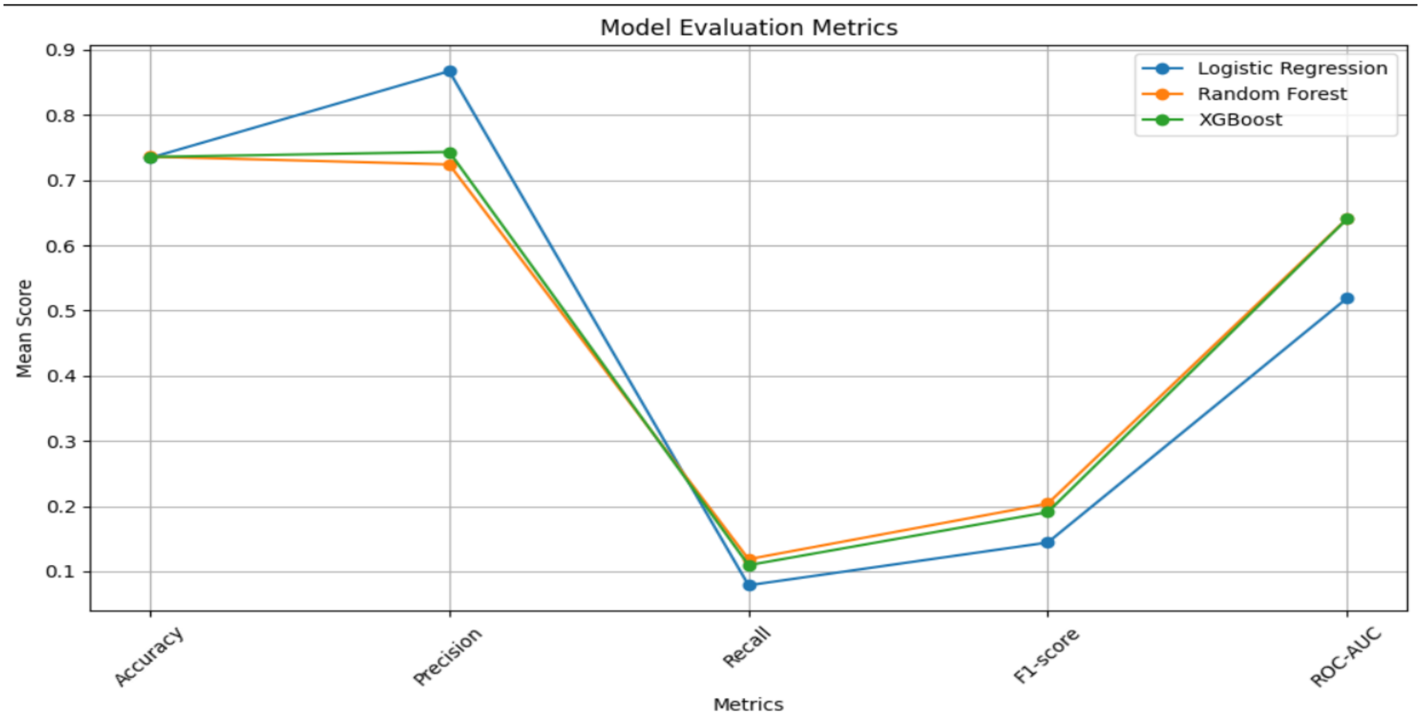


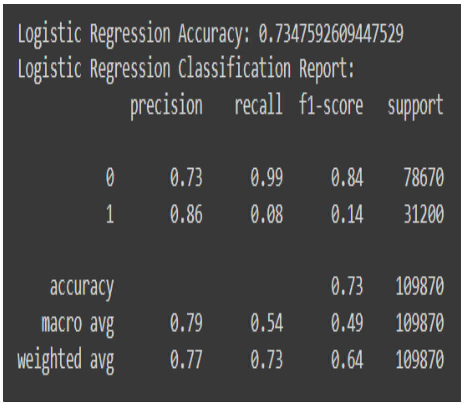
```
# Load the pipeline from the saved file
pipeline = joblib.load('C:\\Study\\Projects\\CyberSecurity\\Flask\\phishing.pkl')

# Ensure that the loaded pipeline has the correct structure
if 'classifier' in pipeline.named_steps:
    best_model = pipeline.named_steps['classifier']
else:
    best_model = pipeline

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    # Get the input URL from the form
    url = request.form['url']

    # Define the input data
    url_length = len(url)
    X = np.array([[url_length]])
```



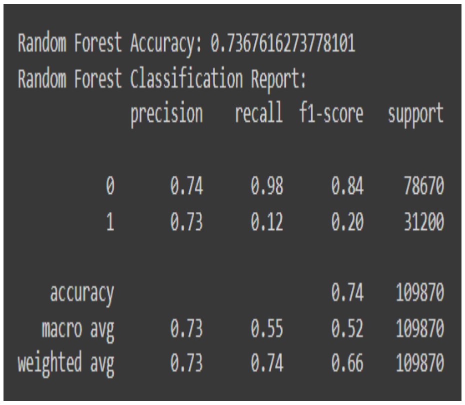
```
<script>
    document.getElementById('url-form').addEventListener('submit', function(event) {
        event.preventDefault();

        // Get the input URL from the form
        const url = document.getElementById('url-input').value;

        // Send the input to the Flask app
        sendInputToFlask(url);
    });

    function sendInputToFlask(url) {
        // Send a POST request to the Flask app with the input URL
        fetch('/predict', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/x-www-form-urlencoded'
            },
            body: `url=${encodeURIComponent(url)}`
        })
        .then(response => response.json())
        .then(data => {
            // Display the predicted label
            let resultText = '';
            if (data.predicted_label === 'Safe Website') {
                resultText = `Predicted label: ${data.predicted_label}`;
            } else {
```

Models Evaluation Matrics:

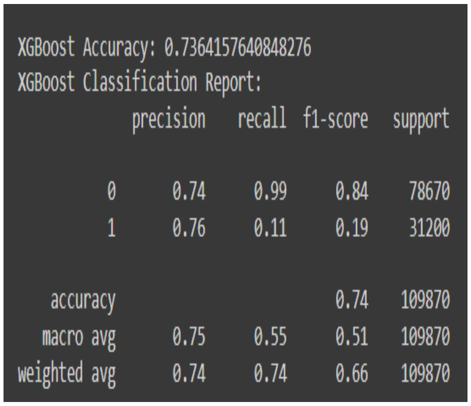




```
Logistic Regression Accuracy: 0.7347592609447529
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.99      0.84     78670
           1       0.86      0.08      0.14     31200

    accuracy                           0.73    109870
   macro avg       0.79      0.54      0.49    109870
weighted avg       0.77      0.73      0.64    109870
```

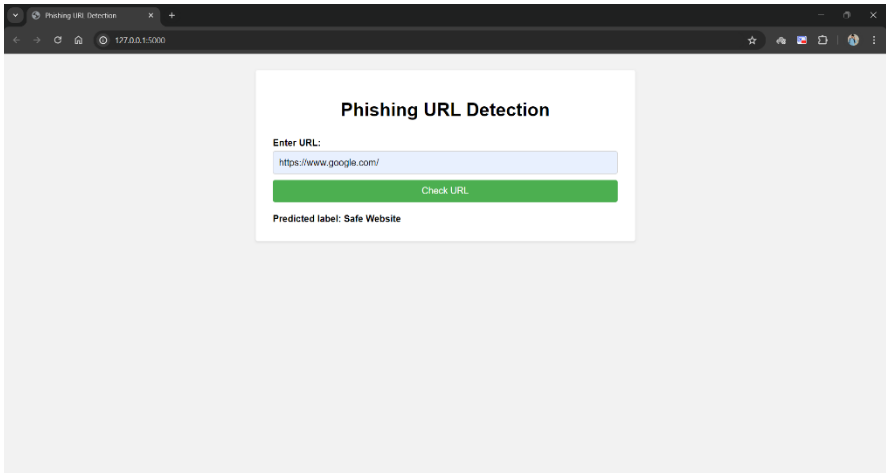


```
Random Forest Accuracy: 0.7367616273778101
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.98      0.84     78670
           1       0.73      0.12      0.20     31200

    accuracy                           0.74    109870
   macro avg       0.73      0.55      0.52    109870
weighted avg       0.73      0.74      0.66    109870
```



```
XGBoost Accuracy: 0.7364157640848276
XGBoost Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.99      0.84     78670
           1       0.76      0.11      0.19     31200

    accuracy                           0.74    109870
   macro avg       0.75      0.55      0.51    109870
weighted avg       0.74      0.74      0.66    109870
```

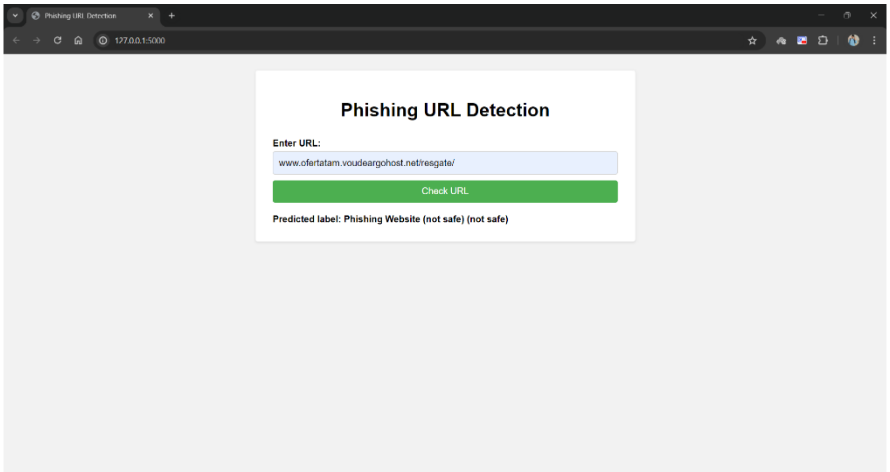
Output:



Phishing URL Detection

Enter URL: https://www.google.com/

Predicted label: Safe Website



Phishing URL Detection

Enter URL: www.ofertatam.voudeargohost.net/resgate/

Predicted label: Phishing Website (not safe) (not safe)

Conclusion

- The model that I created does an excellent job of differentiating between phishing and safe websites.
- The age of the domain and the validity of the SSL certificate are two features that helped in classification a lot.
- This initiative showed the true value of utilizing machine learning to better the digital environment and get safe from cyberattacks.

Future Plans:

- Look into more features to improve the model's accuracy.
- Real-time monitoring allows you to update the model with knowledge about new hazards.
- Collaborating with cybersecurity professionals to increase model validation.

Overall, this experiment shows the importance of machine learning in preventing from cyberattacks and provided the further information for future research and development in this area.

GitHub repo: https://github.com/shailesh-se/Phishing_URL_Prediction.git
Demo: https://youtu.be/DYScz60xTRA