Project Report: Phishing URL Detection App

---

1. Introduction

Phishing attacks pose a significant threat in today's digital world, where malicious actors use deceptive URLs to steal sensitive information. The **Phishing URL Detection App** is designed to combat this by offering an easy-to-use tool that utilizes machine learning to analyze and detect potential phishing URLs. The application integrates a trained machine learning model with a user-friendly interface to provide quick and accurate predictions.

---

2. Dataset and Model Training

Dataset Overview

The dataset used in this project includes various URLs labeled as either "phishing" or "legitimate." Key features extracted from these URLs include:

- **URL Characteristics:** Length of the URL, hostname length, and presence of special characters such as dots, hyphens, and slashes.
- **Security Indicators:** Presence of IP addresses in URLs, suspicious tokens, and character composition analysis.
- **Domain-Based Attributes:** Analysis of subdomains, top-level domains (TLDs), and domain reputation.
- **Text and Link-Based Metrics:** Examination of the textual components and number of hyperlinks in the URL.

The dataset contains a `status` column that classifies URLs as either "phishing" or "legitimate," enabling the model to learn distinguishing patterns.

Model Training Process

1. Data Preparation:

- The dataset is loaded into a Pandas DataFrame.
- Missing or irrelevant data is removed.
- Non-numeric features such as raw URLs are eliminated to focus on numerical analysis.

- The target variable (**status**) is converted into binary values (1 for phishing, 0 for legitimate).

2. Feature Engineering & Model Selection:

- The dataset is split into training and testing sets.
- A **Random Forest Classifier** is selected due to its ability to handle high-dimensional data effectively.
- The model is trained and evaluated using accuracy metrics.
- After achieving a satisfactory performance, the trained model is saved using Joblib for later use.

3. Saving the Model:

- The trained model is stored as **phishing_model.pkl**, allowing the application to load it for predictions without re-training.
- A list of phishing and legitimate URLs is pre-extracted from the dataset for quick lookup.

---

3. How the App Works: UI and Model Integration

User Interface (UI) Design

The app is built using **Streamlit**, which provides an interactive and intuitive web interface. The key UI elements include:

- **URL Input Field:** Users enter a URL for analysis.
- **Scan Button:** Triggers the detection process when clicked.
- **Result Display:** Shows the prediction and confidence score.
- **Background Styling:** Uses a custom background image (`matrix.jpg`) and CSS for a polished look.

Prediction Workflow

1. **Dataset Check:**
    - If the entered URL exists in the **phishing dataset**, the app immediately returns **"Phishing"** with **100% confidence**.
    - If the URL is found in the **legitimate dataset**, it returns **"Legitimate"** with **100% confidence**.
2. **Feature Extraction & Model Prediction:**

    - If the URL is not in the dataset, it undergoes feature extraction (a placeholder function is used, which can be replaced with a more advanced method).
    - The extracted features are processed by the **trained Random Forest model**.
    - The model predicts the classification (Phishing or Legitimate) along with a confidence score.
3. **Displaying Results:**
    - **Phishing URLs:** Displayed with a red warning message.
    - **Legitimate URLs:** Displayed with a green success message.
    - **Confidence Score:** Shown to indicate model certainty.
4. **Developer Credits:**
    - The developer's name is displayed in a bold, cursive style at the bottom of the page.

---

4. Examples

Screenshots of the Application

- **Home Screen:**
- **Prediction Result (Legitimate):**
- **Prediction Result (Phishing):**

(Replace the paths above with actual screenshot file locations.)

---

5. Conclusion

This **Phishing URL Detection App** successfully integrates machine learning with a web-based UI to detect phishing URLs efficiently. The combination of a pre-checked dataset and a trained model ensures both quick and accurate predictions. The application is designed to help users stay safe online by identifying potentially harmful links before they can cause damage.

For further improvements, additional feature extraction techniques and deep learning models can be explored to enhance prediction accuracy. Contributions and suggestions are always welcome!

Developed by:

Pradeep Bhattarai