Problem Statement: Phishing URL Detection

Objective: Develop a robust phishing URL detection model using machine learning techniques .

Dataset Description:

- **Dataset Type:** Tabular
- **Associated Tasks:** Classification
- **Feature Types:** Real, Categorical, Integer
- **Number of Instances:** 188690
- **Number of Features:** 54

Features Available:

- `URLLength`: Integer representing the length of the URL.
- `Domain`: Categorical feature indicating the domain of the URL.
- `DomainLength`: Integer representing the length of the domain name.
- `IsDomainIP`: Binary integer indicating if the domain is represented as an IP address.
- `TLD`: Categorical feature denoting the top-level domain of the URL.
- `URLSimilarityIndex`: Integer measuring similarity to known phishing or legitimate URLs.
- `CharContinuationRate`: Integer indicating character continuation rate in the URL.
- `TLDLegitimateProb`: Continuous feature representing probability of the TLD being legitimate.

Target Variable:

- `Label`: Binary variable where 1 denotes a legitimate URL and `0` denotes a phishing URL.

Task:

Your task involves conducting thorough exploratory data analysis (EDA) to understand feature distributions, handle missing values, and visualize relationships between features and the target variable (Label). Subsequently, select pertinent features that effectively differentiate between phishing and legitimate URLs, possibly engaging in feature engineering to enhance model performance. Following feature selection, apply an appropriate classification approach, split the dataset into training and testing sets, and train the model on the training data. Evaluate the model's performance on the testing set to ensure it correctly predicts whether a URL is phishing or legitimate based on the selected features.

Instructions:

- Use Google Colab or Jupyter Notebook for coding. Extract and load the dataset provided in a zip folder for analysis.
- Write original code; direct copy-pasting will result in **disqualification**. Ensure your code is well-structured with clear comments explaining each step.
- Be ready to explain the mathematical concepts and basic workings of your code during evaluation.
- Submit your finalized Notebook (.ipynb) file by **July 27, 2024**. Late submissions will not be considered. Provide the link to your .ipynb file on the shared submission sheet.