PROJECT: Machine Learning

Phishing Detector using LR

Purpose of the Document

The document has to specify the requirements for the project "Build a detector for Phishing websites (LR)." Apart from specifying the functional and non-functional requirements for the project, it also serves as an input for project scoping.

Problem Statement

The purpose of the project is to use one or more of the classification algorithms to train a model on the Phishing website dataset.

You are provided with the following resources that can be used as inputs for your model:
1. A collection of website URLs for 11000+ websites. Each sample has 30 website parameters and a class label identifying it as a phishing website or not (1 or -1).
2. Code template containing these code blocks:
a. Import modules (Part 1)
b. Load data function + input/output field descriptions

You are expected to write the code for a binary classification model (phishing website or not) using
Python Scikit-Learn that trains on the data and calculates the accuracy score on the test data.

Implementation Details

Step 1: Load the dataset

- Loaded the dataset into the program using package pandas and extracted input (X) and output (y) data from the dataset.

Step 2: Split the features and label into training and testing data

- Performed splitting of dataset into training data (70%) and testing data (30%) using train_test_split.

Step 3: Standardize data

- Standardized data using Standard Scaler.

Step 4: Perform Logistic Regression Classifier.

- Implemented Logistic Regression Classifier on training data with "C" parameter = 100 and printed the predicted the output.
- Printed Confusion Matrix to specify the count of misclassified samples in the test data prediction.

Step 5: Extracted input (X: Prefix_Suffix and URL_of_Anchor) and output (y) data from the dataset with index 5.

Step 6: Split the features and label into training and testing data

- Performed splitting of dataset into training data (70%) and testing data (30%) using train_test_split.

Step 7: Standardize data

- Standardized data using Standard Scaler.

Step 8: Perform Logistic Regression Classifier.

- Implemented Logistic Regression Classifier on training data with "C" parameter = 100 and printed the predicted the output.
- Printed Confusion Matrix to specify the count of misclassified samples in the test data prediction.

Step 9: Visualize the outcome of Logistic Regression Classifier Test set.

Step 10: Performed Step 5 to Step 9 with index 13.