PROJECT REPORT

Introduction:

Phishing is a form of fraud in which the attacker tries to steal sensitive information of the victim. Usually, the victim receives a message that appears to have come from a known contact or a known organization. This message might contain malicious software that might harm the victim's computer or contain links that lead to phony websites.

Acc. to a report by Webroot, cyber criminals are creating an average of around 1.4 million phishing websites every month. They create fake pages designed to mimic the company they're spoofing and then replace them within hours to ensure they're not caught.

For example, attackers might intimidate victims by suggesting that their account is being closed, an invoice is waiting, or even in some cases, they've been summoned to court. In each instance, the victim might panic and click through to the malicious site which will either steal their credentials or drop a malicious payload.

Therefore, it is highly important to catch such activities and stop the attack on innocent people.

Project Definition:

My goal is to detect if a given URL is either malicious or safe using Multi-Layer Perceptron.

Our project was split into the following tasks:

1) Select top features from the dataset for use in training using feature selection.
2) Train a model using the top features to recognize the originality of URLs, and then use it to classify the test set values into the appropriate class.
3) Determine true positives, true negatives, false positives and false negatives. Using this information, compute classification accuracy, precision and recall scores.
4) Plot confusion matrix.

Implementation:

Data:

The data was obtained from the UCI machine Learning repository. It was collected in 2015. It is collection of 2456 URLs. The dataset consist of URLs that are legitimate, suspicious and malicious.

30 attributes were extracted from every URL. These attributes can be grouped into four main classes which are:

- Address bar based features
- Abnormal based features
- HTML and JavaScript based features
- Domain based features

For the project, 80% of the data was used for training and 20% was used for testing.

Feature Selection:

After obtaining the data, selecting the top few features instead of all the features for classification gives a better accuracy because of better training.

Hence, in this project, Logistic Regression with recursive feature elimination was used.

Recursive Feature Elimination (RFE) works by recursively removing attributes and building a model on those attributes that remain. It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute.

Logistic Regression works best when there are no outliers in the dataset and there is no high correlation between the data values. The dataset used in this project agrees with the above-mentioned cases and hence logistic regression is used with RFE.

Both the algorithms were taken from scikit-learn library.

Classification:

The top selected features are used to classify the data into either malicious or safe class. For classification, multi-layer perceptron has been used.

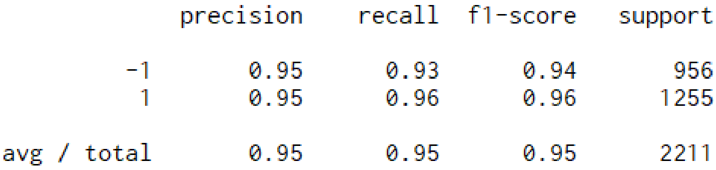
A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable by projecting the data into a space where the data is linearly separable and then classifying it.

2 hidden layers with 5 units in the first layer and 2 units in the second layer were used.

The classifier was obtained from scikit-learn library.

<u>Results:</u>

The model provided an accuracy of 94.979% and the following precision, recall and f-1 scores:

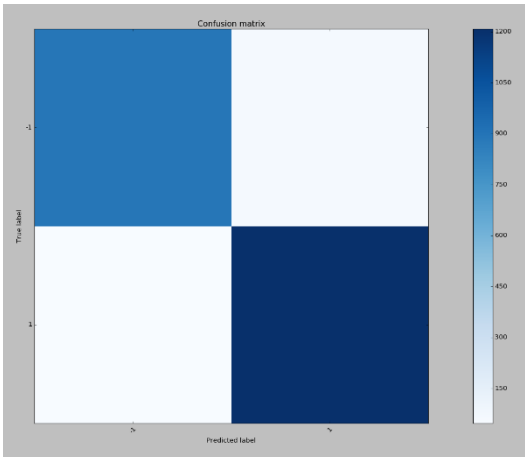


```
             precision    recall  f1-score   support

         -1       0.95      0.93      0.94       956
          1       0.95      0.96      0.96      1255

avg / total       0.95      0.95      0.95      2211
```

The top features selected were: having IP address, shortening service, prefix suffix, having sub-domain, SSL Final State, port, HTTPS token, URL Anchor, Links in Tags, SFH, Redirect, DNS Record, web traffic, google index, links pointing to page. These features contributed the most in training the model.



The obtained Confusion Matrix:

```
array([[ 892,   64],
       [ 47, 1208]], dtype=int64)
```

The explanation of the confusion matrix is as follows:

- 892 instances were correctly classified as ‘-1’ when the actual value was ‘-1’.
- 64 instances were incorrectly classified as ‘1’ when the actual value was ‘-1’.
- 47 instances were incorrectly classified as ‘-1’ when the actual was ‘1’.
- 1208 instances were correctly classified as ‘1’ when the actual was ‘1’.

Conclusion and Future Work:

- The software developed correctly identifies phishing websites with 94.979% accuracy.
- The data used here already had attributes extracted from the URLs. I would like to work on data directly from the messages received by victims and try to extract other important features hidden in the URLs.
- The data used in this project is offline. I would like to take it further and implement real-time detection.
- Other scope includes exploring classifiers beyond Multi-layer perceptron for classification like Random Forests or SVM.