**Cyber Security Capstone Project**

*Submitted By: Srividya Sundaravadivelu*

Table of Contents

[Problem Statement: 1](#_Toc102602670)

[Importing the libraries: 1](#_Toc102602671)

[Data Retrieval 1](#_Toc102602672)

[Data Exploration 1](#_Toc102602673)

[1. Number of Samples 1](#_Toc102602674)

[2. Check for null values 1](#_Toc102602675)

[3. Get the unique values 1](#_Toc102602676)

[4. Heat map 1](#_Toc102602677)

[5. Remove highly correlated features 1](#_Toc102602678)

[6. Remove weakly correlated features to target 1](#_Toc102602679)

[7. Check for class imbalance 1](#_Toc102602680)

[8. Histogram 1](#_Toc102602681)

[9. Count plot 1](#_Toc102602682)

[10. Insights from EDA 1](#_Toc102602683)

[Model Building 1](#_Toc102602684)

[Model 1: SGD Classifier 1](#_Toc102602685)

[1. Performance metrics for Train Data 1](#_Toc102602686)

[i. Accuracy 1](#_Toc102602687)

[ii. Confusion Matrix 1](#_Toc102602688)

[iii. Precision, Recall and F1 Score 1](#_Toc102602689)

[iv. ROC Curve 1](#_Toc102602690)

[v. AUC 1](#_Toc102602691)

[2. Performance metrics for Test Data 1](#_Toc102602692)

[Model 2: Logistic Regression 1](#_Toc102602693)

[1. Performance metrics for Train Data 1](#_Toc102602694)

[i. Accuracy 1](#_Toc102602695)

[ii. Confusion Matrix 1](#_Toc102602696)

[iii. Precision, Recall and F1 Score 1](#_Toc102602697)

[iv. ROC Curve 1](#_Toc102602698)

[v. AUC 1](#_Toc102602699)

[2. Performance metrics for Test Data 1](#_Toc102602700)

[Model 3: Random Forest Classifier 1](#_Toc102602701)

[1. Performance metrics for Train Data 1](#_Toc102602702)

[i. Grid Search CV 1](#_Toc102602703)

[ii. Accuracy 1](#_Toc102602704)

[iii. Confusion Matrix 1](#_Toc102602705)

[iv. Precision, Recall and F1 Score 1](#_Toc102602706)

[v. ROC Curve 1](#_Toc102602707)

[vi. AUC 1](#_Toc102602708)

[2. Performance metrics for Test Data 1](#_Toc102602709)

[3. Feature Importance 1](#_Toc102602710)

[Conclusion 1](#_Toc102602711)

# Problem Statement:

Book-My-Show will enable the ads on their website, but they are also very cautious about their user privacy and information who visit their website. Some ads URL could contain a malicious link that can trick any recipient and lead to a malware installation, freezing the system as part of a ransomware attack or revealing sensitive information. Book-My-Show now wants to analyze that whether the particular URL is prone to phishing (malicious) or not.

Using Machine Learning techniques, we need to predict if a given URL is “Phishing” or “Legitimate”.

# Importing the libraries:

We will be using several Python libraries and frameworks specific to Machine Learning. We have imported the libraries for pandas, numpy, scipy, and scikit-learn which are used for data processing and Machine Learning. We also use matplotlib and seaborn for exploratory data analysis and visualizations.

Text

Description automatically generated

# Data Retrieval

The input dataset contains an 11k sample corresponding to the 11k URL. Each sample contains 32 features that give a different and unique description of URL ranging from -1,0,1.

 1: Phishing

 0: Suspicious

 1: Legitimate

The sample could be either legitimate or phishing. We downloaded the dataset using pandas into a data frame.

Table

Description automatically generated

# Data Exploration

Exploratory data analysis is done to explore and understand the given Cyber Security dataset. We do feature selection to include only the relevant features. Features that are least correlated to dependent variable are discarded. Also, independent variables that are highly correlated among themselves are also discarded as they denote redundancy.

## Number of Samples

A picture containing graphical user interface

Description automatically generated

The dataframe has 11055 rows and 32 columns. "Result" column is the y variable (Dependent variable). Since it is labelled data, we are dealing with a supervised learning problem.

## Check for null values

Table

Description automatically generated

There are no null values in any of the columns. All the columns are of type integer.

## Get the unique values

Graphical user interface

Description automatically generated with medium confidence

All the columns have only 2 or all of the values : -1, 0 and 1.

-1: Phishing 0: Suspicious 1: Legitimate

The "Result" column has 2 possible outcomes -1 and 1 where -1 is Phishing and 1 is Legitimate.

The "Result" is the dependent variable which we must predict. This is the "y" variable. Since the "Result" has 2 classes, this is a Binary classification problem.

## Heat map

Chart

Description automatically generated

From the heatmap, its evident that "popUpWindow" and "Favicon" are having strong correlations, since the color is almost white(closer to 1.0). This suggests data redundancy. One of these columns must be removed.

## Remove highly correlated features

A picture containing graphical user interface

Description automatically generated

We need to find the independent variables which are highly correlated among themselves. This means there is redundant data. We need to drop these redundant columns. Lets set the threshold as 0.75 and select the columns that are having correlation greater than 0.75. These columns must be dropped from the dataframe.

Graphical user interface, text, application

Description automatically generated

The above code returned five columns which are highly correlated among themselves. We will be dropping these columns from the dataframe.

## Remove weakly correlated features to target

Table

Description automatically generated

"Result" is having strong correlation with "SSLfinal\_State" and URL\_of\_Anchor. 'Redirect', 'RightClick', 'Iframe', 'Favicon' are having a weak correlation with "Result".

We dropped the columns which have correlation less than or equal to 0.03. These columns have very little significance to the Result column.

After dropping the columns with correlation <= 0.03, we are left with 21 columns.

## Check for class imbalance

Graphical user interface, text, application, chat or text message

Description automatically generated

Chart, bar chart

Description automatically generated

We can see that around 55 % of values belong to class 1 and 44 percentage of values belong to class -1. We don’t have the imbalanced class problem here.

## Histogram

A picture containing text, crossword puzzle

Description automatically generated

Chart, bar chart

Description automatically generated

Above histograms shows the data distribution of all the columns in the dataset. We can clearly  see that the unique values are -1,0 and 1.

## Count plot

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

## Insights from EDA

1. From the above graphs, we can clearly see that when SSLFinal\_State is Legitimate, the "Result"  is mostly "Legitimate". When the SSLFinal\_State is "Phishing", the "Result" is also mostly "Phishing". This means we can see a strong positive correlation between SSLFinalState and Result columns.

2. Also, when the URL\_of\_Anchor is "Legitimate", the "Result" is most of the time "Legitimate".  When the URL\_of\_Anchor is "Phishing", the "Result" is also "Phishing" most of the time. Hence there is a positive correlation between URL\_of\_Anchor and Result.

3. The web\_traffic and Result seem to be having a positive correlation. When the web\_traffic is "Legitimate", the "Result" is mostly "Legitimate".

# Model Building

## Model 1: SGD Classifier

Graphical user interface, text

Description automatically generated

### Performance metrics for Train Data

Let’s use K Fold cross validation to evaluate the model. Here we divide the train data into 5 folds and reserve 1 for validation and 4 for training. We continue the steps till all the folds participate in both training and validation. By this way, we can validate the model more accurately and keep the test data untouched till the model is ready and finished training. Also, cross validation resolves the problem of over-fitting.

#### Accuracy

Graphical user interface, text, application, Word, website

Description automatically generated

We get around 91 % accuracy in all the 5 folds. But for a classification problem, accuracy is not a good measure. There are other measures like Confusion Matrix, Precision, Recall and F1 score.

#### Confusion Matrix

Text

Description automatically generated

There are 345 + 276 = 621 wrong predictions. There are 3054 + 4063 = 7117 right predictions. Accuracy is 7117/(7117 + 621) = 92%.

#### Precision, Recall and F1 Score

Table

Description automatically generated with medium confidence

#### ROC Curve

  The ROC curve is a plot of True Positive Rate (TPR) versus the False Positive Rate (FPR)

Chart, line chart

Description automatically generated

The dotted line shows a very ineffective classifier, similar to random chance. Our SGD Classifier looks effective as its closer to the top left margin.

#### AUC

The ROC curve can be used to generate the AUC. AUC is the total area under the ROC

  curve.

Graphical user interface, text, application, email

Description automatically generated

### Performance metrics for Test Data

Table

Description automatically generated

## Model 2: Logistic Regression

### Performance metrics for Train Data

#### Accuracy

Text

Description automatically generated

#### Confusion Matrix

Text

Description automatically generated

#### Precision, Recall and F1 Score

Text

Description automatically generated with medium confidence

#### ROC Curve

Graphical user interface, text, application, email

Description automatically generated

#### AUC

Graphical user interface, text, application, Word

Description automatically generated

### Performance metrics for Test Data

Table

Description automatically generated

Logistic Regression and SGD classifier are almost equal in terms of model efficiency, their accuracy approximately 92%.

## Model 3: Random Forest Classifier

### Performance metrics for Train Data

#### Grid Search CV

Graphical user interface, text, application, email

Description automatically generated

#### Accuracy

Text, letter

Description automatically generated

#### Confusion Matrix

Text

Description automatically generated

#### Precision, Recall and F1 Score

Text

Description automatically generated with medium confidence

#### ROC Curve

Graphical user interface, text, application

Description automatically generated

#### AUC

Text

Description automatically generated with medium confidence

### Performance metrics for Test Data

Table

Description automatically generated

We can clearly see that Random Forest is the winner out of the 3 models with an accuracy of 97% , F1 score of 0.97 and AUC of 0.97.

### Feature Importance

Graphical user interface

Description automatically generated

We can see the feature importance for the random forest model. The top three features that contributed toward classifying an URL as "Phishing" or "Legitimate" are SSLFinal\_State, URL\_of\_Anchor and web\_traffic.

# Conclusion

In this Project, we analyzed the cyber security dataset classifying an URL as "Phishing" or "Legitimate". We explored various features in the dataset by drawing graphs. We buit classification models like SGD Classifier, Logistic Regression and Random Forest. We did performance metrics for each of these models. We can conclude that the Random Forest model is the winner among the three models with an accuracy of 97%. The F1 score is 97% and the AUC is 0.97. This indicates that we have built an effective model which can classify the URLs correctly with very minimal misclassification.