ANTI-PHISHING PREDICTION USING MACHINE LEARNING

-MITESH RANGAN TELI
(TCRIG03R35)

ABSTRACT:

Anti-phishing refers to efforts to block phishing attacks. Phishing is a kind of cybercrime where attackers pose as known or trusted entities and contact individuals through email, text or telephone and ask them to share sensitive information. Nowadays, many anti- phishing systems are being developed to identify phishing contents in online communication systems. Despite the availability of anti-phishing systems, phishing continues unabated due to inadequate detection of a zero-day attack, superfluous computational overhead and high false rates.

Although Machine Learning approaches have achieved promising accuracy rate, the choice and the performance of the feature vector limit their effective detection. In this work, an enhanced machine learning-based predictive model is used to improve the efficiency of anti-phishing schemes. The predictive model consists of Feature Selection Module which is used for the construction of an effective feature vector. These features are extracted from the URL, webpage properties and webpage behavior using the incremental component-based system to present the resultant feature vector to the predictive model. The proposed system uses Logistic Regression and Random Forest Classifier which have been trained on a 50-dimensional feature set. The experiments were based on datasets consisting of 10000 phishing instances.

OBJECTIVE:

To propose and implement Machine Learning models that can classify the URLs as legitimate or illegitimate URLs based on the various features like Url Length, Path Level, Hostname Length, Http in Host Name etc.

DATASET:

This dataset contains 48 features extracted from 5000 phishing webpages and 5000 legitimate webpages, which were downloaded from January to May 2015 and from May to June 2017. An improved feature extraction technique is employed by leveraging the browser automation framework (i.e., Selenium Web Driver), which is more precise and robust compared to the parsing approach based on regular expression.

ALGORITHMS USED:

After splitting the dataset into training(80%) and testing (20%) dataset. I've used Scikit libraries for splitting the dataset, for scaling the dataset and for implementing LogisticRegression and Random Forest Classification.

1. Random Forest Classifier

A Random Forest Algorithm is a supervised machine learning algorithm that is extremely popular and is used for Classification and Regression problems in Machine Learning. The greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability. Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.

Before using Random Forest Classifier, I've scaled the data using Standard Scaler and then applied Random Forest Classifier on it. Similarly, I've calculated the confusion matrix, MAE, MSE, recall and precision scores for training and testing dataset.

2. Logistic Regression

Logistic regression is a statistical method that is used for building machine learning models where the dependent variable is dichotomous: i.e. binary. Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables.

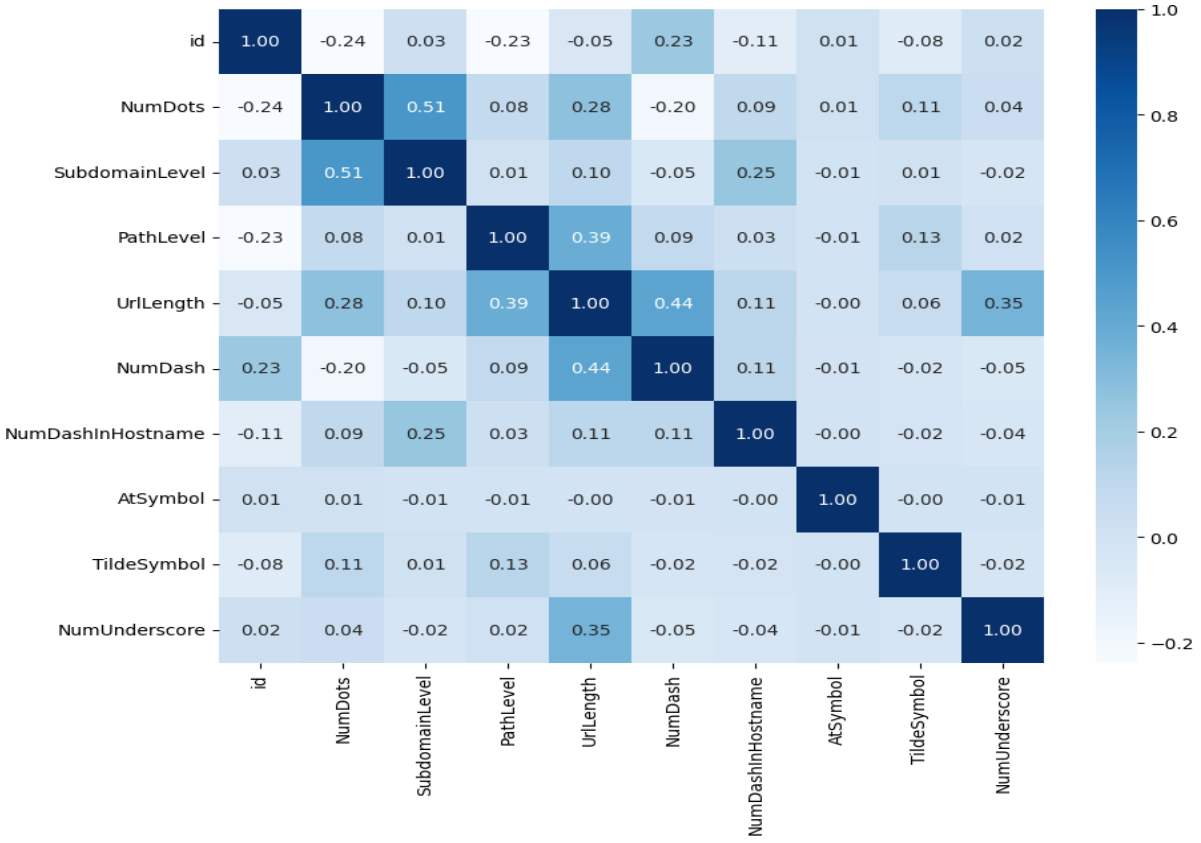
I've applied Logistic Regression on the dataset and predicted the values for corresponding testing dataset. Calculated the confusion matrix, MAE, MSE, recall and precision scores for training and testing dataset.

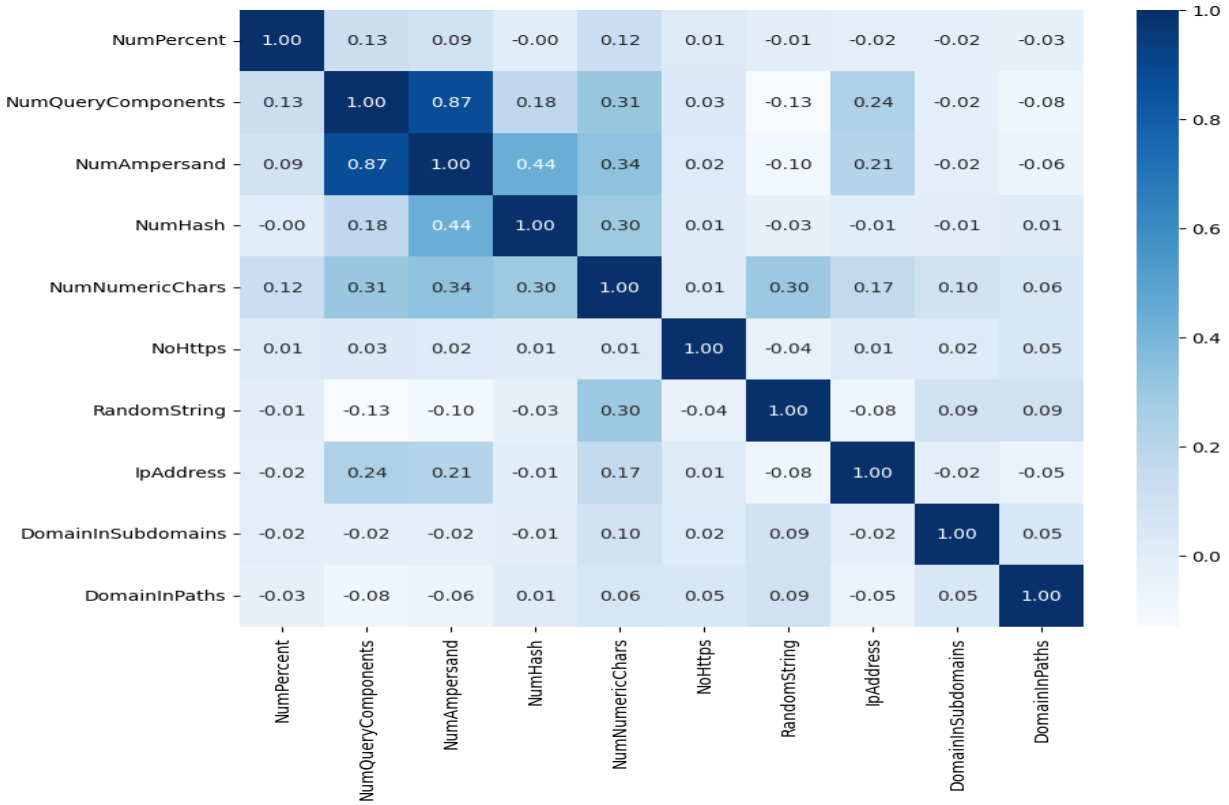
OUTPUT/ACCURACY:

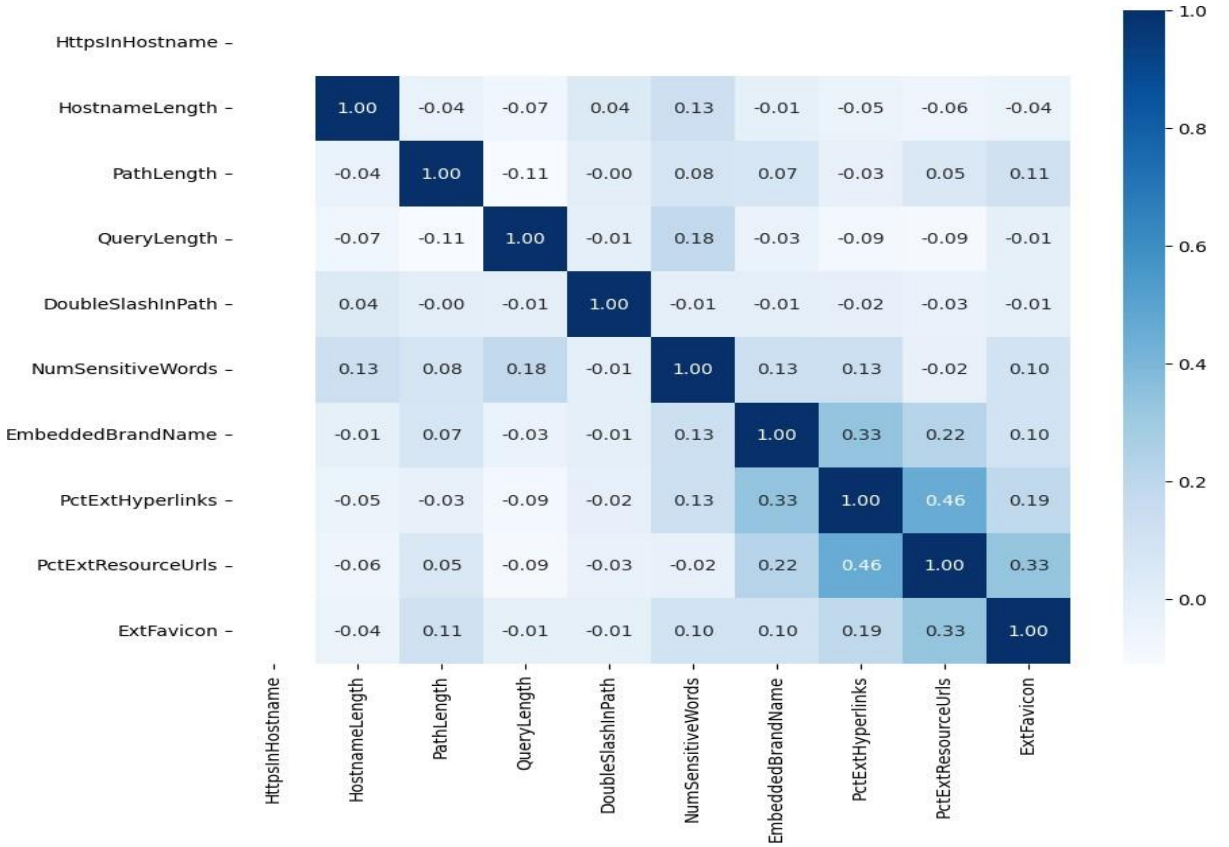
After applying the classification algorithms following are the results :

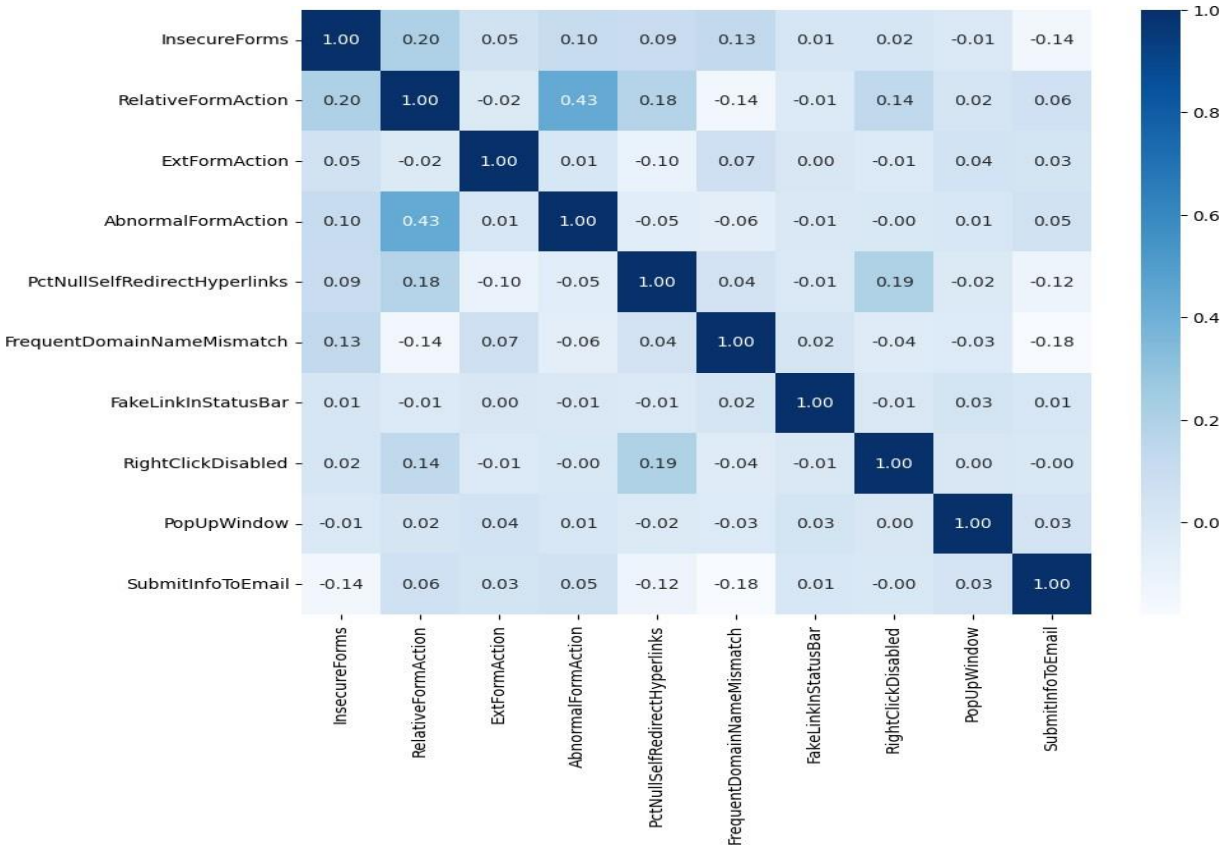
Algorithm used	Accuracy	Precision	Recall
Random Forest classifier	98.35 %	0.985	0.982
Logistic Regression	93.85 %	0.931	0.947

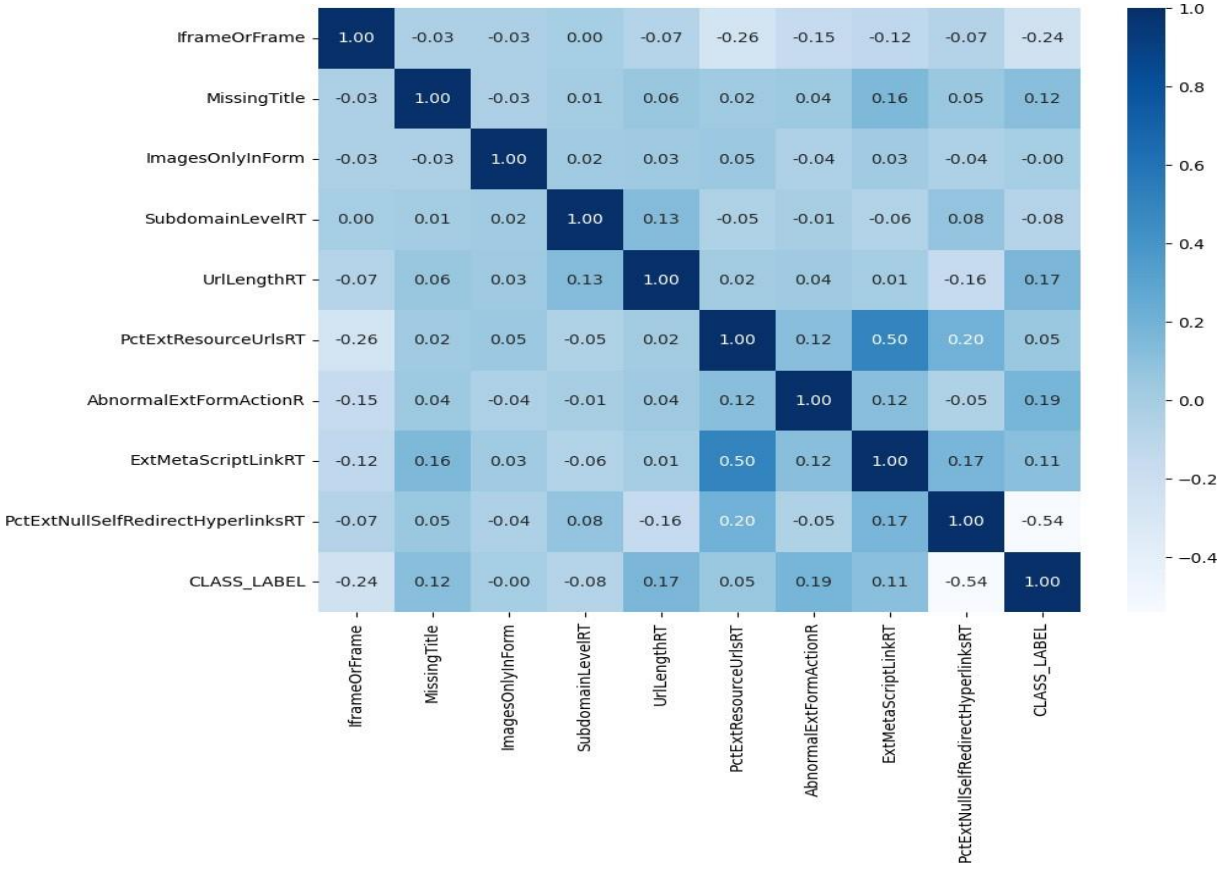
GRAPHS:



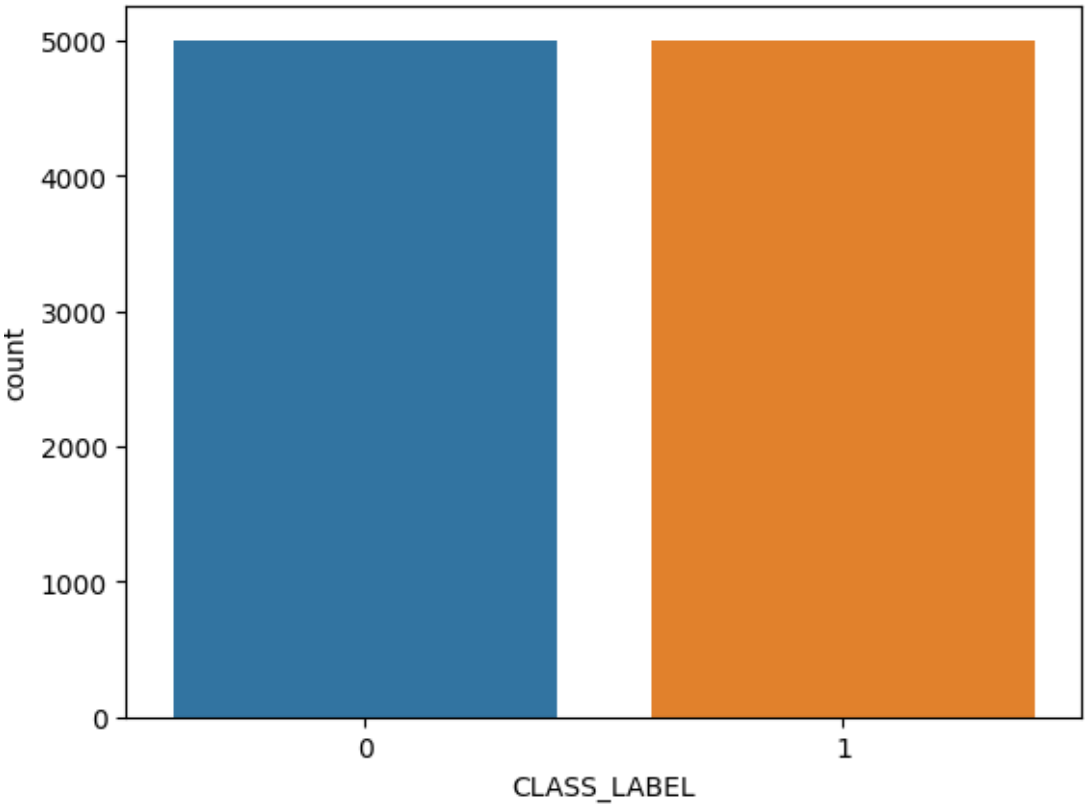

NumPercent 1.00 0.13 0.09 -0.00 0.12 0.01 -0.01 -0.02 -0.02 -0.03
NumQueryComponents 0.13 1.00 0.87 0.18 0.31 0.03 -0.13 0.24 -0.02 -0.08
NumAmpersand 0.09 0.87 1.00 0.44 0.34 0.02 -0.10 0.21 -0.02 -0.06
NumHash -0.00 0.18 0.44 1.00 0.30 0.01 -0.03 -0.01 -0.01 0.01
NumNumericChars 0.12 0.31 0.34 0.30 1.00 0.01 0.30 0.17 0.10 0.06
NoHttps 0.01 0.03 0.02 0.01 0.01 1.00 -0.04 0.01 0.02 0.05
RandomString -0.01 -0.13 -0.10 -0.03 0.30 -0.04 1.00 -0.08 0.09 0.09
IpAddress -0.02 0.24 0.21 -0.01 0.17 0.01 -0.08 1.00 -0.02 -0.05
DomainInSubdomains -0.02 -0.02 -0.02 -0.01 0.10 0.02 0.09 -0.02 1.00 0.05
DomainInPaths -0.03 -0.08 -0.06 0.01 0.06 0.05 0.09 -0.05 0.05 1.00
NumPercent
NumQueryComponents
NumAmpersand
NumHash
NumNumericChars
NoHttps
RandomString
IpAddress
DomainInSubdomains
DomainInPaths
1.0
0.8
0.6
0.4
0.2
0.0


HttpsInHostname
HostnameLength 1.00 -0.04 -0.07 0.04 0.13 -0.01 -0.05 -0.06 -0.04
PathLength -0.04 1.00 -0.11 -0.00 0.08 0.07 -0.03 0.05 0.11
QueryLength -0.07 -0.11 1.00 -0.01 0.18 -0.03 -0.09 -0.09 -0.01
DoubleSlashInPath 0.04 -0.00 -0.01 1.00 -0.01 -0.01 -0.02 -0.03 -0.01
NumSensitiveWords 0.13 0.08 0.18 -0.01 1.00 0.13 0.13 -0.02 0.10
EmbeddedBrandName -0.01 0.07 -0.03 -0.01 0.13 1.00 0.33 0.22 0.10
PctExtHyperlinks -0.05 -0.03 -0.09 -0.02 0.13 0.33 1.00 0.46 0.19
PctExtResourceUrls -0.06 0.05 -0.09 -0.03 -0.02 0.22 0.46 1.00 0.33
ExtFavicon -0.04 0.11 -0.01 -0.01 0.10 0.10 0.19 0.33 1.00
HttpsInHostname
HostnameLength
PathLength
QueryLength
DoubleSlashInPath
NumSensitiveWords
EmbeddedBrandName
PctExtHyperlinks
PctExtResourceUrls
ExtFavicon
1.0
0.8
0.6
0.4
0.2
0.0





Heatmap for 10 columns each



Count plot for class label

CONCLUSION:

After applying the two classifiers it was found that the Random Forest classifier was more accurate in terms of predicting the output and was more efficient.