M.KUMARASAMY
COLLEGE OF ENGINEERING
NAAC Accredited Autonomous Institution
Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 & ISO 14001:2015 Certified Institution
Thalavapalayam, Karur – 639 113.

A Project Report

on

WEB PHISHING DETECTION

Submitted in partial fulfillment of requirements for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of

Mrs. C. SELVARATHI M.E.,
Assistant Professor/CSE

Submitted By

TEAM ID : PNT2022TMID15373

927619BCS4018 - CHARUMATHI C
927619BCS4019 - DEEPAK B
927619BCS4041 - HEMAPPRABHA K
927619BCS4052 - KARTHIKEYAN R

NALAIYA THIRAN - EXPERIENTIAL PROJECT BASED LEARNING INITIATIVE

18CSE040L - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTERPRENURSHIP

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
M.KUMARASAMY COLLEGE OF ENGINEERING
(Autonomous)

KARUR - 639 113

NOVEMBER, 2022

ABSTRACT

A phishing attack is one of the most significant problems faced by online users because of its enormous effect on the online activities performed. In recent years, phishing attacks continue to escalate in frequency, severity and impact. Several solutions, using various methodologies, have been proposed in the literature to counter the web-phishing threats. Not withstanding, the existing technology cannot detect the new phishing attacks accurately due to the insufficient integration of features of the text, image and frame in the evaluation process. The use of related features of images, frames and text of legitimate and non-legitimate websites and associated artificial intelligence algorithms to develop an integrated method to address these together. The proposed solution achieves 92.2% accuracies. To our best knowledge, this is the first work that considers the best-integrated text, image and frame feature based solution for phishing detection scheme.

TABLE OF CONTENT

LIST OF TABLE

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Phishing costs Internet users billions of dollars per year. It refers to luring techniques used by identity thieves to fish for personal information in a pond of unsuspecting Internet users. Phishers use spoofed e-mail, phishing software to steal personal information and financial account details such as usernames and passwords. This paper deals with methods for detecting phishing Web sites by analyzing various features of benign and phishing URLs by Machine learning techniques. We discuss the methods used for detection of phishing Web sites based on lexical features, host properties and page importance properties. We consider various machine learning algorithms for evaluation of the features in order to get a better understanding of the structure of URLs that spread phishing. The fine-tuned parameters are useful in selecting the apt machine learning algorithm for separating the phishing sites from benign sites.

1.2 PURPOSE

The main purpose of the project is to detect the fake or phishing websites who are trying to get access to the sensitive data or by creating the fake websites and trying to get access of the user personal credentials. We are using machine learning algorithms to safeguard the sensitive data and to detect the phishing websites who are trying to gain access on sensitive data.This system can be used by many E-commerce or other websites in order to have good customer relationship. User can make online payment securely. Logistic Regression algorithm used in this system provides better performance as compared to other classification algorithms.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISITING PROBLEM

Phishing Detection techniques do suffer low detection accuracy and high false alarm especially when novel phishing approaches are introduced. Besides, the most common technique used, blacklist-based method is inefficient in responding to emanating phishing attacks since registering new domain has become easier, no comprehensive blacklist can ensure a perfect up-to-date database. Furthermore, page content inspection has been used by some strategies to overcome the false negative problems and complement the vulnerabilities of the stale lists. Moreover, page content inspection algorithms each have different approach to phishing website detection with varying degrees of accuracy. Therefore, ensemble can be seen to be a better solution as it can combine the similarity in accuracy and different error-detection rate properties in selected algorithms.

2.2 REFERENCES

1. Adebowale, M. A., Lwin, K. T., Sánchez, E. and Hossain, M. A. (2019) 'Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text', Expert Systems with Applications, 115, pp. 300-313.
2. Aggarwal, A., Rajadesingan, A. and Kumaraguru, P. 'PhishAri: Automatic real-time phishing detection on twitter', eCrime Researchers Summit (eCrime), Las Croabas, Puerto Rico, 23-24 Oct. 2012: IEEE, 1- 12.
3. Lee, J.-L. and Park, D.-h. (2016) 'Phishing Detection Using Web Site Heuristics', International Information Institute (Tokyo), 19(2), pp. 523-530.
4. Yang, P., Zhao, G. and Zeng, P. (2019) 'Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning', IEEE Access, 7, pp. 15196 15209.

5. Mohammad R., Thabtah F. McCluskey L., (2015) Phishing websites dataset. Available: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites Accessed January 2016.
6. Gunter Ollmann, "The Phishing Guide Understanding & Preventing Phishing Attacks", IBMInternet Security Systems, 2007.
7. http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/
8. https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html
9. https://resources.infosecinstitute.com/category/enterprise/phishing/thephishing-landscape/phishing-data-attack-statistics/#gref
10. http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/

2.3 PROBLEM STATEMENT DEFENITION

Phishing is one of the techniques which are used by the intruders to get access to the user credentials or to gain access to the sensitive data. This type of accessing the is done by creating the replica of the websites which looks same as the original websites which we use on our daily basis but when a user click on the link he will see the website and think its original and try to provide his credentials . To overcome this problem we are using some of the machine learning algorithms in which it will help us to identify the phishing websites based on the features present in the algorithm. By using these algorithm we can be able to keep the user personal credentials or the sensitive data safe from the intruders.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

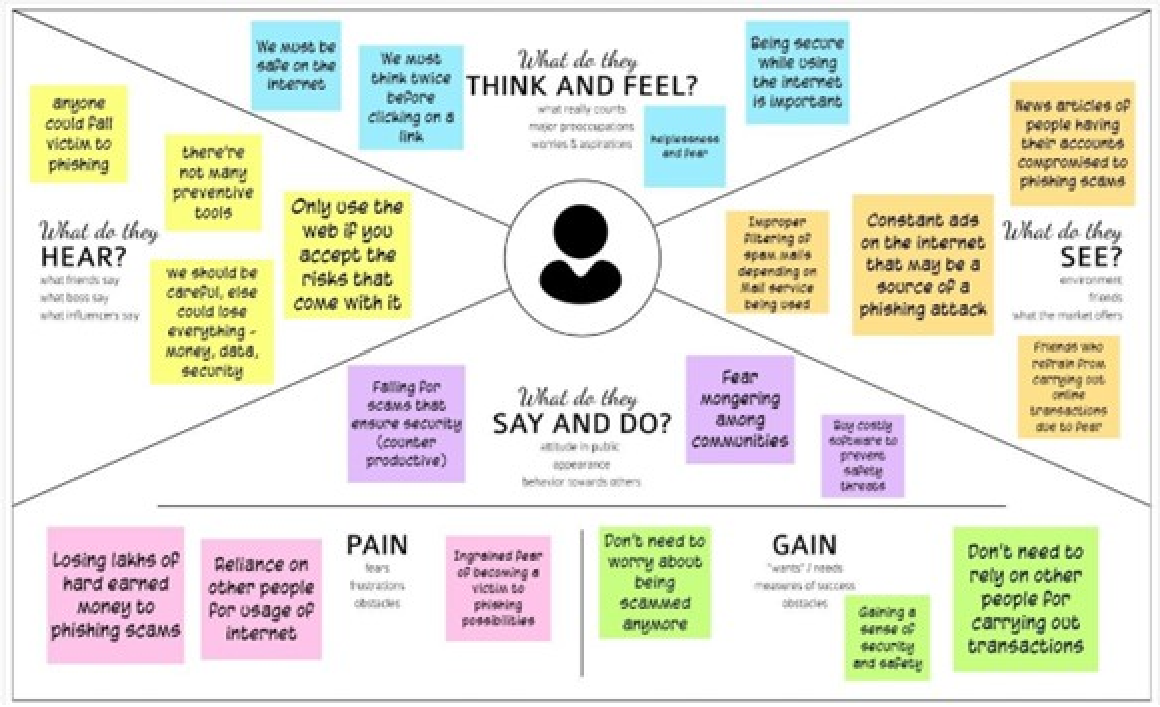

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
We must be safe on the internet
We must think twice before clicking on a link
Being secure while using the internet is important
helplessness and fear
What do they HEAR?
what friends say
what boss say
what influencers say
anyone could fall victim to phishing
there're not many preventive tools
we should be careful, else could lose everything - money, data, security
Only use the web if you accept the risks that come with it
What do they SEE?
environment
friends
what the market offers
News articles of people having their accounts compromised to phishing scams
Improper filtering of spam mails depending on Mail service being used
Constant ads on the internet that may be a source of a phishing attack
Friends who refrain from carrying out online transactions due to fear
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Falling for scams that ensure security (counter productive)
Fear mongering among communities
Buy costly software to prevent safety threats
PAIN
fears
frustrations
obstacles
Losing lakhs of hard earned money to phishing scams
Reliance on other people for usage of internet
Ingrained fear of becoming a victim to phishing possibilities
GAIN
"wants" / needs
measures of success
obstacles
Don't need to worry about being scammed anymore
Gaining a sense of security and safety
Don't need to rely on other people for carrying out transactions

Figure 3.1 Empathy Map

An **empathy map** is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to

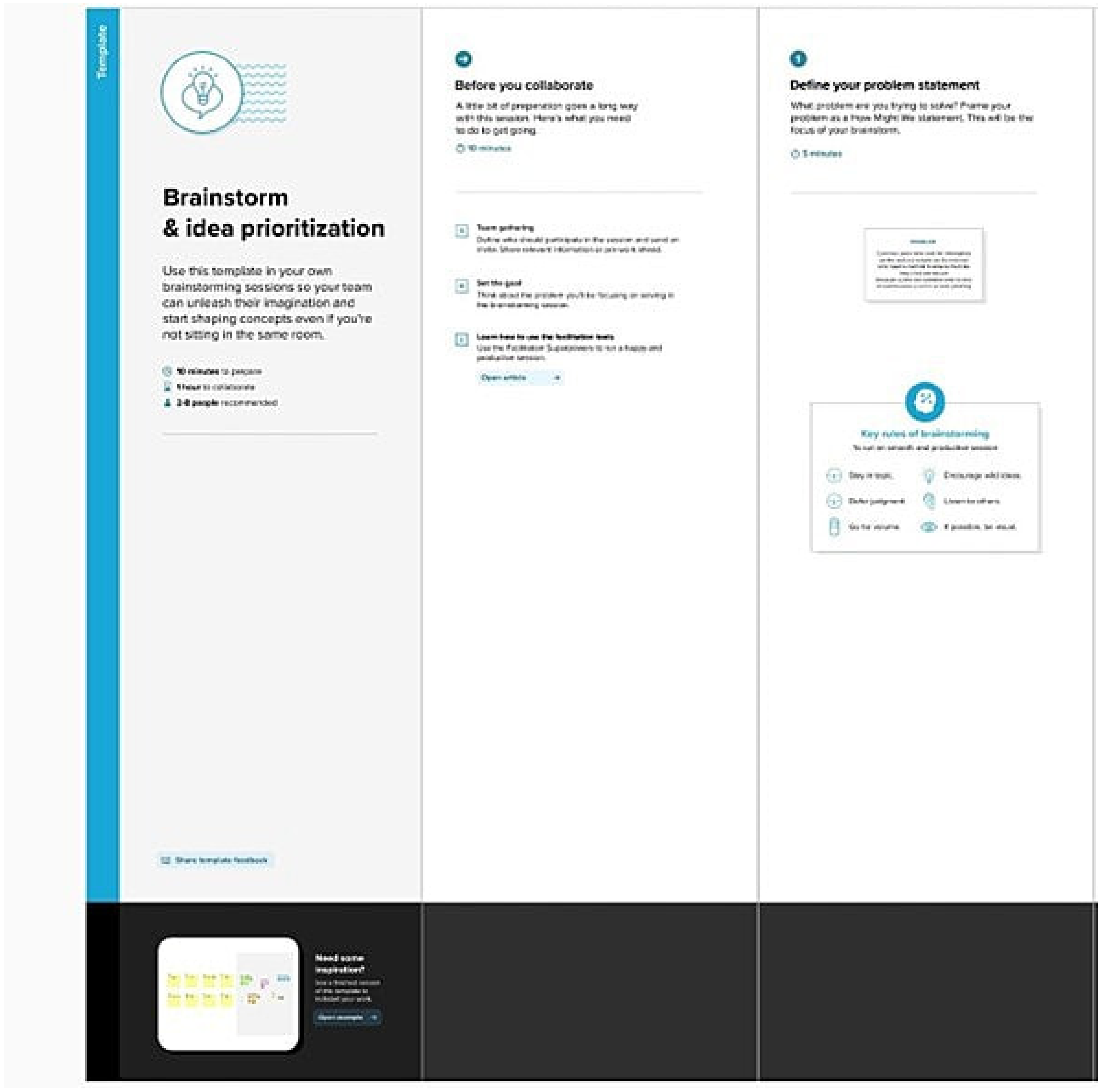
1) Create a shared understanding of user needs, and

2) Aid in decision making.

3.2 IDEATION AND BRAINSTORMING

Step-1: Team Gathering, Collaboration and Select the Problem Statement

A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

[illegible]

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

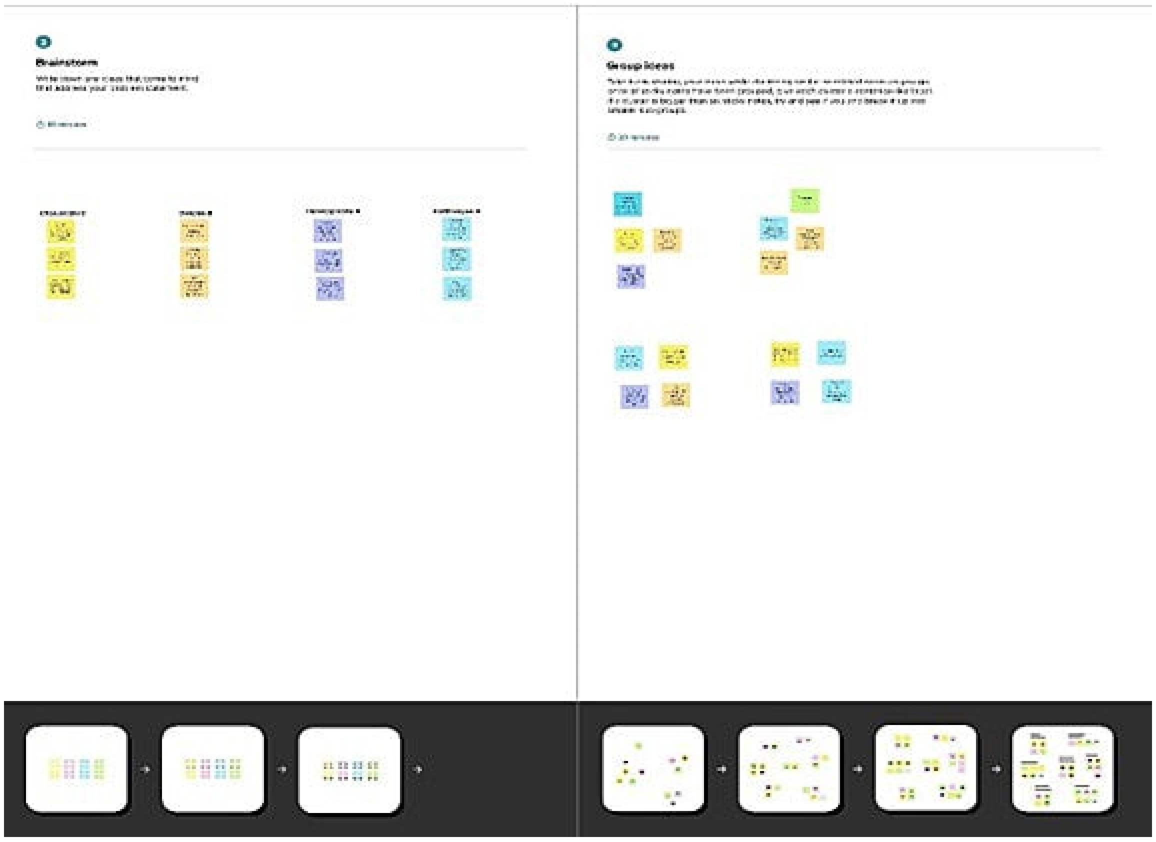
Share template feedback

Need some inspiration?
See a finished version of this template to kickstart your work.
Open example

Figure 3.2 Ideation and Brainstorming

Step-2: Brainstorm, Idea Listing and Grouping

The idea listing and grouping is used to organize and analyse large numbers of ideas by categorising them. By organising and reorganising ideas, students gain a better appreciation of, and dialogue about, their ideas. As students create idea clusters, new contexts and connections among themes emerge.



+

Figure 3.3 Ideation and Brainstorming

Step-3 : Idea Prioritization

Idea prioritization is just a part of the idea management process. Having a structured idea management process and a systematic way of gathering, evaluating and prioritizing new ideas takes time. To make it work, the entire idea management process should be integrated to the everyday ways of working.

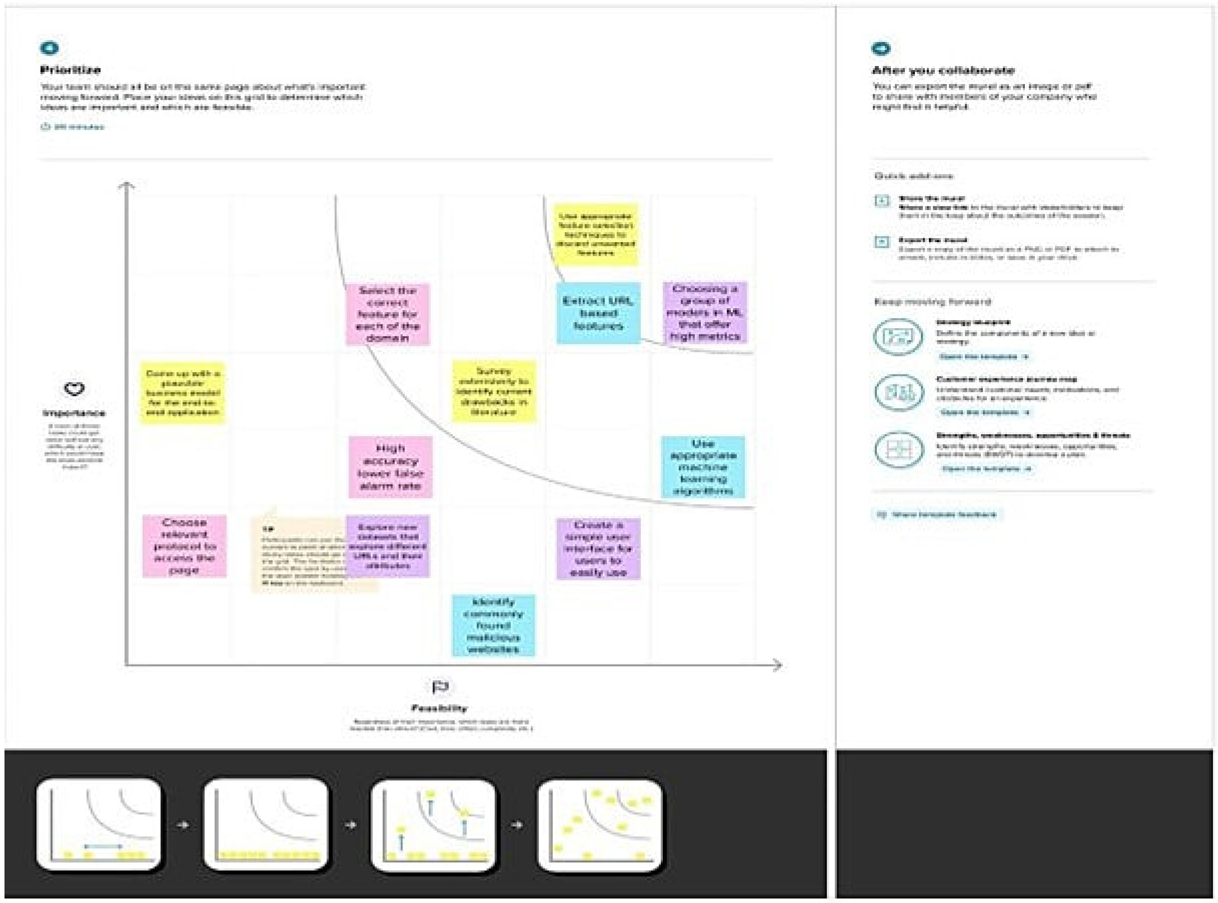

Prioritize
After you collaborate
Quick add-ons
Keep moving forward
Importance
Use appropriate feature selection techniques to discard unwanted features
Select the correct feature for each of the domain
Extract URL based features
Choosing a group of models in ML that offer high metrics
Come up with a plausible business model for the end-to-end application
Survey extensively to identify current drawbacks in literature
High accuracy lower false alarm rate
Use appropriate machine learning algorithms
Choose relevant protocol to access the page
Explore new datasets that explore different URLs and their attributes
Create a simple user interface for users to easily use
Identify commonly found malicious websites
Feasibility

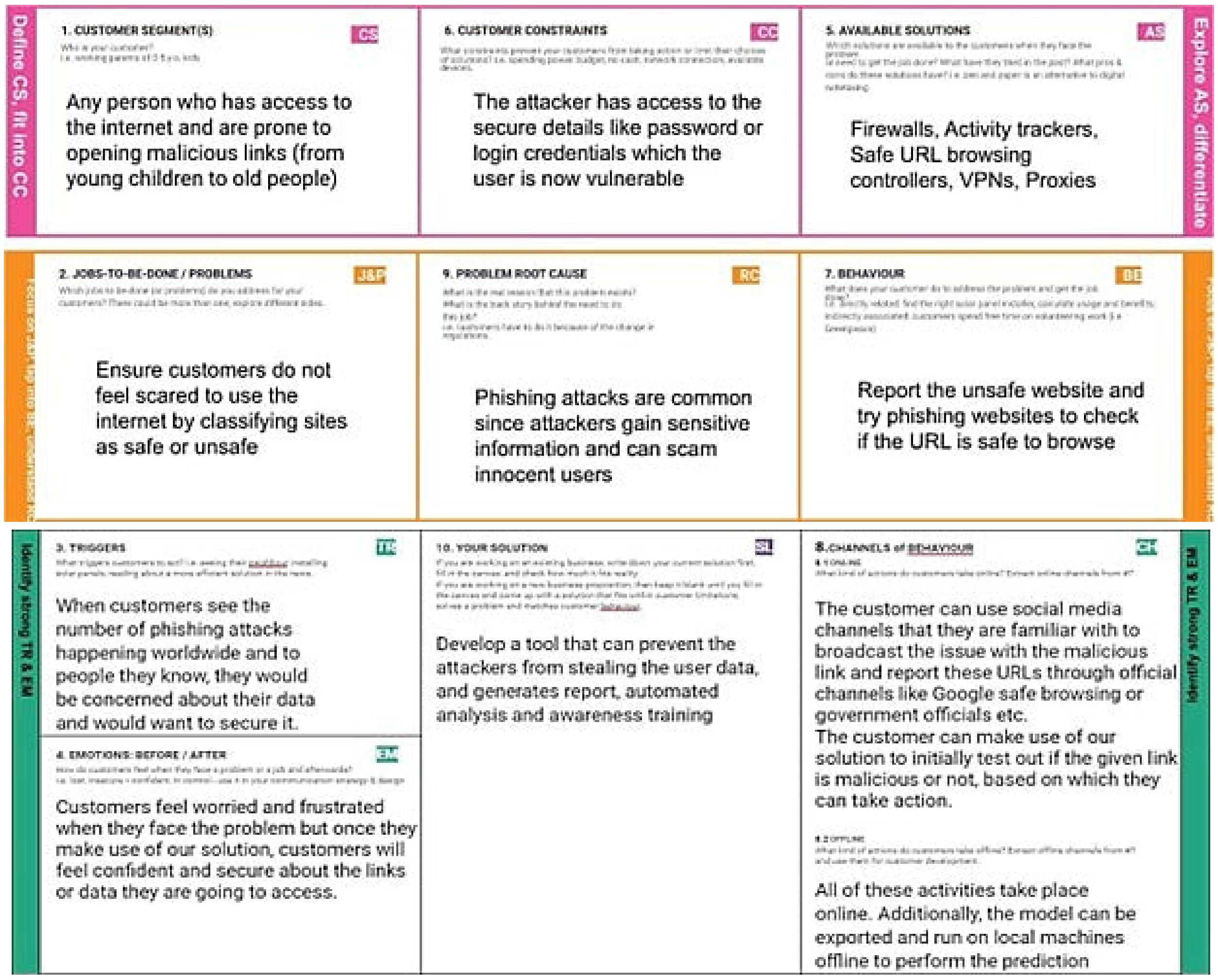
Figure 3.4 Ideation and Brainstorming

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	To reduce the people falling for web phishing scams by creating a sophisticated tool that classifies a website as malicious or safe to use
2.	**Idea/Solution description**	Identify web phishing, classify whether it is an attack and preventmalicious intrusive websites
3.	**Novelty / Uniqueness**	1. Uses an Ensemble model 2. Explores weighted features for Neural Networkapproaches 3. Extensive feature extraction strategy from theURL 4. Simple, Easy-to-Understand UI
4.	**Social Impact / Customer Satisfaction**	1. Users need not fear of losing lakhs of hard earned money to phishing scams& Users neednot feel scaredto use the internet 2. Primarily targets the benefitof senior citizens and technologically challenged sections of the society 3. Customers don't need to rely on offline transactions because ofthefear of initiating transactions online
5.	**Business Model (Revenue Model)**	1. B2B (Machine Learning model/API can be sold to various companies for their employees) and B2C Model (End product sold to individuals such as children's devices and seniorcitizensprone to attacks) 2. Site can charge a one timefee for a device/user basedon demographic surveys (Rs. 50 peryear) 3. Companies can be charged a discounted fee due to bulk purchase of the Application Programming Interface (API) 4. Premium users will haveaccess to detailsof the URL and reasonings for why a site has been classified 'unsafe'

Table 3.1 Proposed Solution

3.4 PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Any person who has access to the internet and are prone to opening malicious links (from young children to old people)

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

The attacker has access to the secure details like password or login credentials which the user is now vulnerable

5. AVAILABLE SOLUTIONS AS

[illegible]

Firewalls, Activity trackers, Safe URL browsing controllers, VPNs, Proxies

Explore AS, differentiate

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides.

Ensure customers do not feel scared to use the internet by classifying sites as safe or unsafe

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Phishing attacks are common since attackers gain sensitive information and can scam innocent users

7. BEHAVIOUR BE

[illegible]

Report the unsafe website and try phishing websites to check if the URL is safe to browse

Identify strong TR & EM

3. TRIGGERS TR

[illegible]

When customers see the number of phishing attacks happening worldwide and to people they know, they would be concerned about their data and would want to secure it.

4. EMOTIONS: BEFORE / AFTER EM

[illegible]

Customers feel worried and frustrated when they face the problem but once they make use of our solution, customers will feel confident and secure about the links or data they are going to access.

10. YOUR SOLUTION SL

[illegible]

Develop a tool that can prevent the attackers from stealing the user data, and generates report, automated analysis and awareness training

8.CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

The customer can use social media channels that they are familiar with to broadcast the issue with the malicious link and report these URLs through official channels like Google safe browsing or government officials etc.
The customer can make use of our solution to initially test out if the given link is malicious or not, based on which they can take action.

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

All of these activities take place online. Additionally, the model can be exported and run on local machines offline to perform the prediction

Identify strong TR & EM

Figure 3.5 Problem Solution Fit

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks.

FR NO.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Input	User inputsan URL in required field to check its validation.
FR-2	Website Comparison	Model compares the websites using Blacklist and Whitelist approach
FR-3	Feature extraction	After comparing, if none foundon comparison then itextracts feature using heuristic and visualsimilarity approach.
FR-4	Prediction	Model predicts the URL using Machine Learning algorithms such as Logistic Regression, KNN
FR-5	Classifier	Model sendsall output to classifier and produces final result.
FR-6	Announcement	Model then displays whether website is a legalsite or a phishing site.
FR-7	Events	This model needs the capability of retrieving and displaying accurate result for a website.

Table 4.1 Functional Requirements

4.2 NON-FUNCTIONAL REQUIREMENTS

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Analysis of consumers' product usability in the design process with user experience as the core maycertainly help designers better grasp users' prospective demandsin web phishing detection, behaviour, and experience.
NFR-2	Security	It guarantees that any data included within the system or its components will be safe from malwarethreats or unauthorised access. If you wish to prevent unauthorised access to the admin panel, describe the login flow and different user roles as system behaviour or user actions.
NFR-3	Reliability	It specifies the likelihood that the system or its component will operate without failure for a specific amount of time under prescribedconditions.
NFR-4	Performance	It is concerned with a measurement of the system's reaction time undervarious load circumstances.
NFR-5	Availability	It represents the likelihood that a userwill be ableto accessthe system at a certain momentin time.While it can be represented as an expected proportion of successful requests, it can also be defined as a percentage of time the system is operational within a certain time period.

Table 4.2 Non Functional Requirements

CHAPTER 5

PROJECT DESIGN

5.1 DATAFLOW DIAGRAM

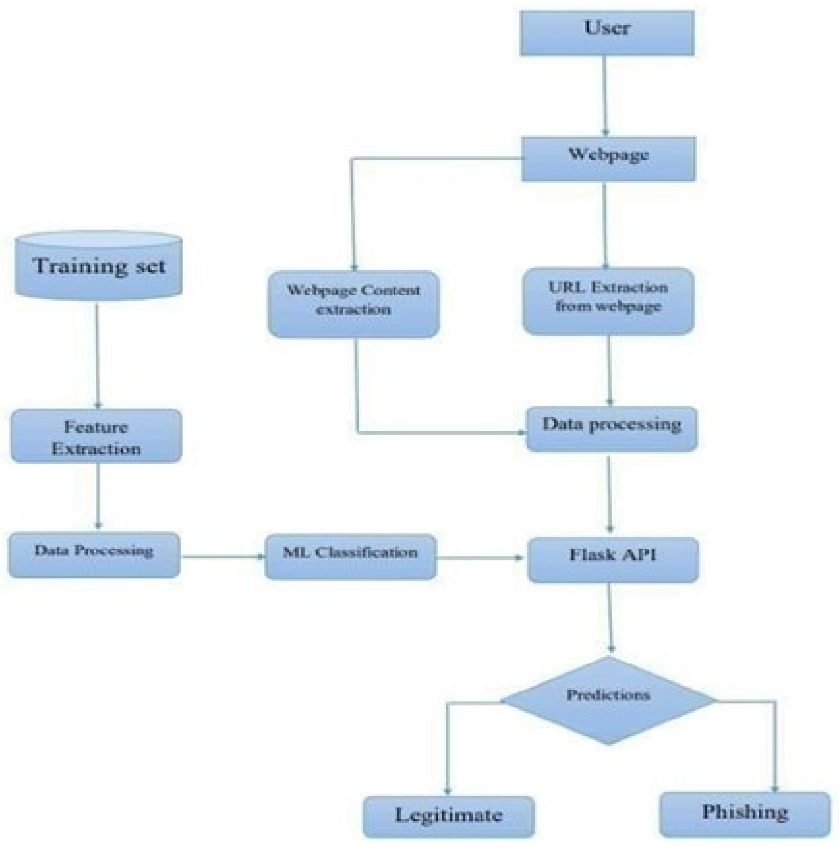


Figure 5.1 Dataflow Daigram of Web Phishing Detection

1. Website data is collected from Kaggle
2. Exploratory data analysis done on the input dataset.
3. Then removal of null values, duplicates and outliers.
4. Then the dependent and independent variable is defined.
5. Train test split is done.
6. Regression model is built.
7. Then the model is fitted with front end which is developed using HTML, CSS with the help of Python Flask Web Framework.
8. Finally, the output will be predicted for the user input.

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

The solution architecture which can explains to collect the dataset and move to pre-processing, then need to train and test datas with the help of machine learning algorithms, to predict the output. Technical architecture includes the major components of the system, their relationships, and the contracts that define the interactions between the components. The goal of technical architects is to achieve all the business needs with an application that is optimized for both performance and security.

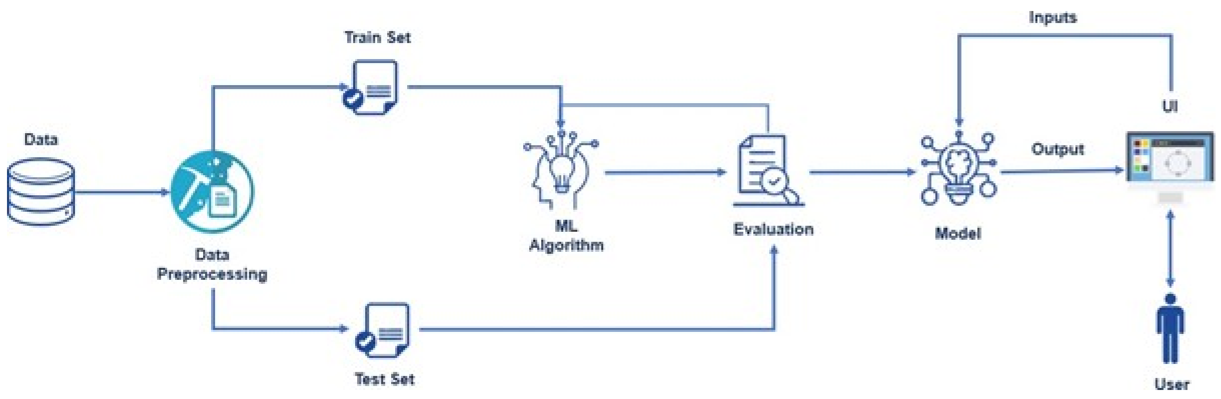


Figure 5.2 Solution and technical architecture

5.3 USER STORY

User Type	Functional Requirement (Epic)	User Story number	User Story/Task	Acceptance Criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application before entering my email,password, and cofirming my password	I can access my account/dashboard	High	Sprint-1
Customer (Mobile user)	Registration	USN-2	As a user,I will receive confirmation email once I have registerd for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer (Mobile user)	Registration	USN-3	As a user ,Ican register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
Customer (Mobile user)	Registration	USN-4	As a user ,I can register for the application through Gmail	I can register and access it	Medium	Sprint-1
Customer (Mobile user)	Login	USN-5	As a user,I can log into the application by entering email & password	I can login using my credentials	High	Sprint-1
Customer (Web User)	User Input	USN-1	As a user I can input the particular URL in the required field and waiting for validation	I can go access the website without any problem	High	Sprint-1
Customer Care Executive	Feature Extraction	USN-1	After I compare in case if none found on comparison then we can extract feature	As a User I can have comparison between websites for security	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story number	User Story/Task	Acceptance Criteria	Priority	Release
Administrator	Prediction	USN-1	Here the Model will predict the URL websites using Machine Learning algorithm such as Logistic Regression	In this I can have correct prediction on the particular algorithms	High	Sprint-1
	Classifier	USN-2	Here I will send all the model output to classifier in order to produce final result	In this I will find the correct classifier for producing the result	Medium	Sprint-2

Table 5.1 User Story

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requiremet (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect the appropriate dataset for detecting phishing website.	10	High	Charumathi C
Sprint-1	Data Preprocessing	USN-2	Used to transform the data into useful format.	7	Medium	Hemapprabha K, Deepak B
Sprint-2	Model Building	USN-3	Detect the phishing Website.	10	High	Karthikeyan R
Sprint-2	Model Building	USN-4	Splitting the Model into Training and Testing from the overall dataset	7	Medium	Karthikeyan R
Sprint-3	Training and Testing	USN-5	Train the Model using Logistic Regression algorithm and Testing the Performance of the model.	10	High	Charumathi C
Sprint-3	Application Building	USN-6	Build the HTML and Python code	7	Medium	Hemapprabha K
Sprint-4	Implementation of the Application	USN-7	Run Flask App	10	High	Deepak B
Sprint-4	Implementation of the Application	USN-8	Deploy the Model on IBM Cloud.	7	Medium	Charumathi C

Table 6.1 Sprint Planning and Estimation

6.1.1 Project Tracker, Velocity& Burndown Chart

Sprint	Total StoryPoints	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	7	19 Nov 2022

Table 6.1.1 Project Tracker,Velocity & Burndown Chart

VELOCITY

Imagine we have 6 – days sprint duration, and the velocity of the team is 10 (points persprint). Let’s calculate the team’s average (AV) per iteration unit (Story pointsper day).

AV = Sprint Duration/Velocity

= 6/10

= 0.6

6.3 SPRINT DELIVERY SCHEDULE

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gatheringinformation by referring thetechnical papers, research publications, journals etc.	8 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, prepare list of problem Statements that are to be solvedby this project.	8 SEPTEMBER 2022
Ideation	List the ideas by organizing abrainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	16 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solutiondocument, which includes novelty, feasibility of idea, revenue model, social impact,scalability of solution, etc.	23 SEPTEMBER 2022
Solution Architecture	Prepare Solution Architecture document.	27 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fitdocument.	1 OCTOBER 2022

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry toexit).	6 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	15 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit forreview.	15 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	16 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity listof the project.	21 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 &4	Develop & submit the developed code by testingit.	19 NOVEMBER 2022

Table 6.2 Sprint Delivery Schedule

CHAPTER 7

CODING AND SOLUTION

7.1 FEATURE 1

Logistic Regresssion is used to train and test the model for detecting the phishing website with the help of collected and preprocessed dataset collections. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Moreover, NumPy forms the foundation of the Machine Learning stack. Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. Sea born is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.EDA is applied to investigate the data and summarize the key insights. It will give you the basic understanding of your data, it is distribution, null values and much more. You can either explore data using graphs or through some python functions. There will be two types of analysis. Descriptive statistics are brief informational coefficients that summarize a given data set, which can be either a representation of the entire population or a sample of a population. Descriptive statistics are broken down into measures of central tendency and measures of variability.

Measures of central tendency include the mean, median, and mode, while measures of variability include standard deviation, variance, minimum and maximum variables, kurtosis, and Skewness. Label Encoding refers to converting the labels into a numeric form to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream is converted back into an object hierarchy. 19 XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework.

7.2 FEATURE 2

The framework is the basis upon which software programs are built. It serves as a foundation for software developers, allowing them to create a variety of applications for certain platforms. It is a set of functions and predefined classes used to connect with the system software and handle inputs and outputs. It simplifies the life of a developer while giving them the ability to use certain extensions and makes the online applications scalable and maintainable. Flask is a web application framework written in Python. A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, among other examples. Flask is a web application framework written in Python.

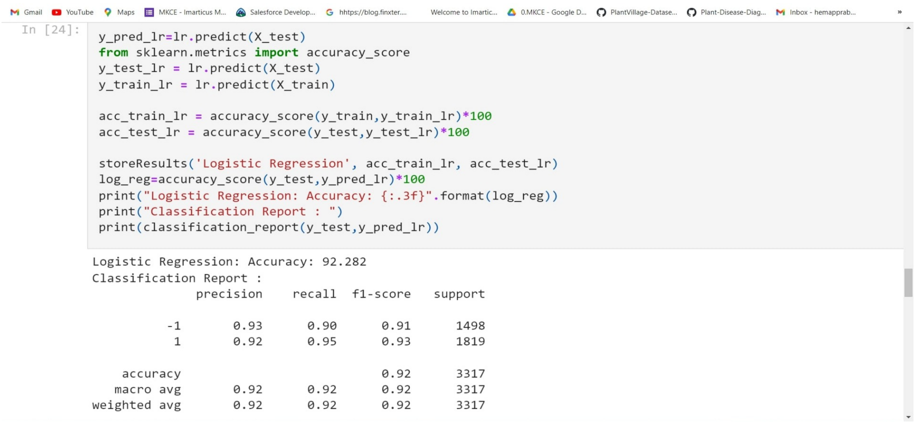
Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. Both are Pocco projects. The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications. Flask is often referred to as a micro-framework. It is designed to keep the core of the application simple and scalable. Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application. Unlike the Django framework, Flask is very Pythonic. It’s easy to get started with Flask, because it doesn’t have a huge learning curve.HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. The model deployed using Flask is used to detect the Web Phishing. Hypertext markup language (HTML) is the basic language used to create documents for the Web and, along 20 with HTTP (hypertext transfer protocol) and URLs (universal resource locators), is one of the three main protocols of the Web. Hypertext is text that contains hyperlinks. A hyperlink is an automated cross-reference to another location on the same document or to another document which, when selected by a user, causes the computer to display the linked location or document within a concise period.

A markup language is a set of tags that can be embedded in digital text to provide additional information about it, including its content, structure and appearance. This information facilitates automated operations on the text, including formatting it for display, searching it and even modifying it. Some type of markup language is employed by every word processing program and by nearly every other program that displays text, although such languages and their tags are typically hidden from the user.HTML consists of a set of predefined tags that can be embedded in text by web site designers in order to indicate the details of how web pages are rendered (i.e., converted into a final, easily usable, form) by web browsers. These details include paragraphing, margins, fonts (including style and size), columns, colors (background and text), links, the location of images, text flow around images, tables, and user input form elements (such as spaces for adding text and submit buttons).

CHAPTER 8

TESTING

8.1 TEST CASES



```
In [24]:  y_pred_lr=lr.predict(X_test)
          from sklearn.metrics import accuracy_score
          y_test_lr = lr.predict(X_test)
          y_train_lr = lr.predict(X_train)

          acc_train_lr = accuracy_score(y_train,y_train_lr)*100
          acc_test_lr = accuracy_score(y_test,y_test_lr)*100

          storeResults('Logistic Regression', acc_train_lr, acc_test_lr)
          log_reg=accuracy_score(y_test,y_pred_lr)*100
          print("Logistic Regression: Accuracy: {:.3f}".format(log_reg))
          print("Classification Report : ")
          print(classification_report(y_test,y_pred_lr))
```

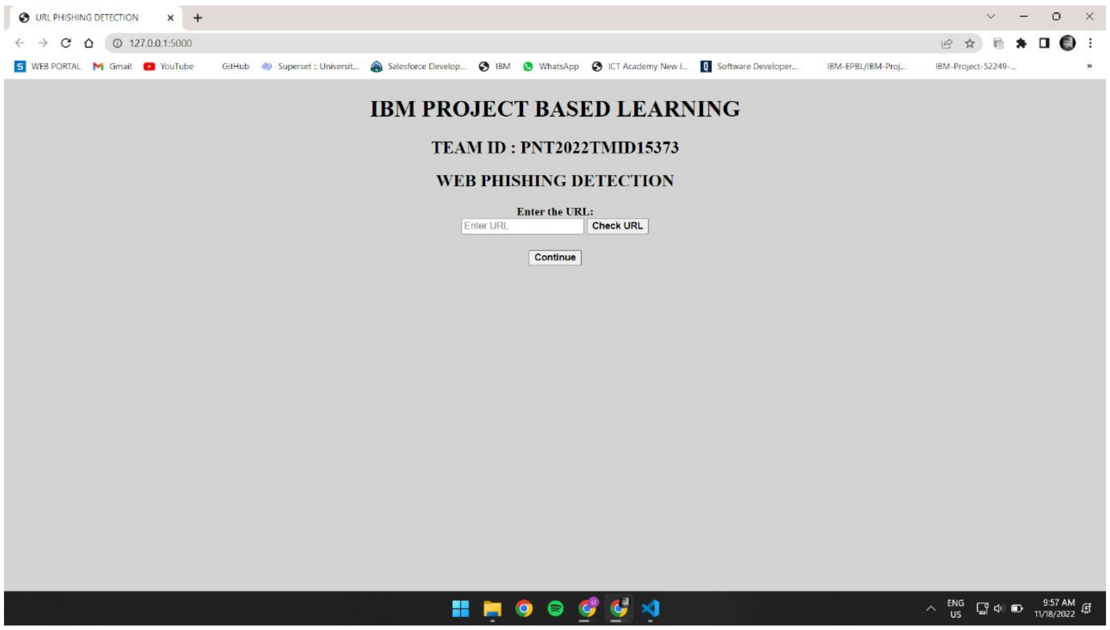
```
Logistic Regression: Accuracy: 92.282
Classification Report :
              precision    recall  f1-score   support

          -1       0.93      0.90      0.91      1498
           1       0.92      0.95      0.93      1819

    accuracy                           0.92      3317
   macro avg       0.92      0.92      0.92      3317
weighted avg       0.92      0.92      0.92      3317
```

Figure 8.1

8.2 USER ACCEPTANCE TESTING



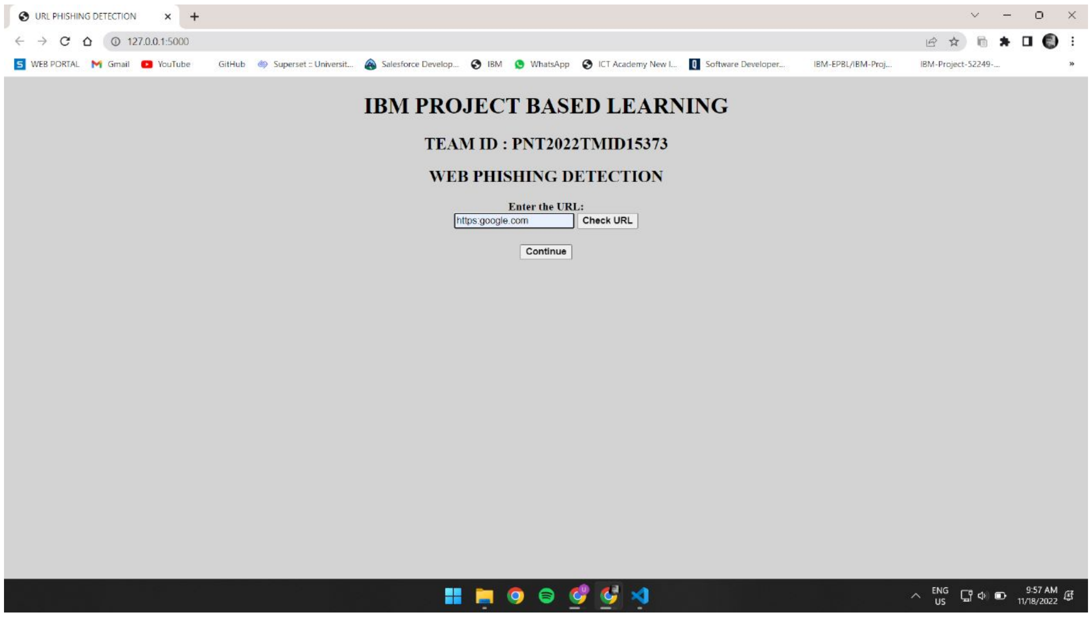
IBM PROJECT BASED LEARNING

TEAM ID : PNT2022TMID15373

WEB PHISHING DETECTION

Enter the URL:

Figure 8.2.1



IBM PROJECT BASED LEARNING

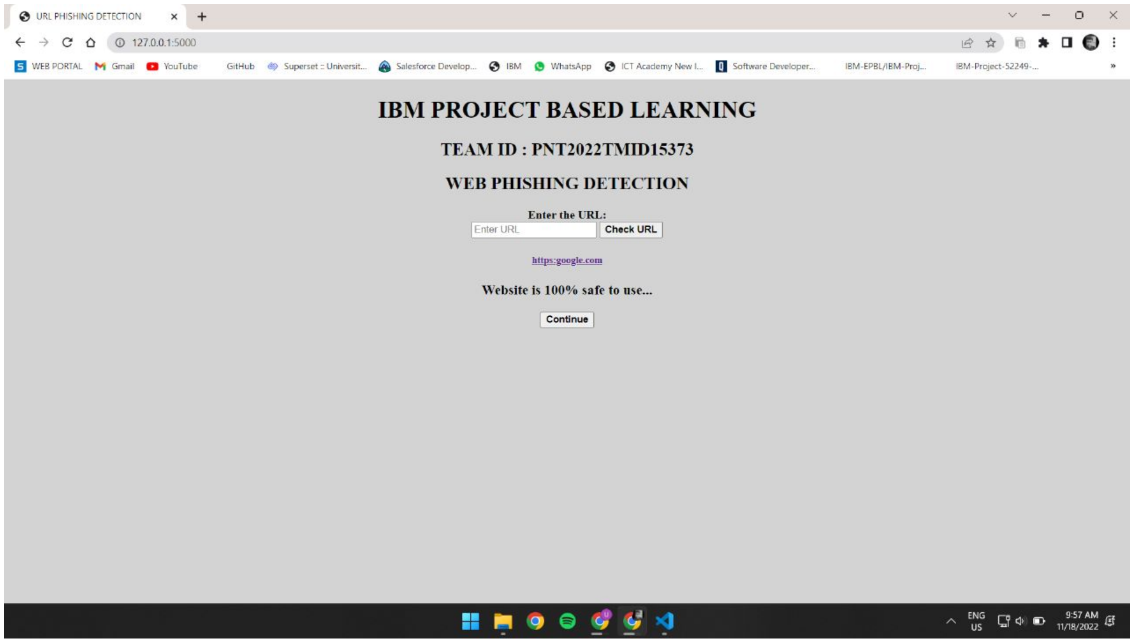
TEAM ID : PNT2022TMID15373

WEB PHISHING DETECTION

Enter the URL:

https:google.com

Figure 8.2.2



IBM PROJECT BASED LEARNING

TEAM ID : PNT2022TMID15373

WEB PHISHING DETECTION

Enter the URL:

https:google.com

Website is 100% safe to use...

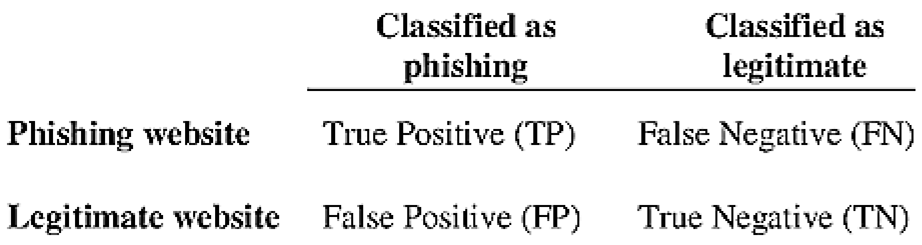
Figure 8.2.3

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS

Confusion Matrix: The confusion matrix is used to measure the introduction of two class issue for the given instructive record. The right corner to corner parts TP(True positive) and TN (True Negative) adequately describe instances similarly as FP (false positive) and FN (false negative) wrongly request instances. Confusion Matrix correctly classify instance TP+TN incorrectly classify instances.

- True positives imply the phishing website that were precisely named by the classifier.
- True negatives are the legitimate website that were precisely set apart by the classifier.
- False positives are the legitimate that were erroneously set apart as phishing website
- False negatives are the phishing website that were incorrectly stamped legitimate website.



	Classified as phishing	Classified as legitimate
Phishing website	True Positive (TP)	False Negative (FN)
Legitimate website	False Positive (FP)	True Negative (TN)

Figure 9.1 Performance Metrics

Classification Report: A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report .

- ➤ Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.

  Precision = TP/(TP + FP)

- ➤ Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

  Recall = TP/(TP+FN)

- ➤ The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

  F1 Score = 2*(Recall * Precision) / (Recall + Precision)

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

A mailbox-level anti-phishing solution offers an additional layer of protection by analyzing account information and understanding users' communication habits. This delivers an enhanced level of phishing protection to detect attacks faster, alert users and remediate threats as quickly as possible.

- Measure the degrees of corporate and employee vulnerability.
- Eliminate the cyber threat risk level.
- Increase user alertness to phishing risks.
- Instill a cyber security culture and create cyber security heroes.

DISADVANTAGES

Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage to reputation, and disruption of operational activities. These effects work together to cause loss of company value, sometimes with irreparable repercussions.Phishing is a problem on two fronts. First, a hacker may gain valuable access to a single account through a successful phishing attempt. Second, if an employee is using the same password for multiple company accounts, then the hacker has now gained access to a great deal of confidential company data.

CHAPTER 11

CONCLUSION

The importance to safeguard online users from becoming victims of online fraud, divulging confidential information to an attacker among other effective uses of phishing as an attacker’s tool. Unfortunately, many of the existing phishing-detection tools, especially those that depend on an existing blacklist, suffer limitations such as low detection accuracy and high false alarm that is often caused by either a delay in blacklist update as a result of human verification process involved in classification or perhaps, it can be attributed to human error in classification which may lead to improper classification of the classes. The inconsistent nature of attacks behaviors and continuously changing URL phish patterns require timely updating of the reference model. Therefore, it requires an effective technique to regulate retraining as to enable machine learning algorithm to actively adapt to the changes in phish patterns.Our phishing detection method focused on the learning process. The outcome of the experiment reached over 92% of Accuracy when websites with Logistic Regression are detected.

CHAPTER 12

FUTURE SCOPE

Phishing attacks are targeting these users depending on the trikes of social engineering. Despite there are several ways to carry out these attacks, unfortunately the current phishing detection techniques cover some attack vectors like email and fake websites.Phishing is when attackers send malicious emails designed to trick people into falling for a scam. Typically, the intent is to get users to reveal financial information, system credentials or other sensitive data.Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage to reputation, and disruption of operational activities. These effects work together to cause loss of company value, sometimes with irreparable repercussions.

CHAPTER 13
APPENDIX

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import pickle

df=pd.read_csv("C:/Users/karthi/Desktop/IBM_NT/PROJECT DEVELOPMENT PHASE/SPRINT
1/Dataset/Dataset_website.csv")

df

df.head()

df.shape

df['Result'].unique()

df.size

df.info()

df.describe()


df.isnull().sum()

def plot_corr(df,size=10):
   corr=df.corr()
   fig,ax=plt.subplots(figsize=(size,size))
   ax.legend()
   cax=ax.matshow(corr)
   fig.colorbar(cax)
   plt.xticks(range(len(corr.columns)), corr.columns, rotation='vertical')
   plt.yticks(range(len(corr.columns)), corr.columns)
plot_corr(df)

with sns.color_palette('muted'):
    sns.countplot(x=df['Result'])

X=df.iloc[:,1:31].valuesfrom sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=0)

X_train.shape

y_train.shape

X_test.shape

y_test.shape

# Creating holders to store the model performance results
ML_Model = []
acc_train = []
acc_test = []

#function to call for storing the results
def storeResults(model, a,b):
  ML_Model.append(model)
  acc_train.append(round(a, 3))
  acc_test.append(round(b, 3))

from sklearn.metrics import accuracy_score, classification_report

from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(X_train,y_train)

y_pred_lr=lr.predict(X_test)
from sklearn.metrics import accuracy_score
y_test_lr = lr.predict(X_test)
y_train_lr = lr.predict(X_train)

acc_train_lr = accuracy_score(y_train,y_train_lr)*100
acc_test_lr = accuracy_score(y_test,y_test_lr)*100

storeResults('Logistic Regression', acc_train_lr, acc_test_lr)
log_reg=accuracy_score(y_test,y_pred_lr)*100
print("Logistic Regression: Accuracy: {:.3f}".format(log_reg))
print("Classification Report : ")
print(classification_report(y_test,y_pred_lr))

# Random Forest model
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
forest = RandomForestClassifier(max_depth=5)

# fit the model
forest.fit(X_train, y_train)

#predicting the target value from the model for the samples
y_pred_forest=forest.predict(X_test)
y_test_forest = forest.predict(X_test)
y_train_forest = forest.predict(X_train)

#computing the accuracy of the model performance
acc_train_forest = accuracy_score(y_train,y_train_forest)*100
acc_test_forest = accuracy_score(y_test,y_test_forest)*100

storeResults('Random Forest', acc_train_forest, acc_test_forest)
ran_forest=accuracy_score(y_test,y_pred_forest)*100
print("Random forest: Accuracy: {:.3f}".format(ran_forest))
print("Classification Report : ")
print(classification_report(y_test,y_pred_forest))

#Support vector machine model
from sklearn.svm import SVC

# instantiate the model
svm = SVC(kernel='linear', C=1.0, random_state=12)
#fit the model
svm.fit(X_train, y_train)

#predicting the target value from the model for the samples
y_pred_svm=svm.predict(X_test)
y_test_svm = svm.predict(X_test)
y_train_svm = svm.predict(X_train)
```

```
#computing the accuracy of the model performance
acc_train_svm = accuracy_score(y_train,y_train_svm)*100
acc_test_svm = accuracy_score(y_test,y_test_svm)*100

svm_acc=accuracy_score(y_test,y_pred_forest)*100
print("Random forest: Accuracy: {:.3f}".format(svm_acc))
print("Classification Report : ")
print(classification_report(y_test,y_pred_svm))
storeResults('SVM', acc_train_svm, acc_test_svm)

#Decision Tree Classifier model
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(X_train,y_train)

#predicting the target value from the model for the samples
y_pred_dt=dt.predict(X_test)
y_test_dt = dt.predict(X_test)
y_train_dt = dt.predict(X_train)

#computing the accuracy of the model performance
acc_train_dt = accuracy_score(y_train,y_train_dt)*100
acc_test_dt = accuracy_score(y_test,y_test_dt)*100

dt_acc=accuracy_score(y_test,y_pred_forest)*100
print("Random forest: Accuracy: {:.3f}".format(dt_acc))
print("Classification Report : ")
print(classification_report(y_test,y_pred_dt))
storeResults('Decision Tree Classifier', acc_train_dt, acc_test_dt)

results = pd.DataFrame({ 'ML Model': ML_Model,
   'Train Accuracy': acc_train,
   'Test Accuracy': acc_test})
results

import matplotlib.pyplot as plt
plt.bar(ML_Model,acc_test,width=0.3,color=['green','blue','red','orange'])

pickle.dump(lr,open('Phishing.pkl','wb'))
```

```
loaded_model=pickle.load(open('Phishing.pkl','rb'))
res=loaded_model.score(X_test,y_test_lr)
print(res)
```

app.py

```
#importing required libraries

from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
warnings.filterwarnings('ignore')
from feature import FeatureExtraction

file = open("Phishing.pkl","rb")
gbc = pickle.load(file)
file.close()


app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def index():
    if request.method == "POST":

        url = request.form["url"]
        obj = FeatureExtraction(url)
        x = np.array(obj.getFeaturesList()).reshape(1,30)

        y_pred =gbc.predict(x)[0]
        #1 is safe
        #-1 is unsafe
        y_pro_phishing = gbc.predict_proba(x)[0,0]
        y_pro_non_phishing = gbc.predict_proba(x)[0,1]
        # if(y_pred ==1 ):
        pred = "It is {0:.2f} % safe to go ".format(y_pro_phishing*100)
        return render_template('index.html',xx =round(y_pro_non_phishing,2),url=url )
    return render_template("index.html", xx =-1)


if __name__ == "__main__":
    app.run(debug=True)
```

feature.py

```
import ipaddress
import re
import urllib.request
from bs4 import BeautifulSoup
import socket
import requests
from googlesearch import search
import whois
from datetime import date, datetime
import time
from dateutil.parser import parse as date_parse
from urllib.parse import urlparse

class FeatureExtraction:
    features = []
    def __init__(self,url):
        self.features = []
        self.url = url
        self.domain = ""
        self.whois_response = ""
        self.urlparse = ""
        self.response = ""
        self.soup = ""

        try:
            self.response = requests.get(url)
            self.soup = BeautifulSoup(response.text, 'html.parser')
        except:
            pass

        try:
            self.urlparse = urlparse(url)
            self.domain = self.urlparse.netloc
        except:
            pass

        try:
            self.whois_response = whois.whois(self.domain)
        except:
            pass




        self.features.append(self.UsingIp())
        self.features.append(self.longUrl())
        self.features.append(self.shortUrl())
        self.features.append(self.symbol())
        self.features.append(self.redirecting())
        self.features.append(self.prefixSuffix())
        self.features.append(self.SubDomains())
        self.features.append(self.Hppts())
        self.features.append(self.DomainRegLen())
        self.features.append(self.Favicon())


        self.features.append(self.NonStdPort())
        self.features.append(self.HTTPSDomainURL())
        self.features.append(self.RequestURL())
        self.features.append(self.AnchorURL())
        self.features.append(self.LinksInScriptTags())
        self.features.append(self.ServerFormHandler())
        self.features.append(self.InfoEmail())
        self.features.append(self.AbnormalURL())
        self.features.append(self.WebsiteForwarding())
        self.features.append(self.StatusBarCust())

        self.features.append(self.DisableRightClick())
        self.features.append(self.UsingPopupWindow())
        self.features.append(self.IframeRedirection())
        self.features.append(self.AgeofDomain())
        self.features.append(self.DNSRecording())
        self.features.append(self.WebsiteTraffic())
```

```
        self.features.append(self.PageRank())
        self.features.append(self.GoogleIndex())
        self.features.append(self.LinksPointingToPage())
        self.features.append(self.StatsReport())


     # 1.UsingIp
    def UsingIp(self):
        try:
            ipaddress.ip_address(self.url)
            return -1
        except:
            return 1

    # 2.longUrl
    def longUrl(self):
        if len(self.url) < 54:
            return 1
        if len(self.url) >= 54 and len(self.url) <= 75:
            return 0
        return -1

    # 3.shortUrl
    def shortUrl(self):
        match =
re.search('bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyurl|tr\.im|is\.gd|cli\.gs|'
                    'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.pr|twurl\.nl|snipurl\.com|'

'short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|snipr\.com|fic\.kr|loopt\.us|'
                    'doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|om\.ly|to\.ly|bit\.do|t\.co|lnkd\.in|'
                    'db\.tt|qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.com|ow\.ly|bit\.ly|ity\.im|'

'q\.gs|is\.gd|po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|buzurl\.com|cutt\.us|u\.bb|yourls\.org|'

'x\.co|prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.net|1url\.com|tweez\.me|v\.gd
|tr\.im|link\.zip\.net', self.url)
        if match:
            return -1
        return 1

    # 4.Symbol@
```

```
def symbol(self):
    if re.findall("@",self.url):
        return -1
    return 1

# 5.Redirecting//
def redirecting(self):
    if self.url.rfind('//')>6:
        return -1
    return 1

# 6.prefixSuffix
def prefixSuffix(self):
    try:
        match = re.findall('\-', self.domain)
        if match:
            return -1
        return 1
    except:
        return -1

# 7.SubDomains
def SubDomains(self):
    dot_count = len(re.findall("\.", self.url))
    if dot_count == 1:
        return 1
    elif dot_count == 2:
        return 0
    return -1

# 8.HTTPS
def Hppts(self):
    try:
        https = self.urlparse.scheme
        if 'https' in https:
            return 1
        return -1
    except:
        return 1

# 9.DomainRegLen
```

```
    def DomainRegLen(self):
        try:
            expiration_date = self.whois_response.expiration_date
            creation_date = self.whois_response.creation_date
            try:
                if(len(expiration_date)):
                    expiration_date = expiration_date[0]
            except:
                pass
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            age = (expiration_date.year-creation_date.year)*12+ (expiration_date.month-
creation_date.month)
            if age >=12:
                return 1
            return -1
        except:
            return -1

    # 10. Favicon
    def Favicon(self):
        try:
            for head in self.soup.find_all('head'):
                for head.link in self.soup.find_all('link', href=True):
                    dots = [x.start(0) for x in re.finditer('\.', head.link['href'])]
                    if self.url in head.link['href'] or len(dots) == 1 or domain in head.link['href']:
                        return 1
            return -1
        except:
            return -1

    # 11. NonStdPort
    def NonStdPort(self):
        try:
            port = self.domain.split(":")
            if len(port)>1:
                return -1
        return 1
    except:
        return -1

# 12. HTTPSDomainURL
def HTTPSDomainURL(self):
    try:
        if 'https' in self.domain:
            return -1
        return 1
    except:
        return -1

# 13. RequestURL
def RequestURL(self):
    try:
        for img in self.soup.find_all('img', src=True):
            dots = [x.start(0) for x in re.finditer('\.', img['src'])]
            if self.url in img['src'] or self.domain in img['src'] or len(dots) == 1:
                success = success + 1
            i = i+1

        for audio in self.soup.find_all('audio', src=True):
            dots = [x.start(0) for x in re.finditer('\.', audio['src'])]
            if self.url in audio['src'] or self.domain in audio['src'] or len(dots) == 1:
                success = success + 1
            i = i+1

        for embed in self.soup.find_all('embed', src=True):
            dots = [x.start(0) for x in re.finditer('\.', embed['src'])]
            if self.url in embed['src'] or self.domain in embed['src'] or len(dots) == 1:
                success = success + 1
            i = i+1

        for iframe in self.soup.find_all('iframe', src=True):
            dots = [x.start(0) for x in re.finditer('\.', iframe['src'])]
            if self.url in iframe['src'] or self.domain in iframe['src'] or len(dots) == 1:
                success = success + 1
            i = i+1

        try:
                percentage = success/float(i) * 100
                if percentage < 22.0:
                    return 1
                elif((percentage >= 22.0) and (percentage < 61.0)):
                    return 0
                else:
                    return -1
            except:
                return 0
        except:
            return -1

    # 14. AnchorURL
    def AnchorURL(self):
        try:
            i,unsafe = 0,0
            for a in self.soup.find_all('a', href=True):
                if "#" in a['href'] or "javascript" in a['href'].lower() or "mailto" in a['href'].lower() or not (url
in a['href'] or self.domain in a['href']):
                    unsafe = unsafe + 1
                i = i + 1

            try:
                percentage = unsafe / float(i) * 100
                if percentage < 31.0:
                    return 1
                elif ((percentage >= 31.0) and (percentage < 67.0)):
                    return 0
                else:
                    return -1
            except:
                return -1

        except:
            return -1

    # 15. LinksInScriptTags
    def LinksInScriptTags(self):
        try:
            i,success = 0,0
        for link in self.soup.find_all('link', href=True):
            dots = [x.start(0) for x in re.finditer('\.', link['href'])]
            if self.url in link['href'] or self.domain in link['href'] or len(dots) == 1:
                success = success + 1
            i = i+1

        for script in self.soup.find_all('script', src=True):
            dots = [x.start(0) for x in re.finditer('\.', script['src'])]
            if self.url in script['src'] or self.domain in script['src'] or len(dots) == 1:
                success = success + 1
            i = i+1

        try:
            percentage = success / float(i) * 100
            if percentage < 17.0:
                return 1
            elif((percentage >= 17.0) and (percentage < 81.0)):
                return 0
            else:
                return -1
        except:
            return 0
    except:
        return -1

# 16. ServerFormHandler
def ServerFormHandler(self):
    try:
        if len(self.soup.find_all('form', action=True))==0:
            return 1
        else :
            for form in self.soup.find_all('form', action=True):
                if form['action'] == "" or form['action'] == "about:blank":
                    return -1
                elif self.url not in form['action'] and self.domain not in form['action']:
                    return 0
                else:
                    return 1
    except:
        return -1
```

```
# 17. InfoEmail
def InfoEmail(self):
    try:
        if re.findall(r"[mail\(\)|mailto:?]", self.soap):
            return -1
        else:
            return 1
    except:
        return -1

# 18. AbnormalURL
def AbnormalURL(self):
    try:
        if self.response.text == self.whois_response:
            return 1
        else:
            return -1
    except:
        return -1

# 19. WebsiteForwarding
def WebsiteForwarding(self):
    try:
        if len(self.response.history) <= 1:
            return 1
        elif len(self.response.history) <= 4:
            return 0
        else:
            return -1
    except:
         return -1

# 20. StatusBarCust
def StatusBarCust(self):
    try:
        if re.findall("<script>.+onmouseover.+</script>", self.response.text):
            return 1
        else:
            return -1
    except:
         return -1
```

```
# 21. DisableRightClick
def DisableRightClick(self):
    try:
        if re.findall(r"event.button ?== ?2", self.response.text):
            return 1
        else:
            return -1
    except:
         return -1

# 22. UsingPopupWindow
def UsingPopupWindow(self):
    try:
        if re.findall(r"alert\(", self.response.text):
            return 1
        else:
            return -1
    except:
         return -1

# 23. IframeRedirection
def IframeRedirection(self):
    try:
        if re.findall(r"[<iframe>|<frameBorder>]", self.response.text):
            return 1
        else:
            return -1
    except:
         return -1

# 24. AgeofDomain
def AgeofDomain(self):
    try:
        creation_date = self.whois_response.creation_date
        try:
            if(len(creation_date)):
                creation_date = creation_date[0]
        except:
            pass
            today  = date.today()
            age = (today.year-creation_date.year)*12+(today.month-creation_date.month)
            if age >=6:
                return 1
            return -1
        except:
            return -1

    # 25. DNSRecording
    def DNSRecording(self):
        try:
            creation_date = self.whois_response.creation_date
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            today  = date.today()
            age = (today.year-creation_date.year)*12+(today.month-creation_date.month)
            if age >=6:
                return 1
            return -1
        except:
            return -1

    # 26. WebsiteTraffic
    def WebsiteTraffic(self):
        try:
            rank =
BeautifulSoup(urllib.request.urlopen("http://data.alexa.com/data?cli=10&dat=s&url=" +
url).read(), "xml").find("REACH")['RANK']
            if (int(rank) < 100000):
                return 1
            return 0
        except :
            return -1

    # 27. PageRank
    def PageRank(self):
        try:
            prank_checker_response = requests.post("https://www.checkpagerank.net/index.php",
{"name": self.domain})

            global_rank = int(re.findall(r"Global Rank: ([0-9]+)", rank_checker_response.text)[0])
            if global_rank > 0 and global_rank < 100000:
                return 1
            return -1
        except:
            return -1


    # 28. GoogleIndex
    def GoogleIndex(self):
        try:
            site = search(self.url, 5)
            if site:
                return 1
            else:
                return -1
        except:
            return 1

    # 29. LinksPointingToPage
    def LinksPointingToPage(self):
        try:
            number_of_links = len(re.findall(r"<a href=", self.response.text))
            if number_of_links == 0:
                return 1
            elif number_of_links <= 2:
                return 0
            else:
                return -1
        except:
            return -1

    # 30. StatsReport
    def StatsReport(self):
        try:
            url_match = re.search(

'at\.ua|usa\.cc|baltazarpresentes\.com\.br|pe\.hu|esy\.es|hol\.es|sweddy\.com|myjino\.ru|96\.lt
|ow\.ly', url)
            ip_address = socket.gethostbyname(self.domain)
            ip_match =
re.search('146\.112\.61\.108|213\.174\.157\.151|121\.50\.168\.88|192\.185\.217\.116|78\.46\.
211\.158|181\.174\.165\.13|46\.242\.145\.103|121\.50\.168\.40|83\.125\.22\.219|46\.242\.14
5\.98|'

'107\.151\.148\.44|107\.151\.148\.107|64\.70\.19\.203|199\.184\.144\.27|107\.151\.148\.108|
107\.151\.148\.109|119\.28\.52\.61|54\.83\.43\.69|52\.69\.166\.231|216\.58\.192\.225|'

'118\.184\.25\.86|67\.208\.74\.71|23\.253\.126\.58|104\.239\.157\.210|175\.126\.123\.219|14
1\.8\.224\.221|10\.10\.10\.10|43\.229\.108\.32|103\.232\.215\.140|69\.172\.201\.153|'

'216\.218\.185\.162|54\.225\.104\.146|103\.243\.24\.98|199\.59\.243\.120|31\.170\.160\.61|2
13\.19\.128\.77|62\.113\.226\.131|208\.100\.26\.234|195\.16\.127\.102|195\.16\.127\.157|'

'34\.196\.13\.28|103\.224\.212\.222|172\.217\.4\.225|54\.72\.9\.51|192\.64\.147\.141|198\.20
0\.56\.183|23\.253\.164\.103|52\.48\.191\.26|52\.214\.197\.72|87\.98\.255\.18|209\.99\.17\.2
7|'

'216\.38\.62\.18|104\.130\.124\.96|47\.89\.58\.141|78\.46\.211\.158|54\.86\.225\.156|54\.82\
.156\.19|37\.157\.192\.102|204\.11\.56\.48|110\.34\.231\.42', ip_address)
            if url_match:
                return -1
            elif ip_match:
                return -1
            return 1
        except:
            return 1

    def getFeaturesList(self):
        return self.features
```

Index.html

```
<!DOCTYPE html>
<html>
<head>
    <center><h1> IBM PROJECT BASED LEARNING</h1></center>
    <center><h2> TEAM ID : PNT2022TMID15373 </h2></center>
    <title> URL PHISHING DETECTION </title>
    <style>
        body{
            background-color:lightgrey;
        }
    </style>
</head>

<body>
    <center><h2> WEB PHISHING DETECTION </h2></center>
    <center>
    <form action="/" method="post">
        <label for="url" class="form__label"><b>Enter the URL:</b></label><br>
        <input type="text" class="form__input" name ='url' id="url" placeholder="Enter URL"
required="" />

        <button class="button" role="button" ><b>Check URL</b></button>
    </form>
        <h5><a href= {{ url }} target="_blank">{{ url }}</a></h5>
        <h3 id="prediction"></h3>
        <button class="button1" id="button1" role="button"  onclick="window.open('{{url}}')"
target="_blank"><b>Continue</b></button>
        <!--<button class="button2" id="button2" role="button" onclick="window.open('{{url}}')"
target="_blank" >Still want to Continue</button>-->

    </center>

    <!-- JavaScript -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
        integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
        crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
        integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
        crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
        integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
        crossorigin="anonymous"></script>
    <script>
        let x = '{{xx}}';
        let num = x*100;
        if (0<=x && x<0.50){
            num = 100-num;
        }
        let txtx = num.toString();
        if(x<=1 && x>=0.50){
            var label = "Website is "+txtx +"% safe to use...";
            document.getElementById("prediction").innerHTML = label;
            document.getElementById("button2").style.display="block";
        }
        else if (0<=x && x<0.50){
            var label = "Website is "+txtx +"% unsafe to use..."
            document.getElementById("prediction").innerHTML = label ;
            document.getElementById("button1").style.display="block";
        }
    </script>
</body>
</html>
```

Github and project video demo link

Github link:

https://github.com/IBM-EPBL/IBM-Project-20967-1659768167

Project video demo link:

https://drive.google.com/drive/folders/1ENz-iBPhKMzALwlI3qNnQXUeFucdzyoW?usp=share_link