WEB PHISHING DETECTION USING MACHINE LEARNING

TEAM ID: PNT2022TMID08662

Team Leader: DHARUN ADITYA [727619BCS102]

Team Member 1: SHARATHSOORYA [727619BCS050]

Team Member 2: PERUMAL [727619BCS046]

Team Member 3: TAMILARASAN [727619BCS062]

WEB PHISHING DETECTION REPORT

1. INTRODUCTION

1.1 Project Overview

Internet consumers lose billions of dollars each year due to phishing. In order to fish for personal information in a pool of naive Internet users, identity thieves use luring strategies. To acquire usernames and passwords for financial accounts as well as personal information, phishers employ faked emails and phishing software. The topic of this study is how to use machine learning techniques to analyse different characteristics of legitimate and phishing URLs to identify phishing websites.

Nowadays Phishing has become a main area of concern for security researchers because it is very easy to create fake websites which look very similar to legitimate websites. Experts can identify fake websites but not all the users can identify the fake website and such users become victims of phishing attack. To overcome this drawback, we have proposed an intelligent, flexible, and effective system that is based on using classification data mining algorithm to analyse various URLs to accurately detect phishing websites.

The main aim of the attacker is usually to steal banks account credentials. If this continues, clients who become victims to phishing will face huge financial losses. So, to protect internet users from these kinds of phishing attacks and to create awareness, phishing website detection system has been implemented.

The Phishing website detection system is a web-based application which can run on any web browser. Every internet user will have the ability to detect phishing websites by entering URLs which they suspect of being phishing links. Those reported URLs are verified and rated.

1.2 Purpose

The purpose of the project is to determine whether a given URL is phishing website. This is done by building a Machine Learning model that is trained and tested on 11,056 rows of data with 32 different attributes. The machine learning algorithm used to carry this out is Logistic Regression classifier. The model will classify any given URL as safe or unsafe. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of certain classes based on some dependent variables.

2. LITERATURE SURVEY

2.1 Existing problem

Phishing scams are a form of cybercrime that involves defrauding users to obtain sensitive information. Cybercriminals act as legitimate companies or organizations to obtain the information. Phishing remains cybercriminals’ method-of-choice to infect users’ computers. Corporate employees are particularly vulnerable since they are heavily targeted as an easy entry into sensitive data. Cybercriminals use social engineering to trick their victims into launching malicious files on their computers, opening a link to an infected website or sending criminals their private data.

Phishing scams involve sending out emails or texts disguised as legitimate sources. They may look like they are from a trusted vendor or a law enforcement authority, but secretly, they contain malware. These messages are specifically designed to trick the victim into opening the email through the tactics of fear and intimidation. Once a person opens it, the malicious software downloads onto their computer, and the cybercriminal is in your system. Common social engineering methods include sending messages with embedded URLs. Once the person clicks on the link, they are re-directed to a phishing site. A phishing email can be sent with a malicious attachment that is rigged with exploits, often with the claim that the attachment is an unpaid invoice that needs attention.

According to recent research from Iron Scales, 81% of organizations around the world have experienced an increase in email phishing attacks since March 2020. Despite the very real threat that phishing poses to businesses today, almost 1 in 5 organizations only deliver phishing awareness training to their employees once per year. This lack of awareness is a large contributing factor to the fact that phishing remains the threat type most likely to cause a data breach.

2.2 References

[1]. Mehmet Korkmaz, Ozgur KoraySahingoz, BanuDiri, “Detection of Phishing Websites by Using Machine Learning-Based URL Analysis”, 2020.

[2]. Lizhen Tang , Qusay H. Mahmoud, “A Deep Learning-Based Framework for Phishing Website Detection”, 2021

[3]. BuketGeyik, Kubra Erensoy, EmreKocyigit, “Detection of Phishing Websites from URLs by using Classification Techniques on WEKA”, 2021

[4]. Abdulghani Ali Ahmed, Nurul Amirah Abdullah, “Real Time Detection of Phishing Websites”, 2016

[5]. Manuel sánchez-paniagua , eduardo fidalgo fernández ,enrique alegre ,wesam alnabki, víctor gonzález-castro, “Phishing URL Detection: A Real-Case Scenario Through Login URLs”, 2022

[6]. Ishant Tyagi, Jatin Shad, Shubham Sharma, Siddharth Gaur, Gagandeep Kaur, “A Novel Machine Learning Approach to Detect Phishing Websites”, 2018

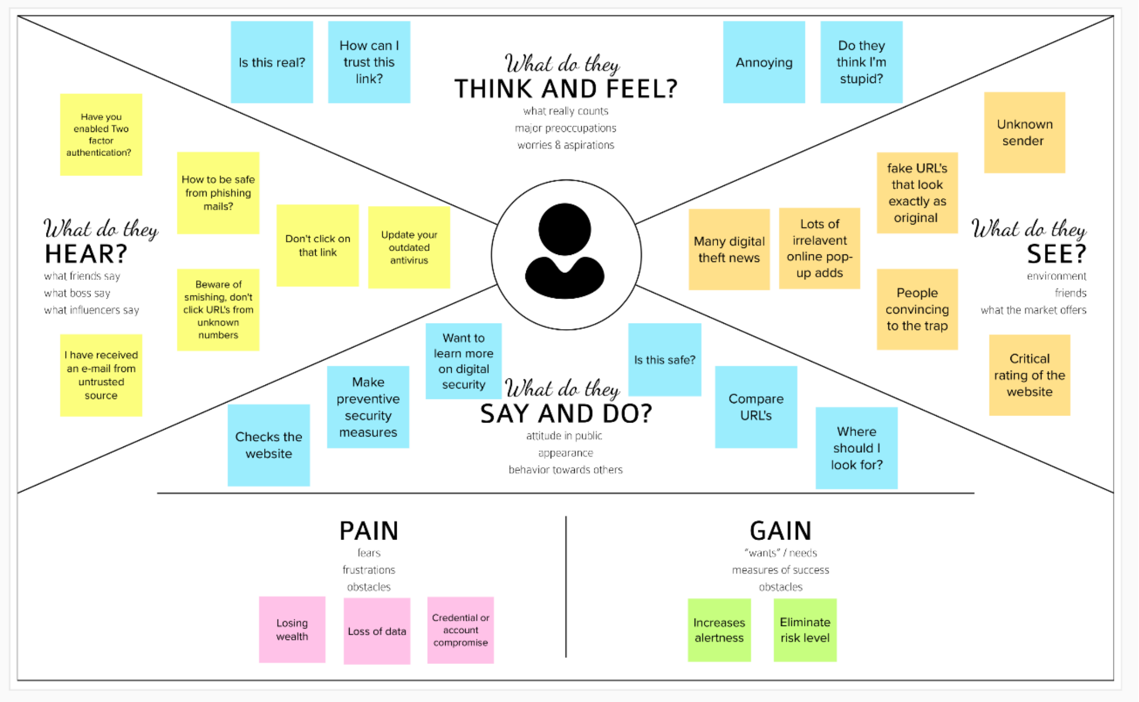
2.3 Problem Statement Definition

This project identifies whether a given URL is a phishing website. This is accomplished by developing a machine learning model that uses 11,056 rows of data with 32 different attributes which include: URL length, HTTPS token, web traffic, google index and age of domain among other attributes. The model is built using 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. Here, the dependant variable is the safety status of a given URL i.e., safe and unsafe.

Problem Statement (PS)	**I am (Customer)**	**I’m trying to**	**But**	**Because**	**Which makes me feel**
PS-1	A buyer	Make payment for the product	The payment details are tampered	The website is not secured	Helplessness
PS-2	A Gamer	Purchase loots and upgrades for the character	It always directing to third-part website	The game application is infected with Malware	How can I trust this link?

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

Is this real?

How can I trust this link?

Annoying

Do they think I'm stupid?

What do they HEAR?
what friends say
what boss say
what influencers say

Have you enabled Two factor authentication?

How to be safe from phishing mails?

Don't click on that link

Update your outdated antivirus

Beware of smishing, don't click URL's from unknown numbers

I have received an e-mail from untrusted source

What do they SEE?
environment
friends
what the market offers

Unknown sender

fake URL's that look exactly as original

Many digital theft news

Lots of irrelevant online pop-up adds

People convincing to the trap

Critical rating of the website

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

Want to learn more on digital security

Make preventive security measures

Checks the website

Is this safe?

Compare URL's

Where should I look for?

PAIN
fears
frustrations
obstacles

Losing wealth

Loss of data

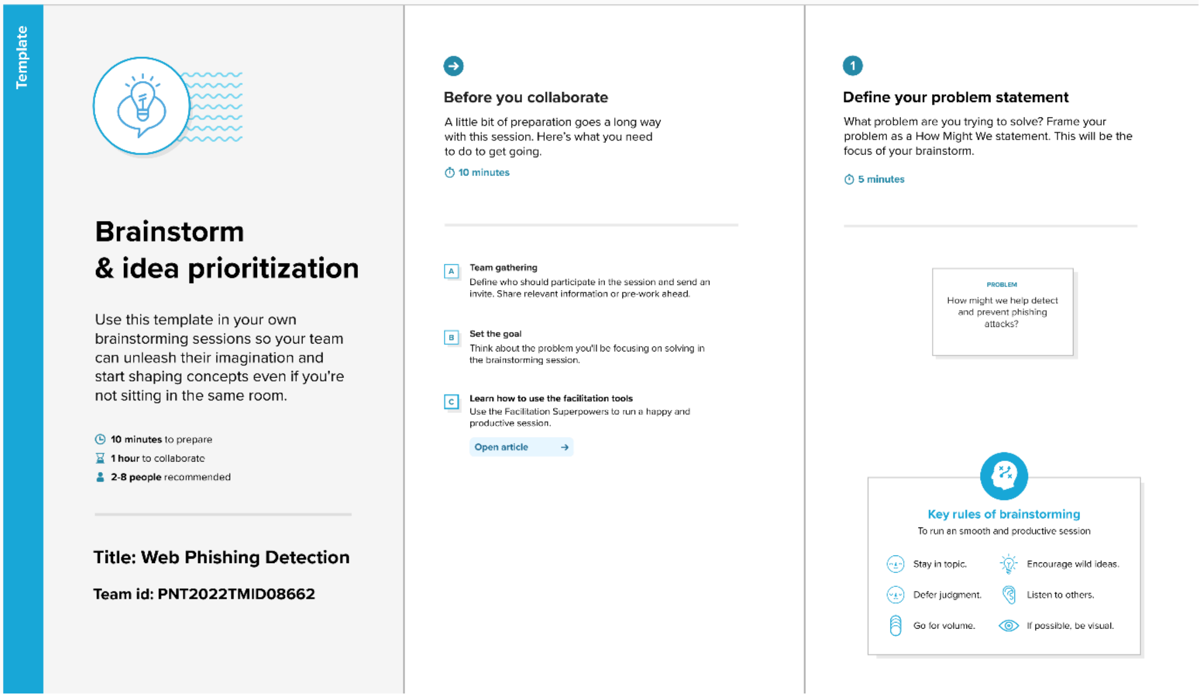
Credential or account compromise

GAIN
"wants" / needs
measures of success
obstacles

Increases alertness

Eliminate risk level

3.2 Ideation & Brainstorming



Template

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

Title: Web Phishing Detection

Team id: PNT2022TMID08662

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

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

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

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

Open article →

1 **Define your problem statement**

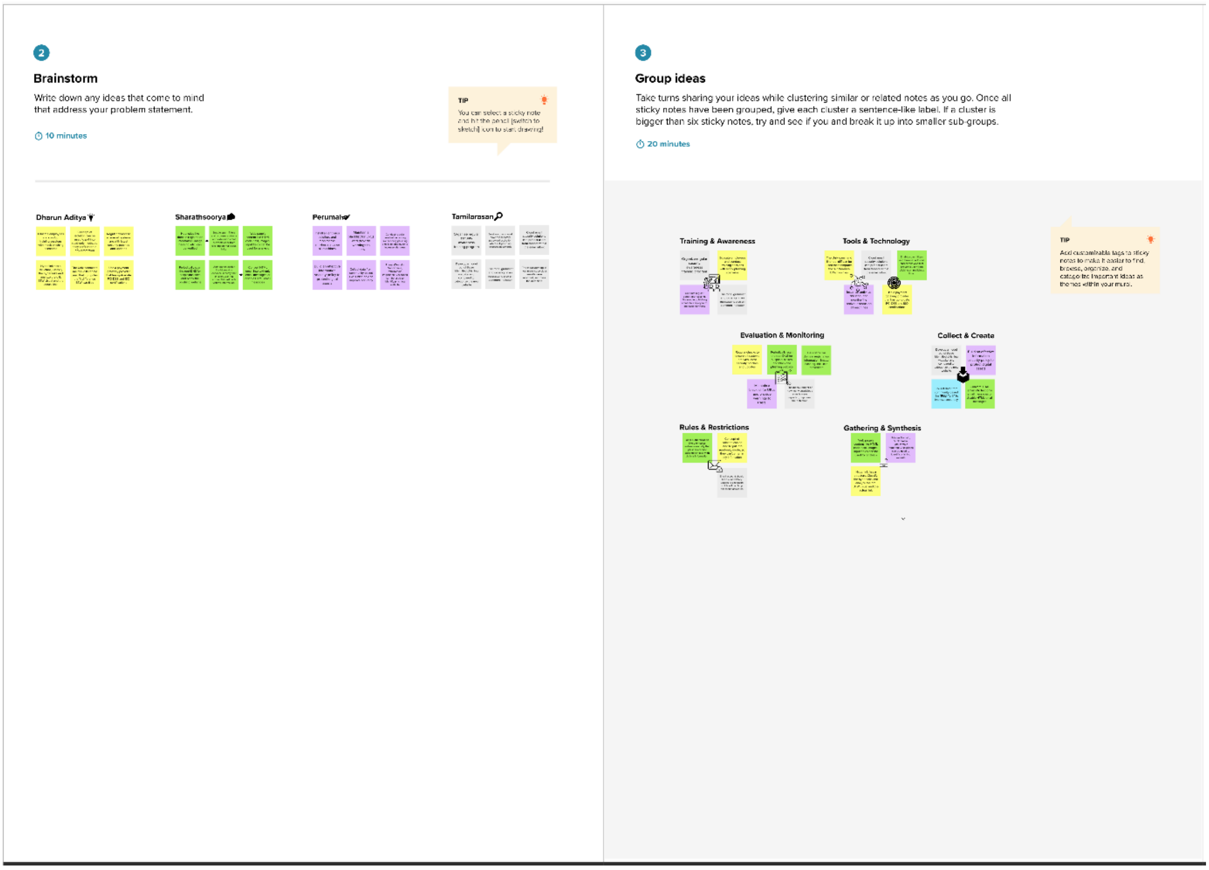
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
How might we help detect and prevent phishing attacks?

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.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

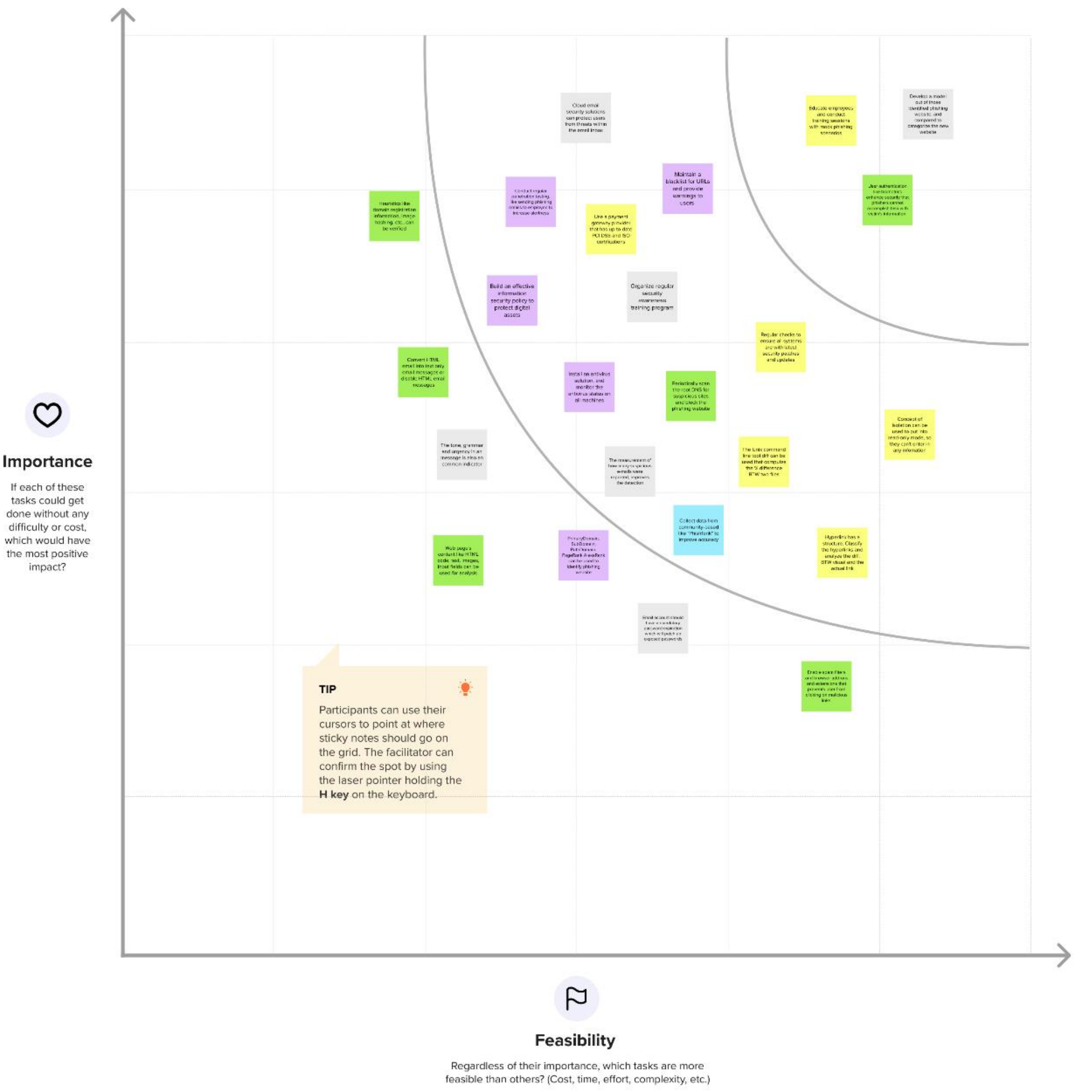
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes


Cloud email security solutions can protect users from threats within the email inbox
Educate employees and conduct training sessions with mock phishing scenarios
Develop a model out of those identified phishing website and compared to categorize the new website
Heuristics like domain registration information, image hashing, etc., can be verified
Conduct regular penetration testing, like sending phishing emails to employees to increase alertness
Use a payment gateway provider that has up to date PCI DSS and ISO certifications
Maintain a blacklist for URLs and provide warnings to users
User authentication mechanisms enhance security that phishers cannot accomplish this with victim's information
Build an effective information security policy to protect digital assets
Organize regular security awareness training program
Regular checks to ensure all systems are with latest security patches and updates
Convert HTML email into text only email messages or disable HTML email messages
Install an antivirus solution, and monitor the antivirus status on all machines
Periodically scan the root DNS for suspicious sites and block the phishing website
Concept of isolation can be used to put into read-only mode, so they can't enter in any information
The tone, grammar and urgency in an message is also an common indicator
The measurement of how many suspicious e-mails were reported, improves the detection
The Unix command line tool diff can be used that computes the % difference BTW two files
Collect data from community-based like "PhishTank" to improve accuracy
Web page's content like HTML code, text, images, input fields can be used for analysis
PrimaryDomain, SubDomain, PathDomain, PageRank AlexaRank can be used to identify phishing website
Hyperlink has a structure. Classify the hyperlinks and analyze the diff. BTW visual and the actual link
Email account should have a mandatory password expiration which will patch up exposed passwords
Enable spam filters and browser add-ons and extensions that prevents user from clicking on malicious links
Importance
Feasibility

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

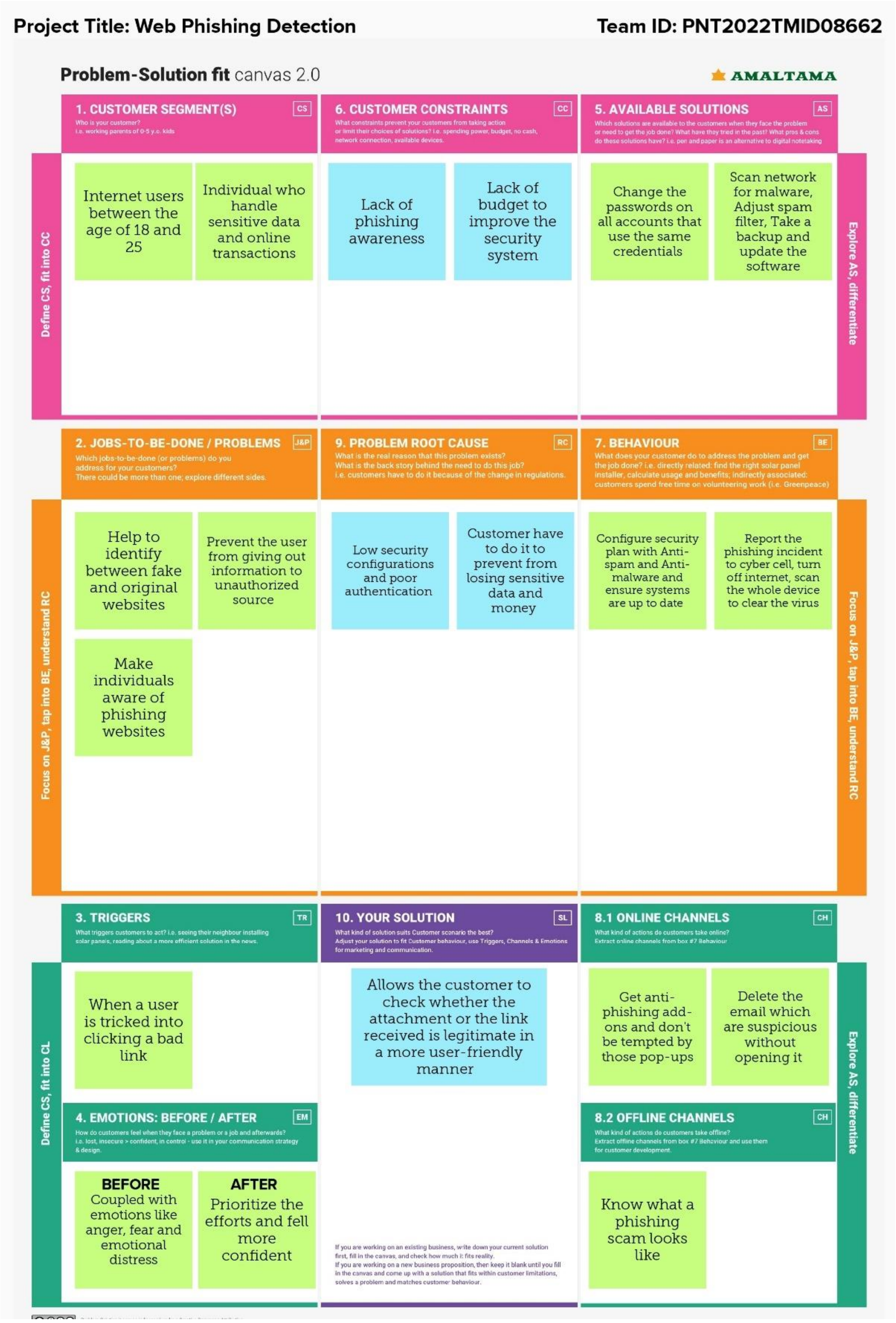
Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The Phish report states that around 74% people were sent fraudulent messages every month. While this cannot be stopped completely, some preventable actions can be taken. To prevent and predict phishing websites, we proposed an intelligent, flexible, and effective system that is based on using classification Data mining algorithm.
2.	Idea / Solution description	In a replicated website there must have some flaws, The phishing website can be detected based on some important characteristics like URL and Domain Identity, and security and encryption criteria in the final phishing detection rate.
3.	Novelty / Uniqueness	In this techy world, there are many technologies offer solution to protect ourselves from phishing attacks, But the data mining algorithm used in this system provides better performance as compared to other traditional classification algorithms.
4.	Social Impact / Customer Satisfaction	The proposes help the user to safely make online transaction without any fear of losing money or sensitive data to the attacker and help them gain some awareness of cyber-threat.
5.	Business Model (Revenue Model)	The number of visitors to the website becomes the number of opportunities the business has at giving an impression, generating qualified leads, sharing the brand, and building relationship.
6.	Scalability of the Solution	The features can progressively increase to scan the attachment, file hash, IP address, etc.,

3.4 Problem Solution Fit



Project Title: Web Phishing Detection **Team ID: PNT2022TMID08662**

Problem-Solution fit canvas 2.0

AMALTAMA

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

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

Internet users between the age of 18 and 25

Individual who handle sensitive data and online transactions

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.

Lack of phishing awareness

Lack of budget to improve the security system

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Change the passwords on all accounts that use the same credentials

Scan network for malware, Adjust spam filter, Take a backup and update the software

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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.

Help to identify between fake and original websites

Prevent the user from giving out information to unauthorized source

Make individuals aware of phishing websites

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.

Low security configurations and poor authentication

Customer have to do it to prevent from losing sensitive data and money

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Configure security plan with Anti-spam and Anti-malware and ensure systems are up to date

Report the phishing incident to cyber cell, turn off internet, scan the whole device to clear the virus

Focus on J&P, tap into BE, understand RC

Define CS, fit into CL

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

When a user is tricked into clicking a bad link

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

BEFORE
Coupled with emotions like anger, fear and emotional distress

AFTER
Prioritize the efforts and fell more confident

10. YOUR SOLUTION SL

What kind of solution suits Customer scenario the best?
Adjust your solution to fit Customer behaviour, use Triggers, Channels & Emotions for marketing and communication.

Allows the customer to check whether the attachment or the link received is legitimate in a more user-friendly manner

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

8.1 ONLINE CHANNELS CH

What kind of actions do customers take online?
Extract online channels from box #7 Behaviour

Get anti-phishing add-ons and don't be tempted by those pop-ups

Delete the email which are suspicious without opening it

8.2 OFFLINE CHANNELS CH

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

Know what a phishing scam looks like

Explore AS, differentiate

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	User Authentication	Authentication via Password
FR-4	User Input	The suspicious URL is entered to check its status
FR-5	Reporting	The latest phishing URL can be reported for further verification if the accuracy is not satisfied
FR-6	Result/output	Model after comparison and analysis displays the safe/unsafe message with percentage

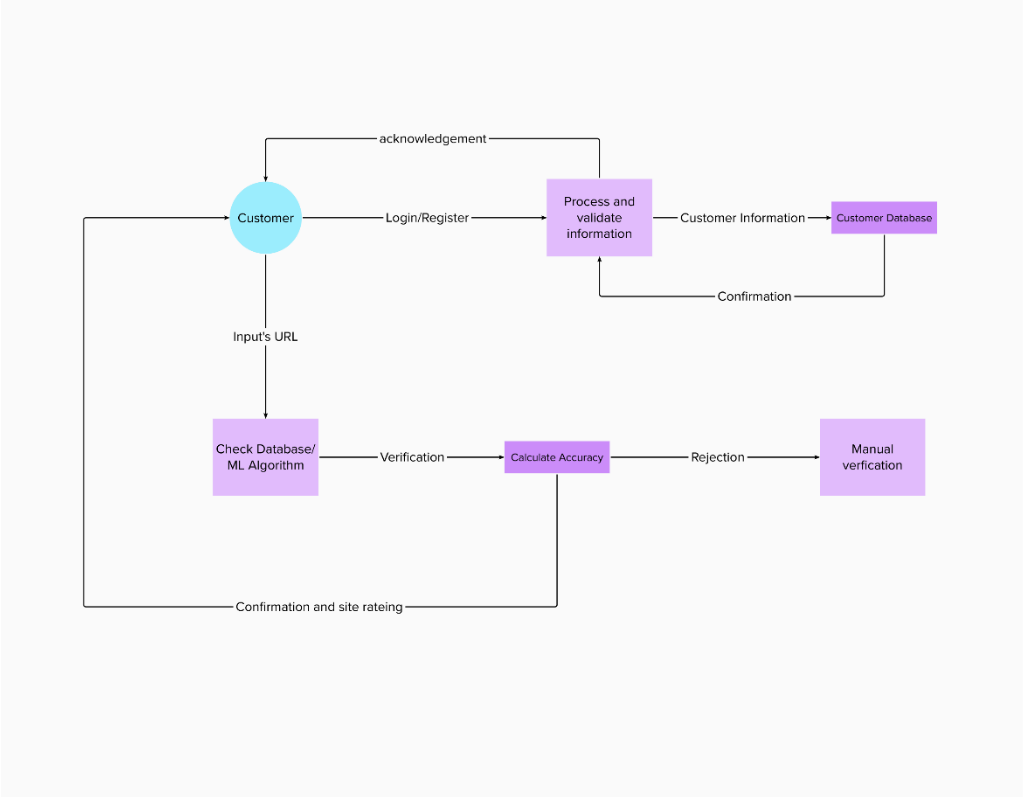
6.1 Non-Functional Requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user interface is clean, so that the user gets the expected result without any difficulties.
NFR-2	**Security**	The database is prevented from any tampering to provide a genuine result.
NFR-3	**Reliability**	If due to some injection attack or failure the backup updates are rolled back.
NFR-4	**Performance**	The result for the search will not take more than a minute to give out the result.
NFR-5	**Availability**	The server can handle required amount of response and are available even in the database updating process.
NFR-6	**Scalability**	The traffic limit and the accuracy will be increased to offer a better service.

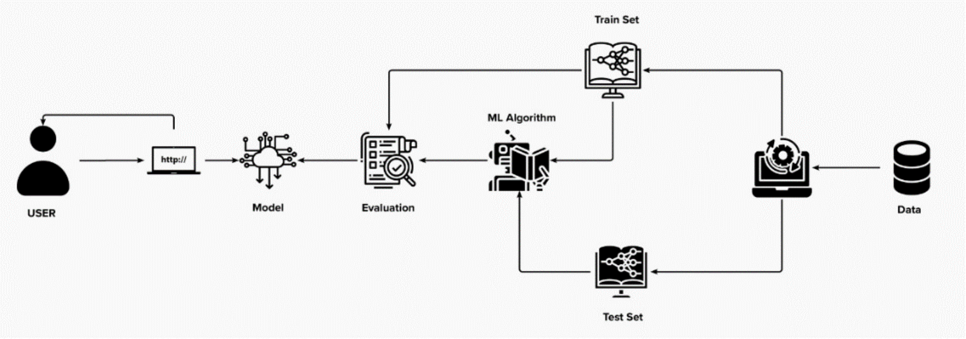
5. PROJECT DESIGN

5.1 Data Flow Diagram

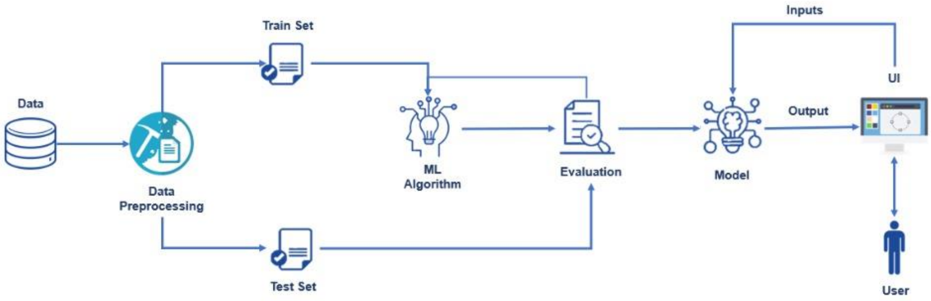


5.2 Solution & Technical Architecture

5.2.1 Solution Architecture



5.2.2 Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can access the website features	High	Sprint-2
	User Input	USN-5	As a user, I can input the URL in the required field and wait for validation	I can access the detailed result of the URL	High	Sprint-3
Administrator	Data Collection	USN-6	The data to identify the phishing link is to be collected	The model is ready to train	High	Sprint-3
	Data Pre-Processing	USN-7	The data is to be cleaned to provide better accuracy	The model is ready with high accuracy	High	Sprint-4

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Model Deployment	USN-8	The trained and tested model is deployed using the Machine learning algorithm	I have the model which is successfully deloyed	High	Sprint-5
	Application Building	USN-9	As a admin, The user page must be designed to access the feature in more ease manner	I have the live website	High	Sprint-5

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Collection of Dataset	USN-1	As a developer, I need to collect related data stored in a digital format to make machine learning models to understand.	3	High	Dharun Aditya Sharathsoorya Perumal Tamilarasan
Sprint-1	Data Pre-Processing	USN-2	As a developer, I need to prepare (cleaning and organizing) the raw data to make it suitable for building and training the model	5	High	Dharun Aditya Sharathsoorya Perumal Tamilarasan
Sprint-2	Exploratory Data Analysis (EDA)	USN-3	As a developer, EDA approach Is used to analyse the data to shortlist the	8	Medium	Dharun Aditya Sharathsoorya Perumal Tamilarasan

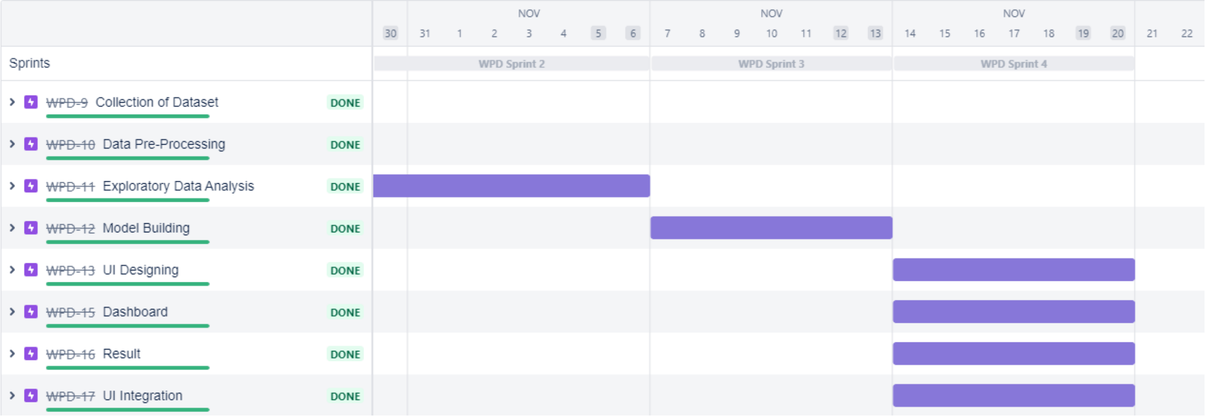
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			relevant columns required to train the model.			
Sprint-3	Model Building	USN-4	As a developer, I need to explore the data and type of algorithm, train and test it to provide better accuracy.	13	High	Dharun Aditya Sharathsoorya Perumal Tamilarasan
Sprint-4	UI Designing	USN-5	As a developer, I need to design an awesome UI to provide a better solution with less effort.	3	Medium	Dharun Aditya Sharathsoorya Perumal Tamilarasan
Sprint-4	UI Integration	USN-6	As a developer, I need to integrate UI page and the model to get user input and display the result in more user-friendly manner.	8	High	Dharun Aditya Sharathsoorya Perumal Tamilarasan
Sprint-4	Dashboard	USN-7	As a user, I can enter the suspicious URL to check the status of the link.	3	Medium	Dharun Aditya Sharathsoorya Perumal Tamilarasan
Sprint-4	Result	USN-8	As a user, I can receive whether the URL is safe or not.	5	High	Dharun Aditya Sharathsoorya Perumal Tamilarasan

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	8	6 Days	31 Oct 2022	05 Nov 2022	8	05 Nov 2022
Sprint-3	13	6 Days	07 Nov 2022	12 Nov 2022	13	13 Nov 2022
Sprint-4	19	6 Days	14 Nov 2022	19 Nov 2022	19	19 Nov 2022

6.3 Reports from JIRA

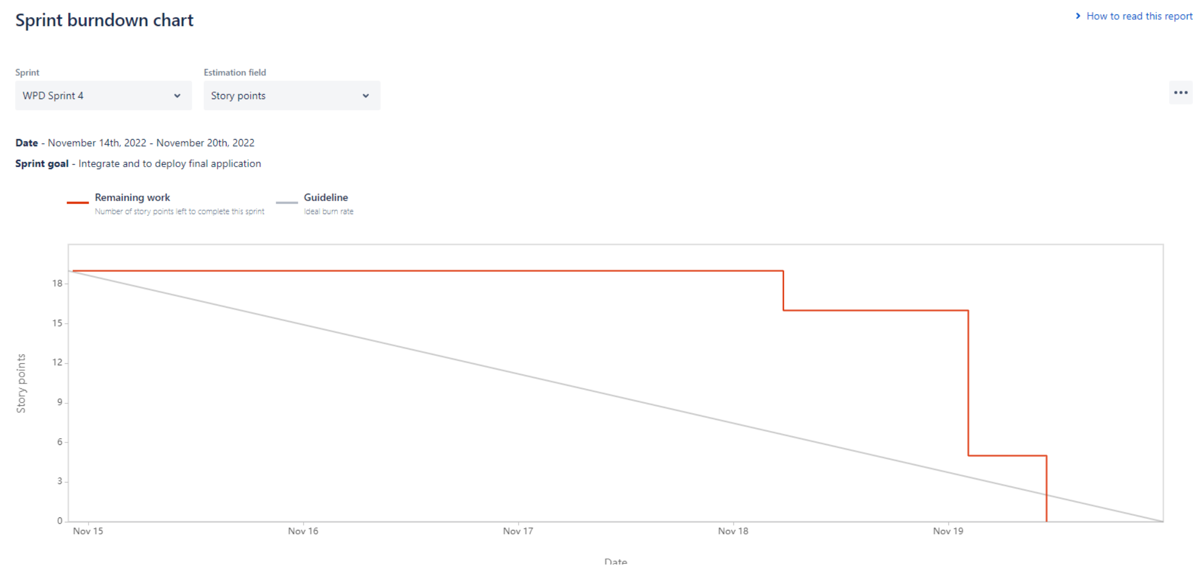
Roadmap:



Sprints

~~WPD-9~~	Collection of Dataset	DONE
~~WPD-10~~	Data Pre-Processing	DONE
~~WPD-11~~	Exploratory Data Analysis	DONE
~~WPD-12~~	Model Building	DONE
~~WPD-13~~	UI Designing	DONE
~~WPD-15~~	Dashboard	DONE
~~WPD-16~~	Result	DONE
~~WPD-17~~	UI Integration	DONE

Burndown Chart:



Sprint burndown chart

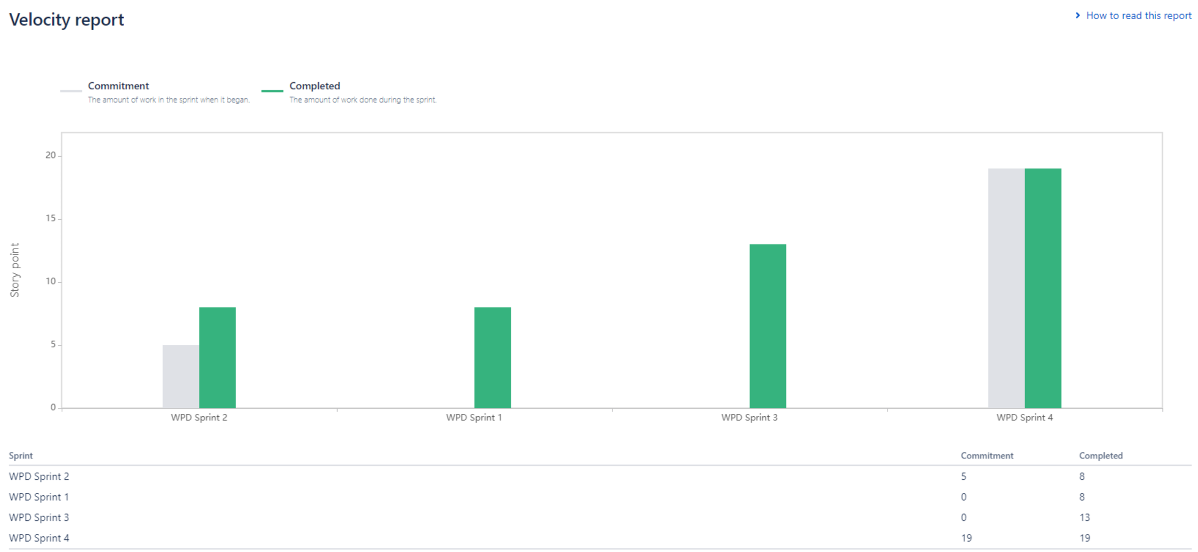
Sprint: WPD Sprint 4

Estimation field: Story points

Date - November 14th, 2022 - November 20th, 2022

Sprint goal - Integrate and to deploy final application

Velocity Chart:



Velocity report

› How to read this report

Commitment
The amount of work in the sprint when it began.

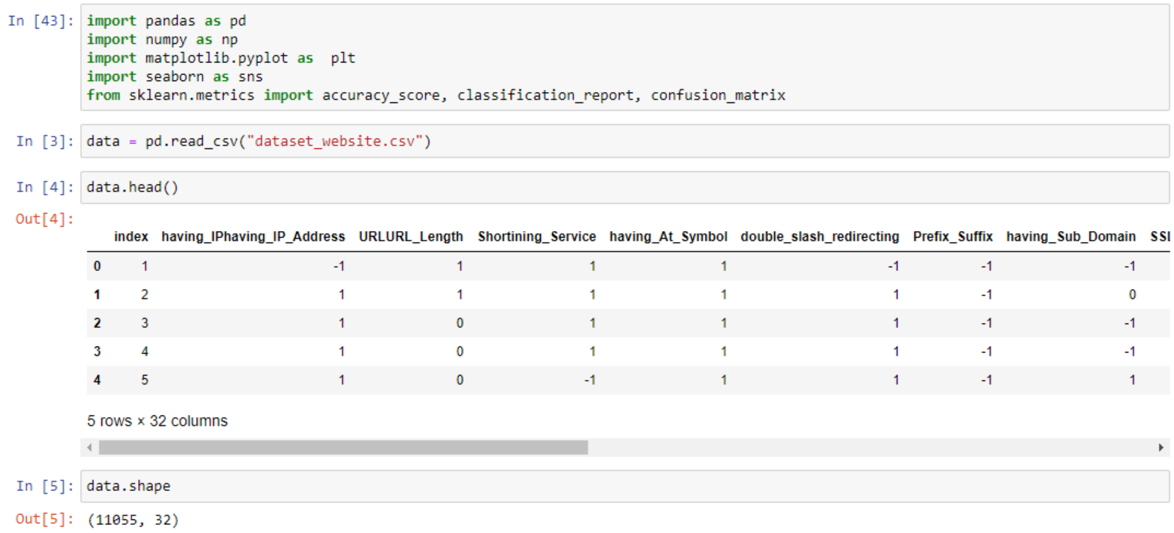
Completed
The amount of work done during the sprint.

Sprint	Commitment	Completed
WPD Sprint 2	5	8
WPD Sprint 1	0	8
WPD Sprint 3	0	13
WPD Sprint 4	19	19

7. CODING & SOLUTIONING

7.1 Feature 1

The Machine Learning model has been trained to detect the Phishing Website using Classification Algorithms with an accuracy of 95%.



```
In [43]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as  plt
         import seaborn as sns
         from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

In [3]: data = pd.read_csv("dataset_website.csv")

In [4]: data.head()
Out[4]:
```

	index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSl
0	1	-1	1	1	1	-1	-1	-1	
1	2	1	1	1	1	1	-1	0	
2	3	1	0	1	1	1	-1	-1	
3	4	1	0	1	1	1	-1	-1	
4	5	1	0	-1	1	1	-1	1	

5 rows × 32 columns

```
In [5]: data.shape
Out[5]: (11055, 32)
```

```
In [6]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11055 entries, 0 to 11054
Data columns (total 32 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   index                        11055 non-null  int64
 1   having_IPhaving_IP_Address   11055 non-null  int64
 2   URLURL_Length                11055 non-null  int64
 3   Shortining_Service           11055 non-null  int64
 4   having_At_Symbol             11055 non-null  int64
 5   double_slash_redirecting     11055 non-null  int64
 6   Prefix_Suffix                11055 non-null  int64
 7   having_Sub_Domain            11055 non-null  int64
 8   SSLfinal_State               11055 non-null  int64
 9   Domain_registeration_length  11055 non-null  int64
 10  Favicon                      11055 non-null  int64
 11  port                         11055 non-null  int64
 12  HTTPS_token                  11055 non-null  int64
 13  Request_URL                  11055 non-null  int64
 14  URL_of_Anchor                11055 non-null  int64
 15  Links_in_tags                11055 non-null  int64
 16  SFH                          11055 non-null  int64
 17  Submitting_to_email          11055 non-null  int64
 18  Abnormal_URL                 11055 non-null  int64
 19  Redirect                     11055 non-null  int64
 20  on_mouseover                 11055 non-null  int64
 21  RightClick                   11055 non-null  int64
 22  popUpWidnow                  11055 non-null  int64
 23  Iframe                       11055 non-null  int64
 24  age_of_domain                11055 non-null  int64
 25  DNSRecord                    11055 non-null  int64
 26  web_traffic                  11055 non-null  int64
 27  Page_Rank                    11055 non-null  int64
 28  Google_Index                 11055 non-null  int64
 29  Links_pointing_to_page       11055 non-null  int64
 30  Statistical_report           11055 non-null  int64
 31  Result                       11055 non-null  int64
dtypes: int64(32)
memory usage: 2.7 MB
```

UNIVARIATE ANALYSIS

```
In [8]: data['Result'].value_counts()
```

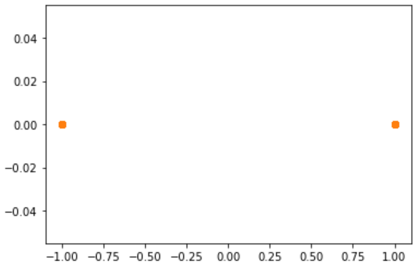
```
Out[8]:  1    6157
        -1    4898
        Name: Result, dtype: int64
```

```
In [9]: data_phish = data.loc[data['Result'] == -1]
        data_no_phish = data.loc[data['Result'] == 1]
```

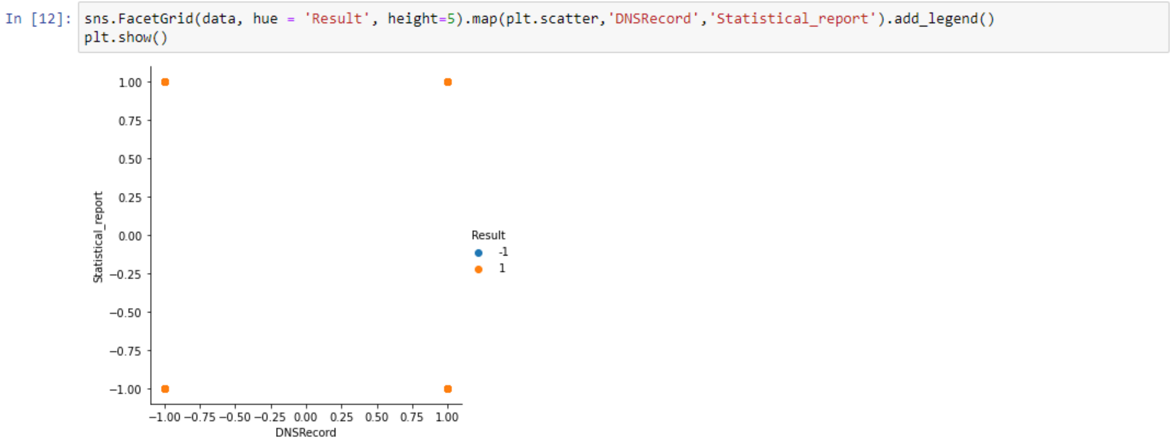
```
In [10]: data['DNSRecord'].value_counts()
```

```
Out[10]:  1    7612
         -1    3443
         Name: DNSRecord, dtype: int64
```

```
In [11]: plt.plot(data_phish['DNSRecord'], np.zeros_like(data_phish['DNSRecord']),'o')
         plt.plot(data_no_phish['DNSRecord'], np.zeros_like(data_no_phish['DNSRecord']),'o')
         plt.show()
```



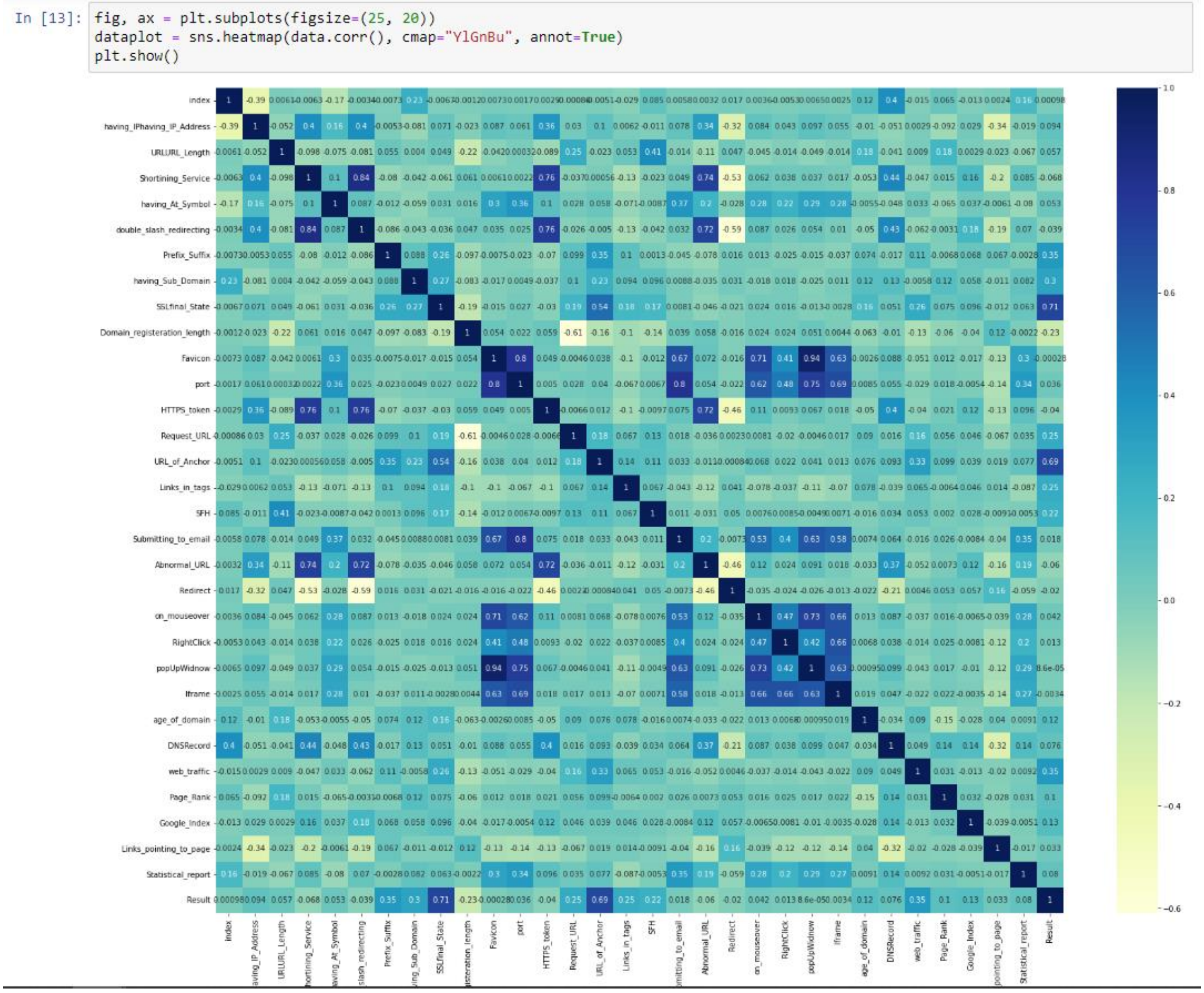
BIVARIATE ANALYSIS



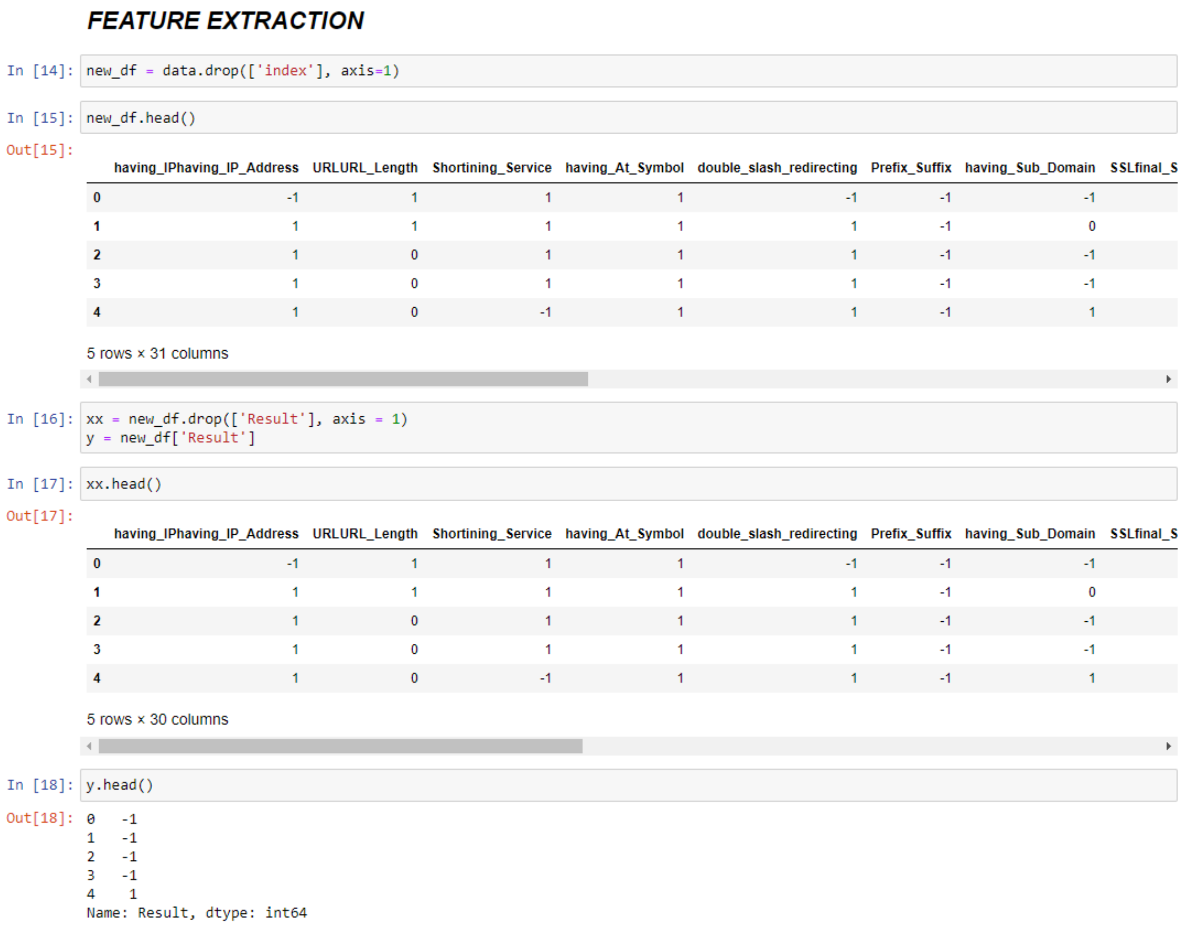
```
In [12]: sns.FacetGrid(data, hue = 'Result', height=5).map(plt.scatter,'DNSRecord','Statistical_report').add_legend()
         plt.show()
```

MULTIVARIATE ANALYSIS

```
In [ ]: sns.pairplot(data, hue='Result',size=3)

/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:2076: UserWarning: The `size` parameter has been renamed to `height
`; please update your code.
  warnings.warn(msg, UserWarning)

Out[12]: <seaborn.axisgrid.PairGrid at 0x7fe9847f1550>
```



```
In [13]: fig, ax = plt.subplots(figsize=(25, 20))
         dataplot = sns.heatmap(data.corr(), cmap="YlGnBu", annot=True)
         plt.show()
```



FEATURE EXTRACTION

In [14]:
```
new_df = data.drop(['index'], axis=1)
```

In [15]:
```
new_df.head()
```

Out[15]:

	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_S
0	-1	1	1	1	-1	-1	-1	
1	1	1	1	1	1	-1	0	
2	1	0	1	1	1	-1	-1	
3	1	0	1	1	1	-1	-1	
4	1	0	-1	1	1	-1	1	

5 rows × 31 columns

In [16]:
```
xx = new_df.drop(['Result'], axis = 1)
y = new_df['Result']
```

In [17]:
```
xx.head()
```

Out[17]:

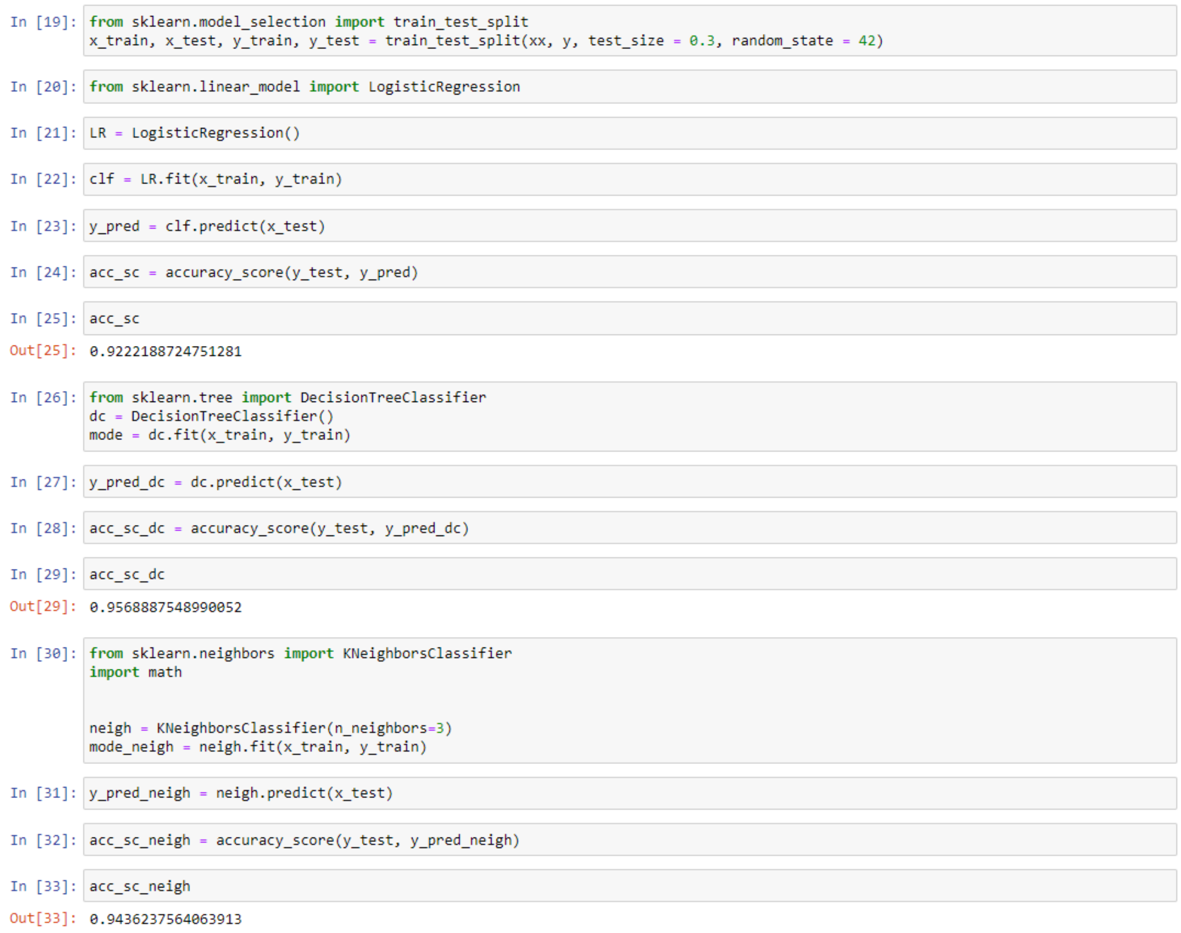
	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_S
0	-1	1	1	1	-1	-1	-1	
1	1	1	1	1	1	-1	0	
2	1	0	1	1	1	-1	-1	
3	1	0	1	1	1	-1	-1	
4	1	0	-1	1	1	-1	1	

5 rows × 30 columns

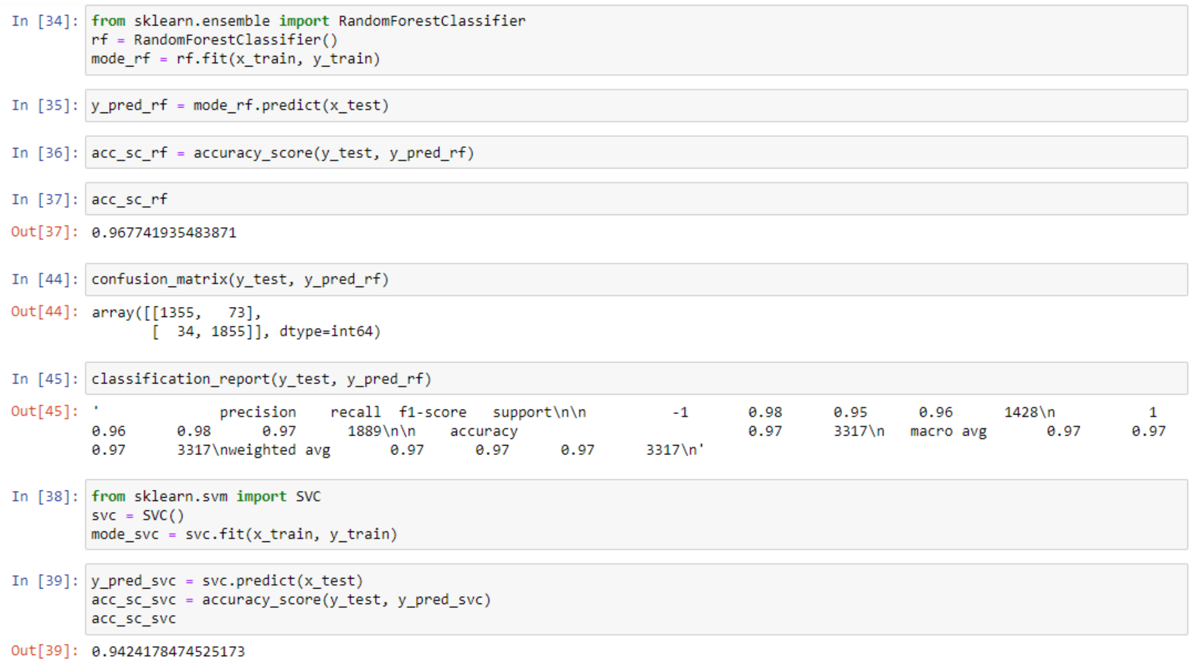
In [18]:
```
y.head()
```

Out[18]:
```
0   -1
1   -1
2   -1
3   -1
4    1
Name: Result, dtype: int64
```

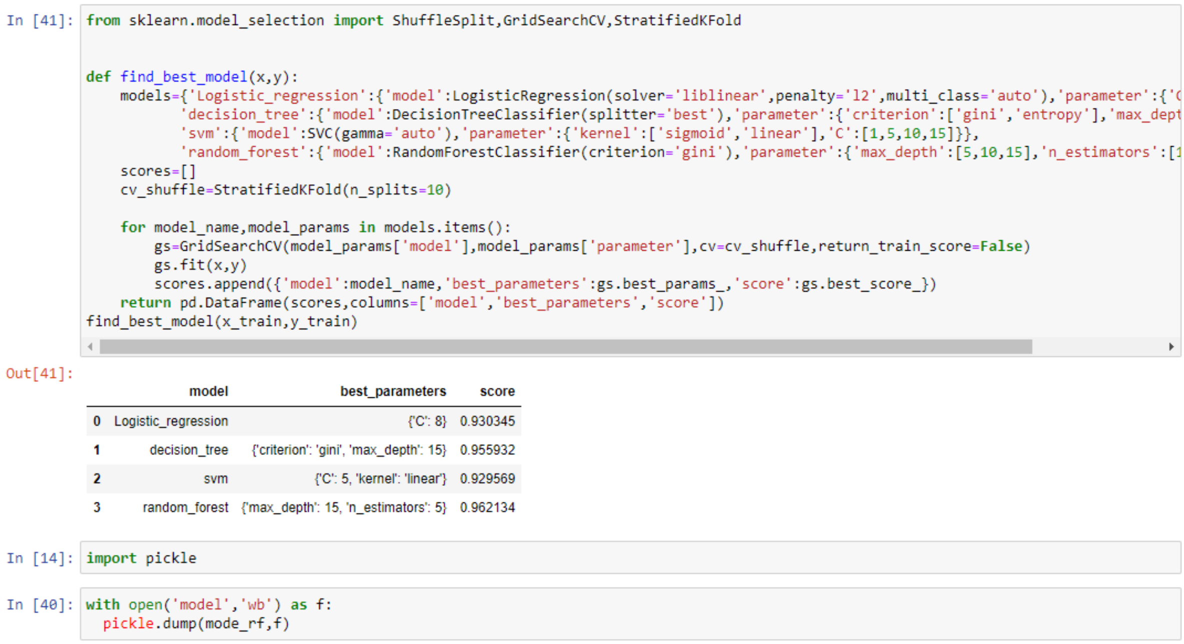
TRAINING, TESTING DATA WITH MODEL BUILDING



```
In [19]: from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(xx, y, test_size = 0.3, random_state = 42)

In [20]: from sklearn.linear_model import LogisticRegression

In [21]: LR = LogisticRegression()

In [22]: clf = LR.fit(x_train, y_train)

In [23]: y_pred = clf.predict(x_test)

In [24]: acc_sc = accuracy_score(y_test, y_pred)

In [25]: acc_sc

Out[25]: 0.9222188724751281

In [26]: from sklearn.tree import DecisionTreeClassifier
         dc = DecisionTreeClassifier()
         mode = dc.fit(x_train, y_train)

In [27]: y_pred_dc = dc.predict(x_test)

In [28]: acc_sc_dc = accuracy_score(y_test, y_pred_dc)

In [29]: acc_sc_dc

Out[29]: 0.9568887548990052

In [30]: from sklearn.neighbors import KNeighborsClassifier
         import math


         neigh = KNeighborsClassifier(n_neighbors=3)
         mode_neigh = neigh.fit(x_train, y_train)

In [31]: y_pred_neigh = neigh.predict(x_test)

In [32]: acc_sc_neigh = accuracy_score(y_test, y_pred_neigh)

In [33]: acc_sc_neigh

Out[33]: 0.9436237564063913
```



```
In [34]: from sklearn.ensemble import RandomForestClassifier
         rf = RandomForestClassifier()
         mode_rf = rf.fit(x_train, y_train)

In [35]: y_pred_rf = mode_rf.predict(x_test)

In [36]: acc_sc_rf = accuracy_score(y_test, y_pred_rf)

In [37]: acc_sc_rf

Out[37]: 0.967741935483871

In [44]: confusion_matrix(y_test, y_pred_rf)

Out[44]: array([[1355,   73],
                [  34, 1855]], dtype=int64)

In [45]: classification_report(y_test, y_pred_rf)

Out[45]: '              precision    recall  f1-score   support\n\n          -1       0.98      0.95      0.96      1428\n           1
         0.96      0.98      0.97      1889\n\n    accuracy                           0.97      3317\n   macro avg       0.97      0.97
         0.97      3317\nweighted avg       0.97      0.97      0.97      3317\n'

In [38]: from sklearn.svm import SVC
         svc = SVC()
         mode_svc = svc.fit(x_train, y_train)

In [39]: y_pred_svc = svc.predict(x_test)
         acc_sc_svc = accuracy_score(y_test, y_pred_svc)
         acc_sc_svc

Out[39]: 0.9424178474525173
```



In [41]:
```
from sklearn.model_selection import ShuffleSplit,GridSearchCV,StratifiedKFold


def find_best_model(x,y):
    models={'Logistic_regression':{'model':LogisticRegression(solver='liblinear',penalty='l2',multi_class='auto'),'parameter':{'C
            'decision_tree':{'model':DecisionTreeClassifier(splitter='best'),'parameter':{'criterion':['gini','entropy'],'max_dept
            'svm':{'model':SVC(gamma='auto'),'parameter':{'kernel':['sigmoid','linear'],'C':[1,5,10,15]}},
            'random_forest':{'model':RandomForestClassifier(criterion='gini'),'parameter':{'max_depth':[5,10,15],'n_estimators':[1
    scores=[]
    cv_shuffle=StratifiedKFold(n_splits=10)

    for model_name,model_params in models.items():
        gs=GridSearchCV(model_params['model'],model_params['parameter'],cv=cv_shuffle,return_train_score=False)
        gs.fit(x,y)
        scores.append({'model':model_name,'best_parameters':gs.best_params_,'score':gs.best_score_})
    return pd.DataFrame(scores,columns=['model','best_parameters','score'])
find_best_model(x_train,y_train)
```

Out[41]:

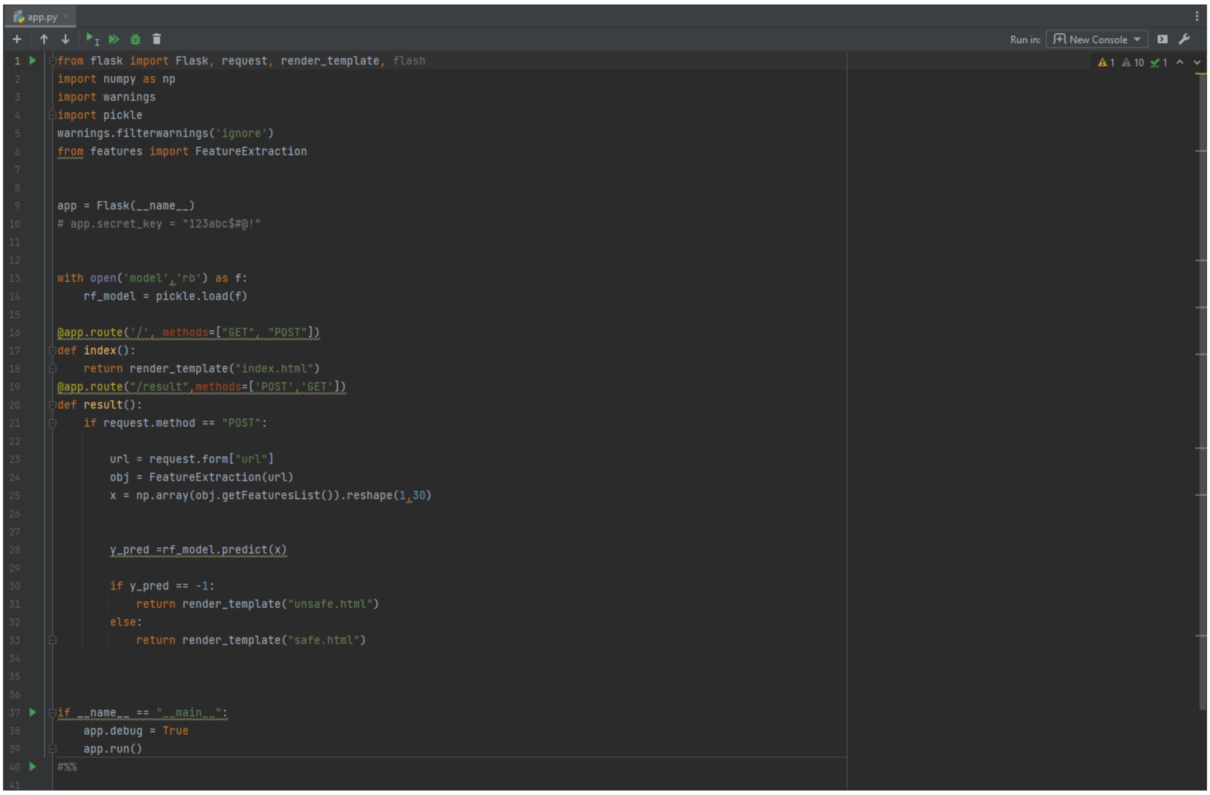
	model	best_parameters	score
0	Logistic_regression	{'C': 8}	0.930345
1	decision_tree	{'criterion': 'gini', 'max_depth': 15}	0.955932
2	svm	{'C': 5, 'kernel': 'linear'}	0.929569
3	random_forest	{'max_depth': 15, 'n_estimators': 5}	0.962134

In [14]:
```
import pickle
```

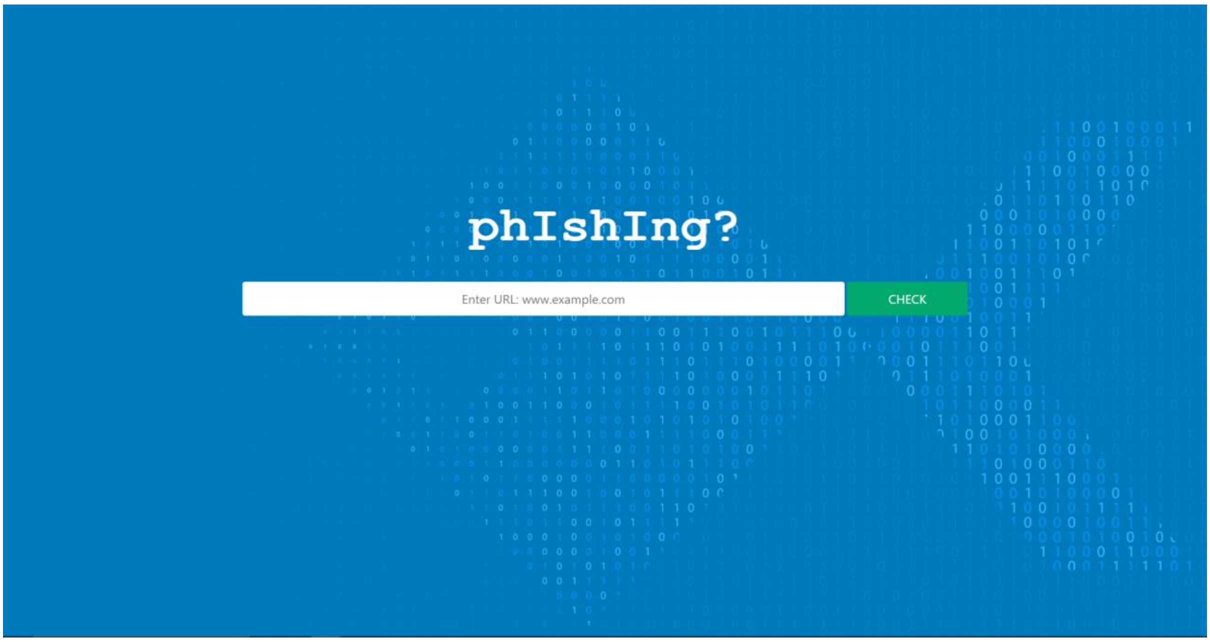
In [40]:
```
with open('model','wb') as f:
  pickle.dump(mode_rf,f)
```

7.2 Feature 2

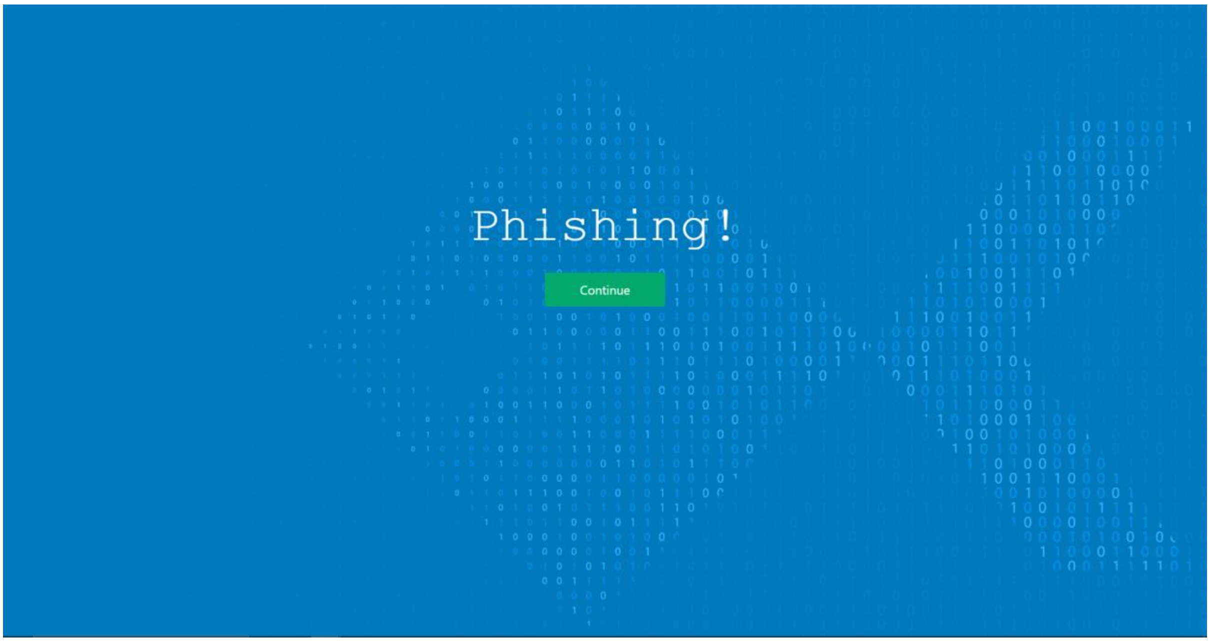
The Web Application contains an Input box where the suspicious link can be inputted to check the legitimacy.



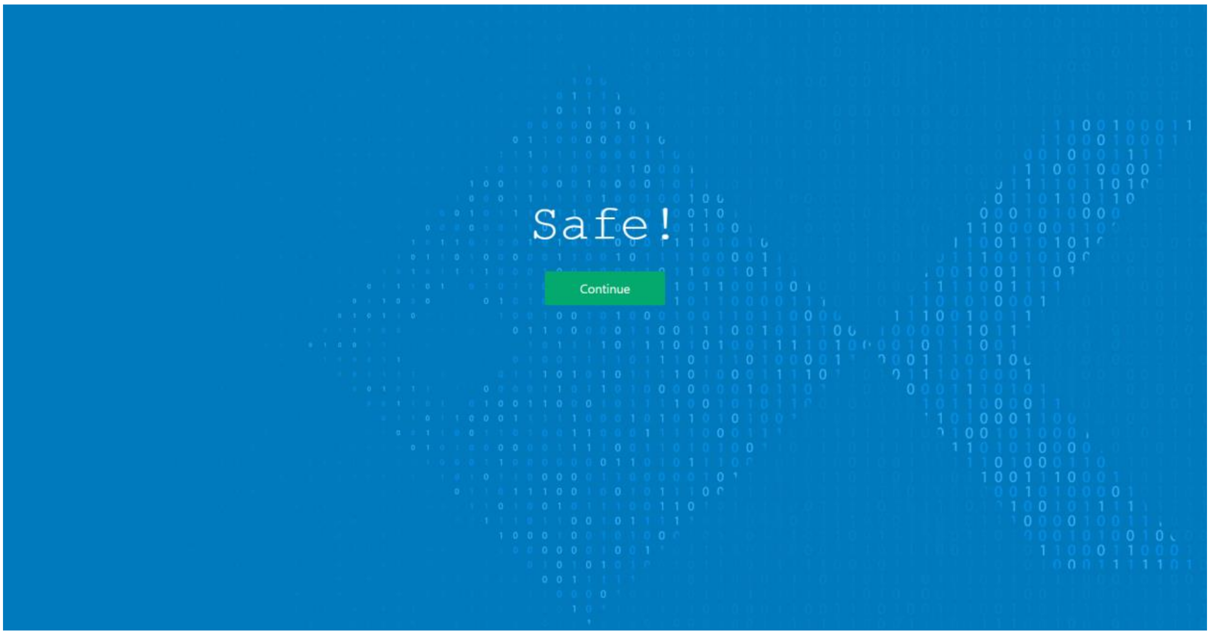
```
from flask import Flask, request, render_template, flash
import numpy as np
import warnings
import pickle
warnings.filterwarnings('ignore')
from features import FeatureExtraction


app = Flask(__name__)
# app.secret_key = "123abc$#@!"


with open('model','rb') as f:
    rf_model = pickle.load(f)

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

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


        y_pred =rf_model.predict(x)

        if y_pred == -1:
            return render_template("unsafe.html")
        else:
            return render_template("safe.html")



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



phIshIng?
Enter URL: www.example.com
CHECK



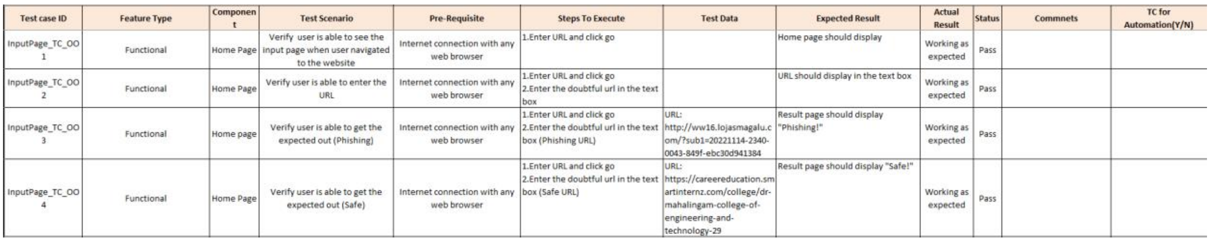
Phishing!
Continue



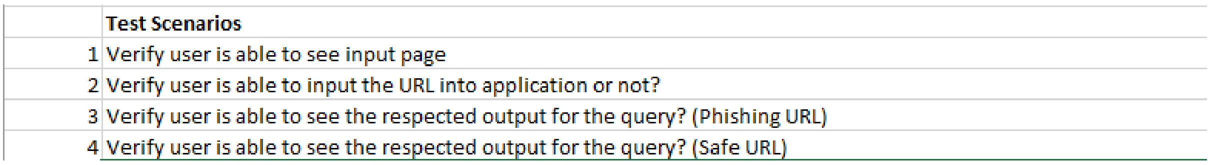
Safe!

8. TESTING

8.1 Test Cases

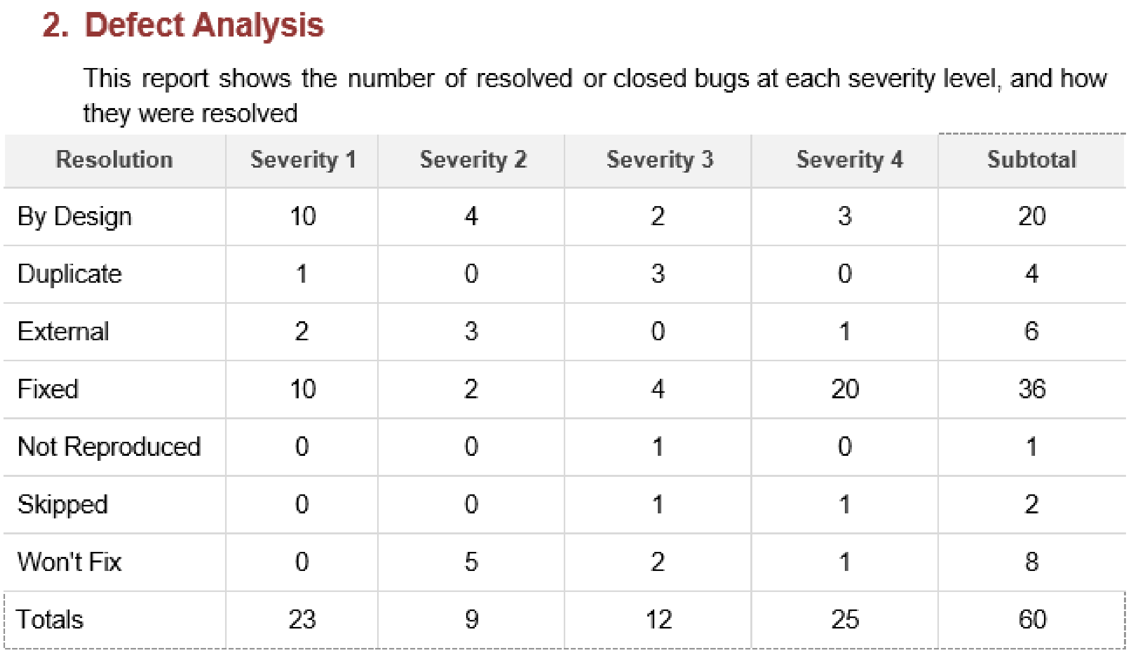


Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)
InputPage_TC_OO1	Functional	Home Page	Verify user is able to see the input page when user navigated to the website	Internet connection with any web browser	1.Enter URL and click go		Home page should display	Working as expected	Pass		
InputPage_TC_OO2	Functional	Home Page	Verify user is able to enter the URL	Internet connection with any web browser	1.Enter URL and click go 2.Enter the doubtful url in the text box		URL should display in the text box	Working as expected	Pass		
InputPage_TC_OO3	Functional	Home page	Verify user is able to get the expected out (Phishing)	Internet connection with any web browser	1.Enter URL and click go 2.Enter the doubtful url in the text box (Phishing URL)	URL: http://ww16.lojasmagalu.com/?sub1=20221114-2340-0043-849f-ebc30d941384	Result page should display "Phishing!"	Working as expected	Pass		
InputPage_TC_OO4	Functional	Home Page	Verify user is able to get the expected out (Safe)	Internet connection with any web browser	1.Enter URL and click go 2.Enter the doubtful url in the text box (Safe URL)	URL: https://careereducation.smartinternz.com/college/dr-mahalingam-college-of-engineering-and-technology-29	Result page should display "Safe!"	Working as expected	Pass		



	Test Scenarios
1	Verify user is able to see input page
2	Verify user is able to input the URL into application or not?
3	Verify user is able to see the respected output for the query? (Phishing URL)
4	Verify user is able to see the respected output for the query? (Safe URL)

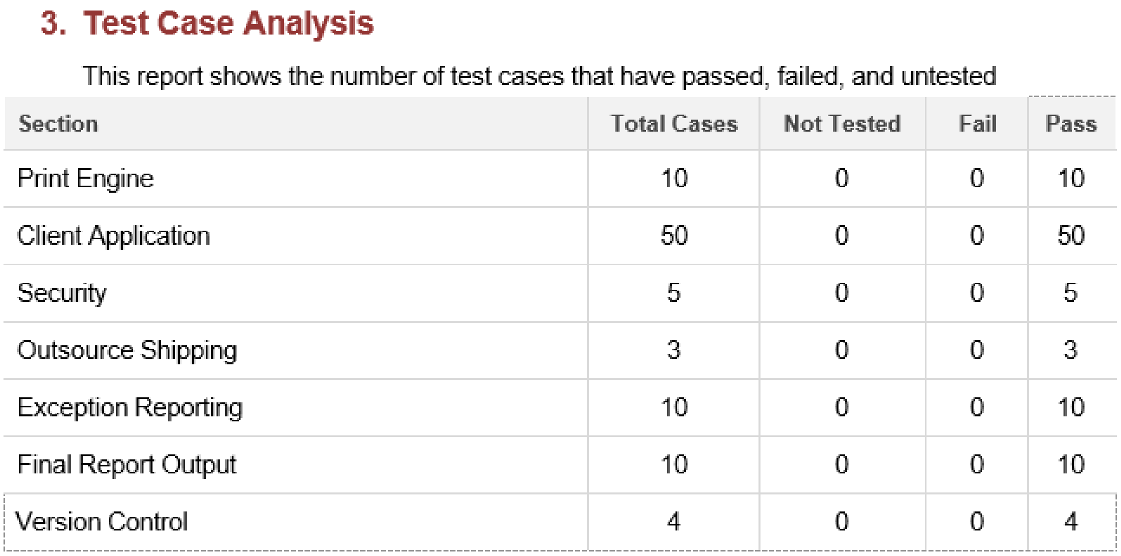
8.2 User Acceptance Testing



2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	10	2	4	20	36
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	23	9	12	25	60



3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

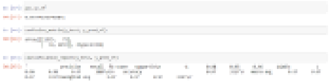
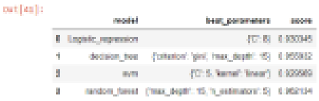
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	50	0	0	50
Security	5	0	0	5
Outsource Shipping	3	0	0	3
Exception Reporting	10	0	0	10
Final Report Output	10	0	0	10
Version Control	4	0	0	4

9. RESULTS

9.1 Performance Metrics

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	○ acc_sc_rf ○ confusion_matrix(y_test, y_pred_rf) ○ classification_report(y_test, y_pred_rf) 
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	 Out[43]: model | best_parameters | score 0 Logistic_regression | {'C': 8} | 0.930045 1 decision_tree | {'criterion': 'gini', 'max_depth': 15} | 0.955932 2 svm | {'C': 5, 'kernel': 'linear'} | 0.929589 3 random_forest | {'max_depth': 15, 'n_estimators': 5} | 0.952134

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

Phishing detection has a lot of advantages such as preventing identity theft, saving naïve internet users for being scammed of their money and bank details, increasing awareness about deceptive lucrative scamming websites on the internet among the public.

10.2 Disadvantages

There aren’t many disadvantages to being warned of a potential phishing website. One minor disadvantage could be wrongful categorization of an otherwise safe website as a phishing website due to some error on the part of the model.

11. CONCLUSION

In conclusion, a phishing detection website is the need of the hour right now and is a boon to the public to stay cybersafe on the internet. Phishing attacks are the most common cyber threat to many businesses. This form of cyber-attack can be remarkably unsophisticated. Yet, the disruption caused can be huge. Phishing emails prey on human behaviour. They will often claim to come from an authority figure. The suspicious email might foster a sense of urgency or offer reward to the recipient. In such an environment, websites such as these are an active effort to prevent such illicit activities.

12. FUTURE SCOPE

Although the use of URL lexical features alone has been shown to result in high accuracy (97%), phishers have learned how to make predicting a URL destination difficult by carefully manipulating the URL to evade detection. Therefore, combining these features with others, such as host, is the most effective approach.

For future enhancements, we intend to build the phishing detection system as a scalable extension and an anti-phish search engine which will incorporate online learning so that new phishing attack patterns can easily be learned and improve the accuracy of our models with better feature extraction

13. APPENDIX

App.py

```
from flask import Flask, request, render_template, flash
import numpy as np
import warnings
import pickle
warnings.filterwarnings('ignore')
from features import FeatureExtraction


app = Flask(__name__)
# app.secret_key = "123abc$#@!"


with open('model','rb') as f:
    rf_model = pickle.load(f)
```

```
@app.route('/', methods=["GET", "POST"])
def index():
    return render_template("index.html")
@app.route("/result",methods=['POST','GET'])
def result():
    if request.method == "POST":

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


        y_pred =rf_model.predict(x)

        if y_pred == -1:
            return render_template("unsafe.html")
        else:
            return render_template("safe.html")



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

Index.html

```
<!DOCTYPE html>
<html>
    <head>
        <script src="js/test.js"></script>
        <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
        <style>
            h1{
                text-align: center;
                margin-top: 16%;
                color: white;
                font-size: 500%;
                        font-family:courier;

            }
            #url{
                text-align: center;
                margin-top: 1%;
            }
           button
                            {
                                        background-color: transparent;
            border: 2px solid darkslategrey;
                              color:black;
                              font-size: 20px;
            cursor: pointer;
                     }
         button:hover{
            background-color: darkslategrey;
            color: white;
         }
         body{
            background-image:
url("https://www.phishingbox.com/themes/phishingbox/assets/img/branding/pbox_binary_ba
ckground_1920-1080.jpg");
            background-size: 100%;
         }
         input[type=text]{
            width: 50%;
            padding: 12px;
            border: none;
            border-radius: 4px;
            box-sizing: border-box;
            margin-top: 6px;
            margin-bottom: 16px;
            resize: vertical;
            font-size: 20px;
         }
         input[type=submit] {
            background-color: #04AA6D;
            color: white;
            width: 10%;
            padding: 12px;
            border: none;
            border-radius: 4px;
            box-sizing: border-box;
            margin-top: 6px;
            margin-bottom: 16px;
            resize: vertical;
            font-size: 20px;
            font-style:inherit;
            font-family:
         }
         input[type=submit]:hover {
            background-color: #45a049;
         }
         #log{


         }

      </style>
```

```
    </head>
    <body>
            <h1><strong>phIshIng?</strong></h1>
            <div id="url">
            <form action="{{url_for('result')}}" method="POST">
                <input type="text" id="url" name="url" placeholder="Enter URL: 
www.example.com"/>
                <input type="submit" value="CHECK"/>
            </form>
            </div>
    </body>
</html>
```

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-5296-1658756183

Project Demo Link:

https://drive.google.com/file/d/1rkBsKjbLJOLLMrXBRuK_Zq5CBgT503Hf/view?usp=sharing