

**HX8001 - PROFESSIONAL READINESS FOR
INNOVATION, EMPLOYABILITY AND
ENTREPRENEURSHIP**

**DETECTION OF PHISHING WEBSITES FROM
URL'S USING IBM WATSON STUDIO**

TEAM ID: PNT2022TMID50233

FACULTY MENTOR NAME: MRS. W. VINOTHINI MARY

TEAM LEADER: T. TEJASWINI

TEAM MEMBERS: J. BLESSY SHARON ROSE

R. SOWMIYA

C. VALARMATHI

P. SUBBULAKSHMI

TABLE OF CONTENTS

SI.NO	TOPIC	PAGE.NO
1	1. ABSTRACT	3
2	2. INTRODUCTION	4
	2.1 Project Overview	4
	2.2 Purpose	4
3	3.LITERATURE SURVEY	5
	3.1 Existing problem	5
	3.2ProblemStatement Definition	5
4	4.IDEATION & PROPOSED SOLUTION	7
	4.1 Empathy Map Canvas	7
	4.2 Ideation & Brainstorming	9
	4.3 proposed solution	14
	4.4 Proposed Solution fit	17
5	5.REQUIREMENT ANALYSIS	19
	5.1 block diagram	19
	5.2hardware/software designing	20
6	6.PROJECT DESIGN	21
	6.1 Data Flow Diagram	21
	6.2.Solution&Technical Architecture	23
7	7.Experimental Investigation	24
8	8.Flow chart	25

9	9.Result	24
10	10.Advantage& Disadvantages	25
11	11.Applications	25
12	12.conclusion	25
13	13.Future code	26
14	14.Appendix	26
	14.1.Source code	25
	14.2.Github & Project Demo link	41
	14.3 Reference Link	41

1. ABSTRACT

There are a number of users who purchase products online and make payments through e-banking. There are e-banking websites that ask users to provide sensitive data such as username, password & credit card details, etc., often for malicious reasons. This type of e-banking website is known as a phishing website. Web service is one of the key communications software services for the Internet. Web phishing is one of many security threats to web services on the Internet.

Common threats of web phishing:

- Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity.
- It will lead to information disclosure and property damage.
- Large organizations may get trapped in different kinds of scams.

This Guided Project mainly focuses on applying a machine-learning algorithm to detect Phishing websites.

In order to detect and predict e-banking phishing websites, we proposed an intelligent, flexible and effective system that is based on using classification algorithms. We implemented classification algorithms and techniques to extract the phishing datasets criteria to classify their legitimacy. The e-banking 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. Once a user makes a transaction online when he makes payment through an e-banking website our system will use a data mining algorithm to detect whether the e-banking website is a phishing website or not.

2. INTRODUCTION

2.1 OVERVIEW:

Phishing attack is a simplest way to obtain sensitive information from innocent users. Aim of the phishers is to acquire critical information like username, password and bank account details. Cyber security persons are now looking for trustworthy and steady detection techniques for phishing websites detection. This paper deals with machine learning technology for detection of phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. Logistic Regression, Decision Tree, random forest and Support vector machine algorithms are used to detect phishing websites. Aim of the paper is to detect phishing URLs as well as narrow down to best machine learning algorithm by comparing accuracy rate, false positive and false negative rate of each algorithm.

2.2 PURPOSE:

Nowadays Phishing becomes a main area of concern for security researchers because it is not difficult to create the fake website which looks so close to legitimate website. Experts can identify fake websites but not all the users can identify the fake website and such users become the victim of phishing attack. Main aim of the attacker is to steal banks account credentials. In United States businesses, there is a loss of US\$2billion per year because their clients become victim to phishing [1]. In 3rd Microsoft Computing Safer Index Report released in February 2014, it was estimated that the annual worldwide impact of phishing could be as high as \$5 billion [2]. Phishing attacks are becoming successful because lack of user awareness. Since phishing attack exploits the weaknesses found in users, it is very difficult to mitigate them but it is very important to enhance phishing detection techniques. The general method to detect phishing websites by updating blacklisted URLs, Internet Protocol (IP) to the antivirus database which is also known as "blacklist" method. To evade blacklists attackers uses creative techniques to fool users by modifying the URL to appear legitimate via obfuscation and many other simple

techniques including: fast-flux, in which proxies are automatically generated to host the web-page; algorithmic generation of new URLs; etc. Heuristic based detection which includes characteristics that are found to exist in phishing attacks in reality and can detect zero-hour phishing attack, but the characteristics are not guaranteed to always exist in such attacks and false positive rate in detection is very high [3]. To overcome the drawbacks of blacklist and heuristics based method, many security researchers now focused on machine learning techniques. Machine learning technology consists of a many algorithms which requires past data to make a decision or prediction on future data. Using this technique , algorithm will analyze various black listed and legitimateURLs and their features to accurately detect the phishing websites.

3. LITERATURE SURVEY

3.1 EXISTING PROBLEM

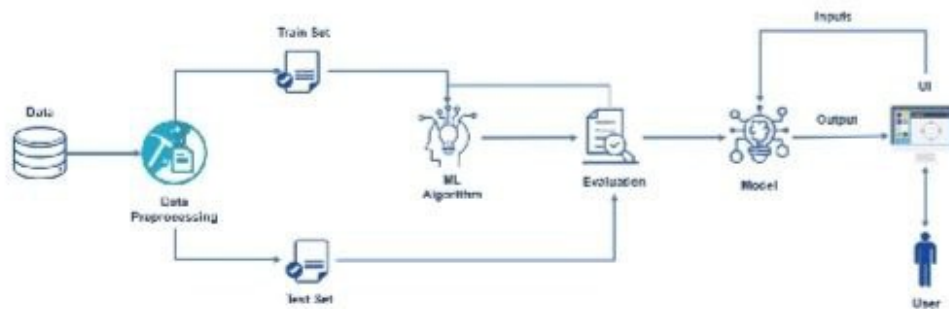
The current problem is Phishing. These social engineering attacks are designed to **fool you** into causing a data breach. Phishing attackers pose as people or organizations you trust to easily deceive you. Criminals of this nature try to coax you into handing over access to sensitive data or provide the data itself.

3.2 PROBLEM STATEMENT DEFINITION

PROBLEM STATEMENT

The problem statement aims at detecting phishing websites based on some important characteristics like URL and domain identity, and security and encryption criteria in the final phishing detection rate to classify their legitimacy with help of data mining algorithm once a user makes a request to a phishing website.

Model for web phishing detection



Question	Description
Who does the problem affect?	The users who purchase products online and make payments through e-banking and large organisation employees at higher levels.
Why is it important to use?	Using web phishing detection for handling phishing websites, dramatically reduces the chances of data theft and other cyber frauds.
What are the benefits?	To detect and predict phishing websites.
How is it better than others?	Provides results with better accuracy than other phishing detection techniques.
When to use?	The scenario where want to provide confidential credentials these web phishing detection could be used.

4. IDEATION AND PROPOSED SOLUTION

4.1 EMPATHY MAP CANVAS

Ideation Phase

Empathize & Discover

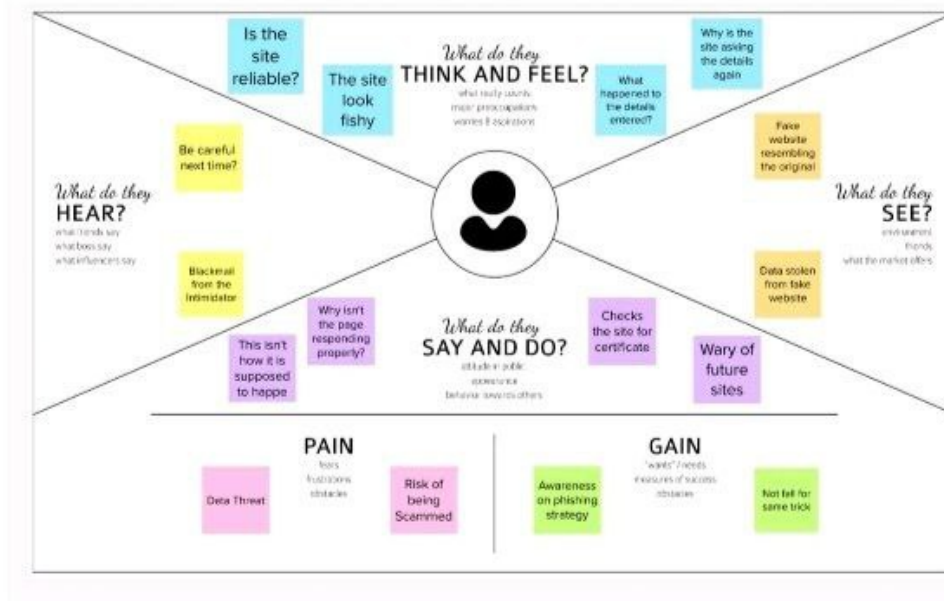
Date	24 September 2022
Team ID	PNT 2022TMID50233
Project Name	Web Phishing Detection
Maximum Marks	4 marks

Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to help teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Empathy map on Web Phishing Detection System:



4.2 IDEATION AND BRAINSTORMING

Ideation Phase

Brainstorm & Idea Prioritization Template

Date	12 october 2022
Team ID	PNT2022TMID50233
Project Name	Web Phishing Detection
Maximum Marks	4 Marks


Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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.

Reference: <https://www.mural.co/templates/empathy-map-canvas>

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



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
👥 2-6 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 on 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

PROBLEM

How might we (your problem statement)?

Key rules of brainstorming

To run an smooth and productive session

⌚ Stay in topic.

⌚ Defer judgment.

⌚ Go for volume.

💡 Encourage wild ideas.

👂 Listen to others.

👁️ If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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

**Phishing detection
using random forest
algorithm**

Identify phishing URL using features.
A classification is made by passing
each input vector down each tree,
randomly.
Each tree gives a classification, or vote,
and the forest chooses the
classification with the most instances,
or votes .
It runs efficiently on large datasets and
it can handle missing values.

**Phishing detection
using decision tree**

Identify the criteria that can
recognize fake URLs.
Build a decision tree that can iterate
through the criteria.
Train our model to recognize fake
vs real URLs.
Evaluate our model to see how it
performs.
Check for false positives/negatives.

TIP

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

**Phishing detection
using databases**

Check if input URL is in
Phishing website list.
If not in phishing website list
, check if keyword is in
phishing website list.
Check if phishing websites
features match.
Check if domain is
registered in WHOIS
database

2

Brainstorm

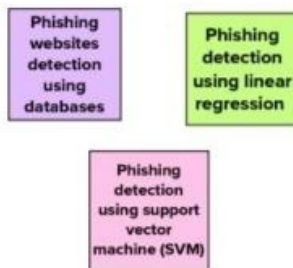
Write down any ideas that come to mind that address your problem statement.

⌚ 10 minutes

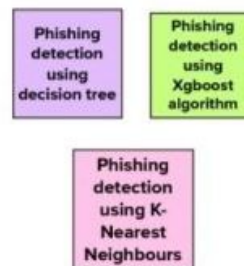
TIP

You can select a sticky note and hit the pencil icon (switch to sketch) or the eraser icon to start drawing

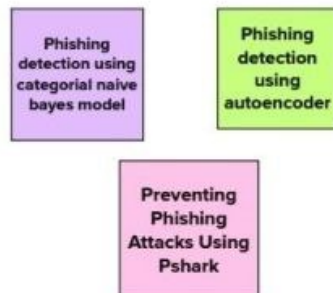
Person 1



Person 2



Person 3



Person 4



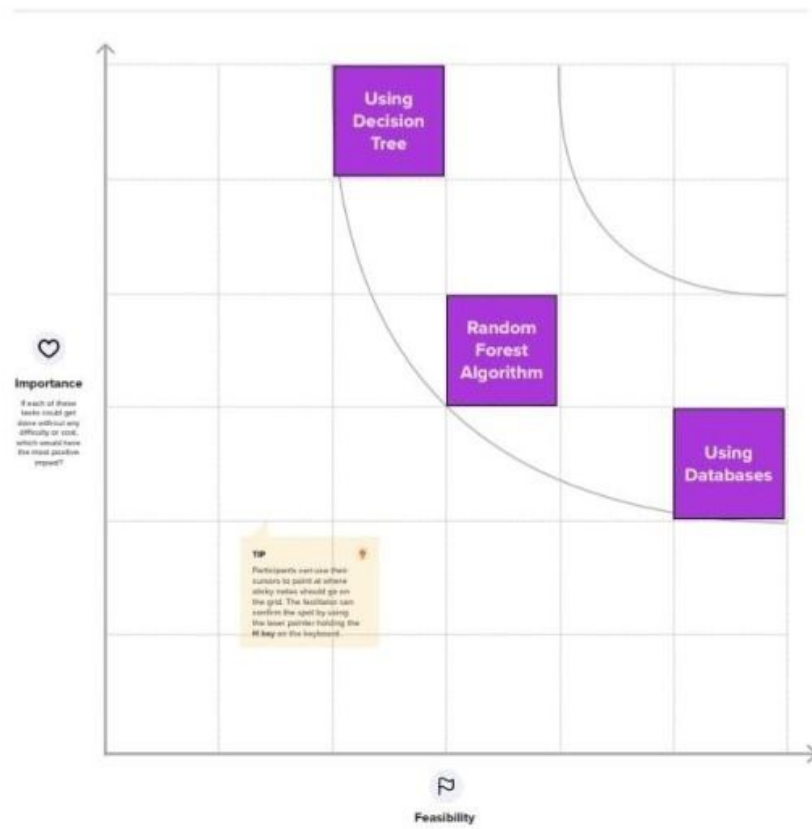
Step-3: Idea Prioritization



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



4.3 PROPOSED SOLUTION

We have implemented python program to extract features from URL. Below are the features that we have extracted for detection of phishing URLs.

- **Presence of IP address in URL:** If IP address present in URL then the feature is set to 1 else set to 0. Most of the benign sites do not use IP address as an URL to download a webpage. Use of IP address in URL indicates that attacker is trying to steal sensitive information.
- **Presence of @ symbol in URL:** If @ symbol present in URL then the feature is set to 1 else set to 0. Phishers add special symbol @ in the URL leads the browser to ignore everything preceding the “@” symbol and the real address often follows the “@” symbol.
- **Number of dots in Hostname:** Phishing URLs have many dots in URL. For example <http://shop.fun.amazon.phishing.com>, in this URL phishing.com is an actual domain name, whereas use of “amazon” word is to trick users to click on it. Average number of dots in benign URLs is 3. If the number of dots in URLs is more than 3 then the feature is set to 1 else to 0.
- **Prefix or Suffix separated by (-) to domain:** If domain name separated by dash (-) symbol then feature is set to 1 else to 0. The dash symbol is rarely used in legitimate URLs. Phishers add dash symbol (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example Actual site is <http://www.onlineamazon.com> but phisher can create another fake website like <http://www.online-amazon.com> to confuse the innocent users.
- **URL redirection:** If “//” present in URL path then feature is set to 1 else to 0. The existence of “//” within the URL path means that the user will be redirected to another website.
- **HTTPS token in URL:** If HTTPS token present in URL then the feature is set to 1 else

to 0. Phishers may add the “HTTPS” token to the domain part of a URL in order to trick users. For example, <http://https-wwwpaypal-it-mpp-home.soft-hair.com>.

- **Length of Host name:** Average length of the benign URLs is found to be a 25, If URL's length is greater than 25 then the feature is set to 1 else to 0.
- **Presence of sensitive words in URL:** Phishing sites use sensitive words in its URL so that users feel that they are dealing with a legitimate webpage. Below are the words that found in many phishing URLs :- 'confirm', 'account', 'banking', 'secure', 'ebyisapi', 'webscr', 'signin', 'mail', 'install', 'toolbar', 'backup', 'paypal', 'password', 'username', etc;
- **Number of slash in URL:** The number of slashes in benign URLs is found to be a 5; if number of slashes in URL is greater than 5 then the feature is set to 1 else to 0.
- **Presence of Unicode in URL:** Phishers can make a use of Unicode characters in URL to trick users to click on it. For example the domain “xn--80ak6aa92e.com” is equivalent to "apple.com".

Visible URL to user is "apple.com" but after clicking on this URL, user will visit to “xn--80ak6aa92e.com” which is a phishing site.

- **Website Rank:** We extracted the rank of websites and compare it with the first One hundred thousand websites of Alexa database. If rank of the website is greater than 10,0000 then feature is set to 1 else to 0.

Project Design Phase-I
Proposed Solution Template

Date	12 October 2022
Team ID	PNT2022TMID50233
Project Name	Project – Web Phishing Detection
Maximum Marks	2 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none"> Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity. It will lead to information disclosure and property damage.
2.	Idea / Solution description	<ul style="list-style-type: none"> A deep learning-based framework by implementing it as a browser plug-in capable of determining whether there is a phishing risk in real-time when the user visits a web page and gives a warning message. The real-time prediction includes whitelist filtering, blacklist interception, and machine learning (ML) prediction.
3.	Novelty / Uniqueness	<ul style="list-style-type: none"> To deal with phishing attacks and distinguishing the phishing webpages automatically, Blacklist based detection technique keeps a list of websites' URLs that are categorized as phishing sites. If a web-page requested by a user exists in the formed list, the connection to the queried website is blocked. Machine Learning (ML) based approaches rely on classification algorithms such as Support Vector Machines (SVM) and Decision Trees (DT) to train a model that can later automatically classify the fraudulent websites at run-time without any human intervention.
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> Large organizations may get trapped in different kinds of scams. There are a number of users who purchase products online and make payments through e-banking. There are e-banking phishing websites that ask

		users to provide sensitive data such as username, password & credit card details, etc., often for malicious reasons.
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> The browser plugin can be provided with a subscription plan or could be sold as a licensed software.
6.	Scalability of the Solution	<ul style="list-style-type: none"> To create microservices with flask web framework so that the model could scaled vertically or horizontally and effective traffic management.

4.4 PROBLEM SOLUTION FIT

Project Title: Web Phishing Detection

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID50233

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

An internet user who is willing to shop products online.

An enterprise user surfing through the internet for some information.

CS

6. CUSTOMER CONSTRAINTS

Customers have very little awareness on phishing websites.

They don't know what to do after losing data.

5. AVAILABLE SOLUTIONS

Which solutions are available

The already available solutions are blocking such phishing sites and by triggering a message to the customer about dangerous nature of the website.

But the blocking of phishing sites are not more effective as the attackers use a different new site to steal potential data thus a AI/ML model can be used to prevent customers from these kinds of sites from stealing data

AS

Explore AS, differentiate

Identify	<p>2. JOBS-TO-BE-DONE / PROBLEMS</p> <p>The phishing websites must be detected in a earlier stage . The user can be blocked from entering such sites for the prevention of such issues.</p>	<p>9. PROBLEM ROOT CAUSE</p> <p>The hackers use new ways to cheat the naive users.</p> <p>Very limited research is performed on this part of the internet.</p>	<p>7. BEHAVIOUR</p> <p>The option to check the legitimacy of the Websites is provided.</p> <p>Users get an idea what to do and more importantly what not to do.</p>	Identify
	<p>3. TRIGGERS</p> <p>A trigger message can be popped warning the user about the site.</p> <p>Phishing sites can be blocked by the ISP and can show a "site is blocked" or "phishing site detected" message.</p>	<p>10. YOUR SOLUTION</p> <p>An option for the users to check the legitimacy of the websites is provided.</p> <p>This increases the awareness among users and prevents misuse of data, data theft etc.,</p>	<p>8. CHANNELS of BEHAVIOUR</p> <p>1.1 ONLINE Customers tend to lose their data to phishing sites.</p> <p>1.2 OFFLINE Customers try to learn about the ways they get cheated from various resources viz., books, other people etc.,</p>	

Identify	<p>2. JOBS-TO-BE-DONE / PROBLEMS</p> <p>The phishing websites must be detected in a earlier stage . The user can be blocked from entering such sites for the prevention of such issues.</p>	<p>9. PROBLEM ROOT CAUSE</p> <p>The hackers use new ways to cheat the naive users.</p> <p>Very limited research is performed on this part of the internet.</p>	<p>7. BEHAVIOUR</p> <p>The option to check the legitimacy of the Websites is provided.</p> <p>Users get an idea what to do and more importantly what not to do.</p>	Identify
	<p>3. TRIGGERS</p> <p>A trigger message can be popped warning the user about the site.</p> <p>Phishing sites can be blocked by the ISP and can show a "site is blocked" or "phishing site detected" message.</p>	<p>10. YOUR SOLUTION</p> <p>An option for the users to check the legitimacy of the websites is provided.</p> <p>This increases the awareness among users and prevents misuse of data, data theft etc.,</p>	<p>8. CHANNELS of BEHAVIOUR</p> <p>1.1 ONLINE Customers tend to lose their data to phishing sites.</p> <p>1.2 OFFLINE Customers try to learn about the ways they get cheated from various resources viz., books, other people etc.,</p>	

5. REQUIREMENT ANALYSIS

5.1 BLOCK DIAGRAM

One machine learning model, Logistic Regression has been selected to detect phishing websites.

Logistic Regression: Logistic Regression is a Machine Learning algorithm which is used for the **classification problems**, it is a predictive analysis algorithm and based on the concept of probability. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Logistic regression is a simple yet very effective classification algorithm so it is commonly used for many **binary classification tasks**. Customer churn, spam email, website or ad click predictions are some examples of the areas where logistic regression offers a powerful solution.

Logistic regression uses an **equation as the representation**, very much like linear regression. Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y).

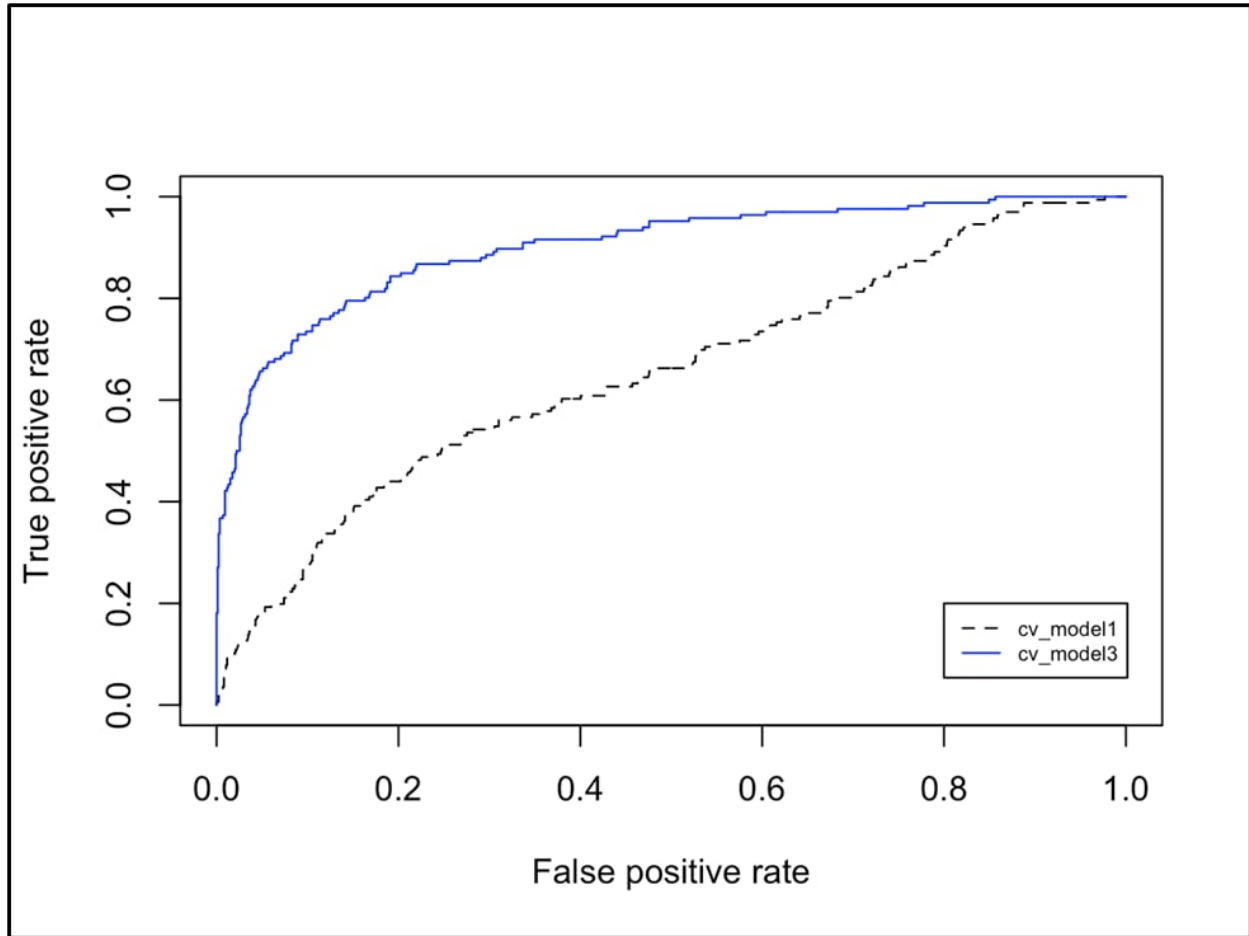


FIG: Logistic regression model comparison.

5.2 HARDWARE/ SOFTWARE DESIGNING

Hardware Requirements:

- Processor Minimum: Minimum 1GHz; Recommended 2GHz or more
- Ethernet Connection(LAN) or a Wireless adapter (WiFi)
- Hard Drive: Minimum 32GB ; Recommended 64 GB or more
- Memory (RAM): Minimum 1GB; Recommended 4GB or above

Software Requirements:

- Python(3.7 or older)
- Anaconda Prompt
- Watson Studio Service
- Spyder 3.8
- Flask
- Jupiter Notebook

6. PROJECT DESIGN

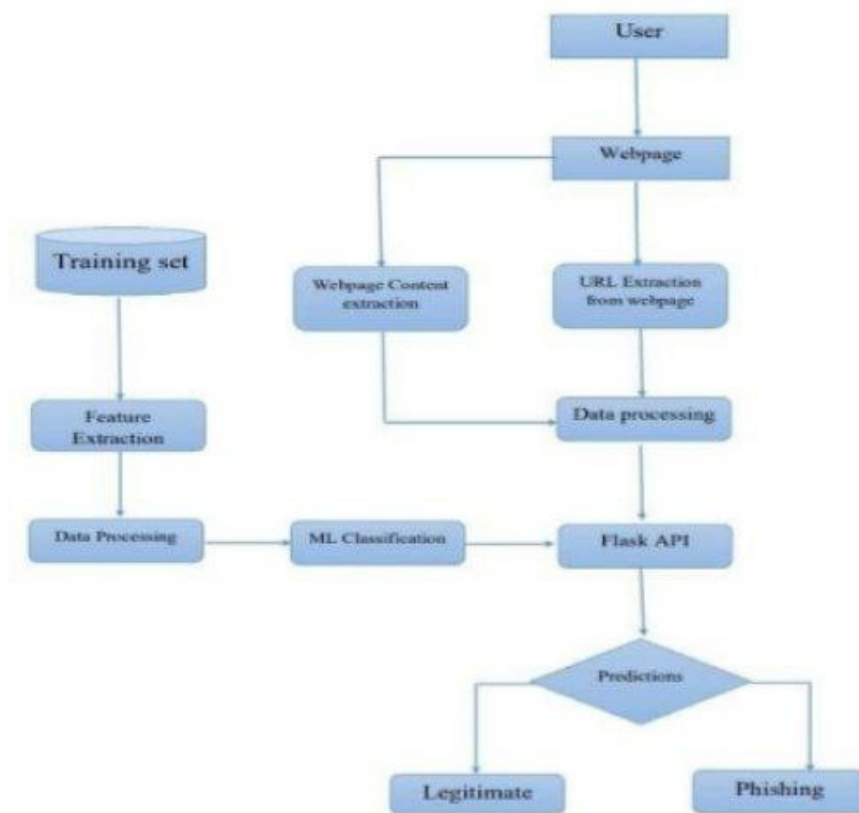
6.1 DATA FLOW DIAGRAM

Project Design Phase-II Data Flow Diagram & User Stories

Date	10 October 2022
Project Name	Web phishing Detection
Maximum Marks	4 Marks

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

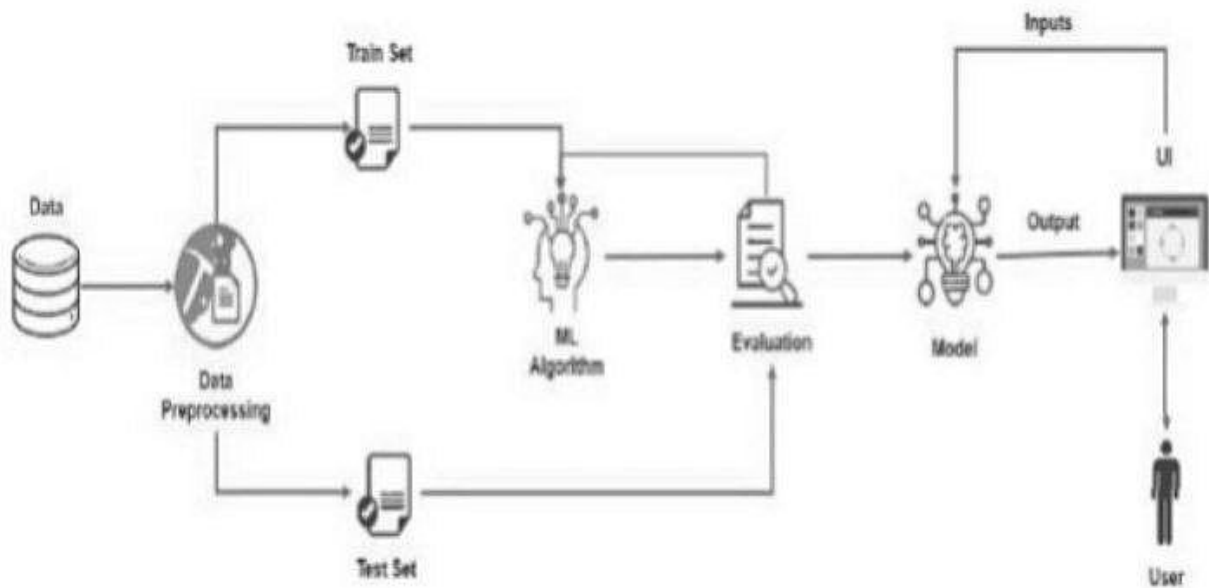


User Stories

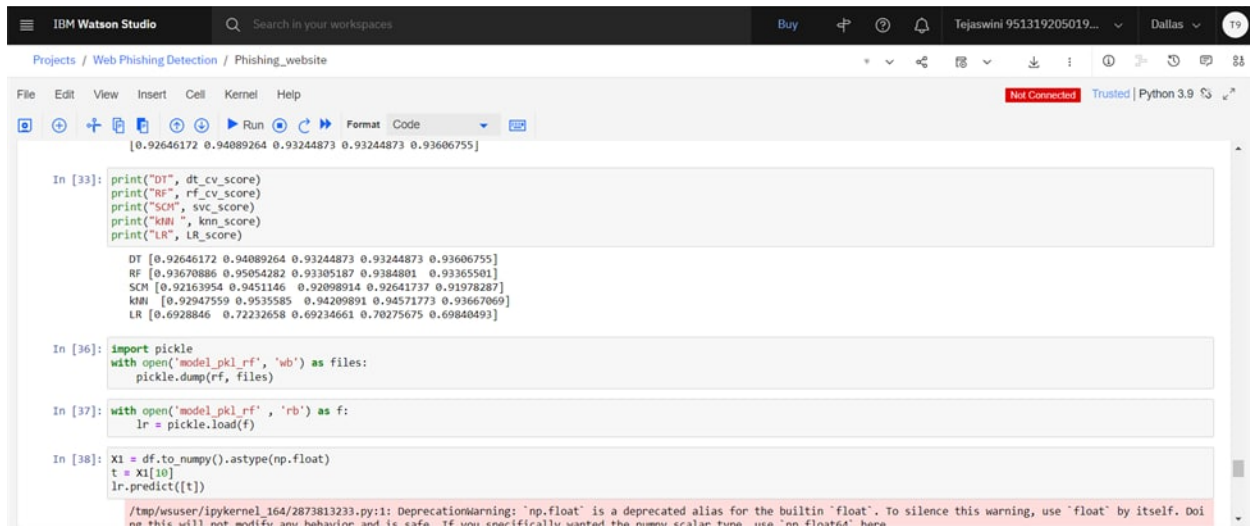
Use the below template to list all the user stories for the product.

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 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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
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 using heuristic and visual similarity approach.	As a User i can have comparison between websites for security.	High	Sprint-1
Administrator	Prediction	USN-1	Here the Model will predict the URL websites using Machine Learning algorithms such as Logistic Regression, KNN	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.	I this i will find the correct classifier for producing the result	Medium	Sprint-2

6.2 SOLUTION AND TECHNICAL ARCHITECTURE



7. EXPERIMENTAL INVESTIGATION



The screenshot displays the IBM Watson Studio interface. At the top, there's a navigation bar with 'IBM Watson Studio', a search bar, and user information. Below it, the project path 'Projects / Web Phishing Detection / Phishing_website' is shown. The main area contains a Jupyter notebook with the following code:

```
In [33]: print("DT", dt_cv_score)
print("RF", rf_cv_score)
print("SVM", svm_score)
print("knn", knn_score)
print("LR", lr_score)

DT [0.92646172 0.94089264 0.93244873 0.93244873 0.93606755]
RF [0.93670886 0.95054282 0.93305187 0.9384801 0.93365501]
SVM [0.92163954 0.9451146 0.92098014 0.92641737 0.91978287]
knn [0.92947559 0.9535585 0.94209891 0.94571773 0.93667069]
LR [0.6928846 0.72232658 0.69234661 0.70275675 0.69840493]

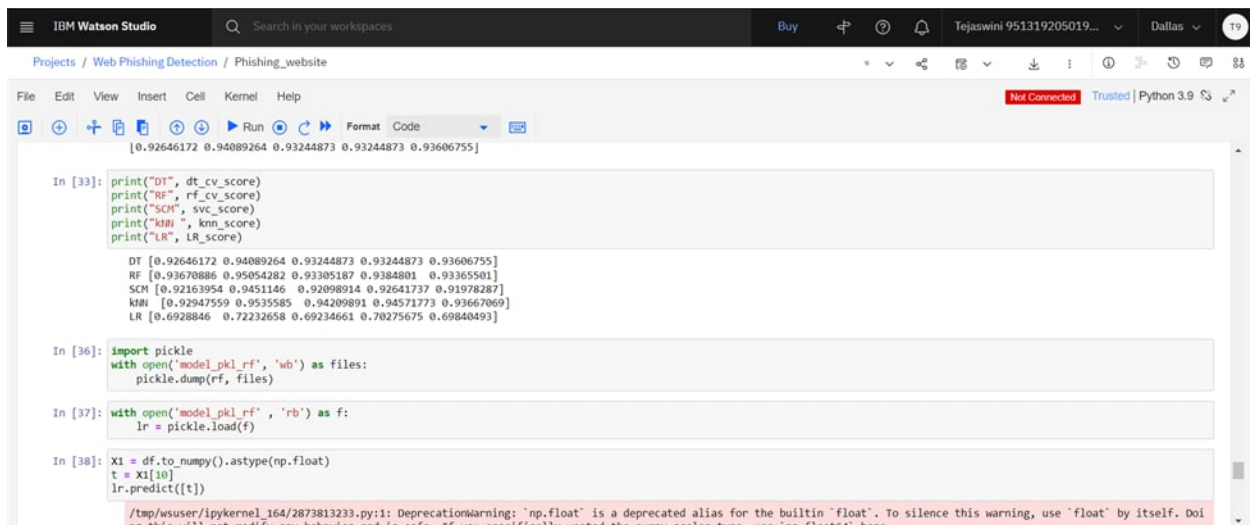
In [36]: import pickle
with open('model.pkl_rf', 'wb') as files:
    pickle.dump(rf, files)

In [37]: with open('model.pkl_rf', 'rb') as f:
    lr = pickle.load(f)

In [38]: x1 = df.to_numpy().astype(np.float)
t = x1[10]
lr.predict([t])

/tmp/ussuser/ipykernel_164/2873813233.py:1: DeprecationWarning: 'np.float' is a deprecated alias for the builtin 'float'. To silence this warning, use 'float' by itself. Do not use 'np.float' here.
```

8. FLOWCHART:



This screenshot is identical to the one above, showing the same IBM Watson Studio interface and Jupyter notebook code for model evaluation and saving.

9. RESULT:

Scikit-learn tool has been used to import Machine learning algorithms. Dataset is divided into training set and testing set . The classifier is trained using training set and testing set is used to evaluate performance of classifier. Performance of classifier has been evaluated by calculating

classifier's accuracy score, false negative rate and false positive rate.

Result also shows that detection accuracy of phishing websites increases as more dataset used as training dataset. All classifiers perform well when 90% of data used as training dataset.

Fig. 1 show the detection accuracy of the classifier when 50%, 70% and 90% of data used as training dataset and graph clearly shows that detection accuracy increases when 90% of data used as training dataset.

10. ADVANTAGES:

- Eliminate the cyber threat risk level.
- Increase user alertness to phishing risk.
- Instill a cyber security culture and create cyber security heroes.
- Change behavior to eliminate the automatic trust response.

DISADVANTAGES:

- Employees and customers inability to detect phishing emails and messages
- Insufficient communication between management organizations and employees or customers

11. APPLICATIONS:

- This system will be useful for many E-Commerce enterprises.
- This system will be useful for many users who purchase products online.

12. CONCLUSION

This paper aims to enhance detection method to detect phishing websites using machine learning technology. We achieved 97.14% detection accuracy using Logistic regression algorithm with lowest false positive rate. Also result shows that classifiers give better performance when we

used more data as training data. In future hybrid technology will be implemented to detect phishing websites more accurately, for which logistic regression algorithm of machine learning technology and blacklist method will be used.

13. FUTURE SCOPE:

- Phishing attacks in the future **could take multiple forms and could evolve beyond recognition**. For right now, your enterprise needs phishing protections such as email security to prevent the majority of phishing attacks from ever reaching your employees in the first place.

14.APPENDIX:

14.1. SOURCE CODE:

final.html:

```
<html>
```

```
  <head>
```

```
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
    rel="stylesheet">
```

```
                                <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>
```

```
  </head>
```

```
  <body>
```

```
    <nav class="navbar navbar-expand-sm bg-primary navbar-dark justify-content-right">
```

```
      <div class="container-fluid">
```

```
<ul class="navbar-nav">

  <label for="email" class="nav-link active">Nila</label>

  <li class="nav-item">

    <a class="nav-link active" href="#">Home</a>

  </li>

  <li class="nav-item">

    <a class="nav-link active" href="#">About</a>

  </li>

  <li class="nav-item">

    <a class="nav-link active" href="#">Contact</a>

  </li>

</ul>

</div>

</nav>

<div class="container-fluid">

  <div class="p-5">

    <center><h1>Phishing Website Detection using Machine Learning</h1></center>

  </div>

</div>
```

```

<div class="container mt-3">

    <form action="{{ url_for('y_predict') }}" method="post">

        <div class="mb-3 mt-3">

            <input type="text" class="form-control" placeholder="Enter the URL to be Verified"
name="URL" id="URL" required>

            <div class="p-3">

<div class="container mt-3">

<div class="d-grid gap-3">

                <center><button type="submit" class="btn btn-danger btn-lg
active">Predict</button></center>

        </div>

    </div>

</div>

</div>

</div>

</div>

</form>

</div>

<center><p><h4>{{ prediction_text }}</h4></p></center>

<center>

    https://www.thesmartbridg.com/Welcome/contactus

</center>

</body>

```

</html>

index.html:

<html>

<head>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet">

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container-fluid p-5 bg-primary text-white ">

<nav class="navbar navbar-expand-sm justify-content-right">

<div class="container-fluid">

<ul class="navbar-nav">

<label for="email" class="nav-link active float-start">Nil</label>

<li class="nav-item">

Home

<li class="nav-item">

About


```

<li class="nav-item">

  <a class="nav-link active" href="#">Contact</a>

</li>

<button type="button" class="btn btn-info">Get Started</button>

</ul>

</nav>

<div class="container mt-5 pt-10">

  <div class="row pt-8">

    <div class="col-sm-6 ">

      <p class="h2">Solution To Detect Phishing Websites</p>

      <p>Be aware of whate's happening with you confidential data</p>

      <div class="pt-6">

        <button type="button" href="" class="btn btn-info">Get Started</button>

        <a>Watch Video</a>

      </div>

    </div>

    <div>

      <div class="col-sm-6">

      </div>

    </div>

  </div>

```

</div>

</div>

<div class="container mt-5 text-danger">

<center><h1 >About</h1></center>

<div class="container mt-5">

<div class="row">

<div class="col-sm-6 text-dark">

<p>Web service is one of the key communications software services for the internet.

web phishing is one of many security threats to web services on the internet.

web phishing aims to steal private information, such as usernames, passwords, and credit card details,

by way of impersonating a legitimate entity.

</p>

</div>

<div class="col-sm-6 text-dark">

<p>The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information. It will lead to information disclosure and property damage

</p>

</div>

</div>

</div>

</div>

<div class="container-fluid p-5 bg-primary text-white ">

<div class="container mt-5">

<div class="row">

<div class="col-md-8 ">

<p class="h2"> check your website</p>

<p>understanding if the website is a valid one or not is important and plays a vital role in the securing the data. To know if the URL is a valid one or your information is at risk check your website. </p>

</div>

<div class="col-md-4 text-center">

<form name="frm" id="frm" method="get" action="/predict">

<button type="submit" class="btn btn-info">Check Your Website</button>

</form>

</div>

</div>

</div>

</div>

</body>

</html>

Phishing_website.ipynb:

```
#!/usr/bin/env python
# coding: utf-8

# In[47]:

'''
Download the dataset.
Preprocess or clean the data.
Analyze the pre-processed data.
Train the machine with preprocessed data using an appropriate machine
learning algorithm.
Save the model and its dependencies.
Build a Web application using a flask that integrates with the model built.
'''

# In[48]:

from sklearn import tree
from sklearn import svm
from sklearn import ensemble
from sklearn import neighbors
from sklearn import linear_model
from sklearn import metrics
from sklearn import preprocessing

# In[49]:

get_ipython().run_line_magic('matplotlib', 'inline')

from IPython.display import Image
import matplotlib as mlp
import matplotlib.pyplot as plt
```

```
import numpy as np
import os
import pandas as pd
import sklearn
import seaborn as sns

# In[50]:

#df = pd.read_csv('../input/mytest.csv')
df = pd.read_csv('dataset_website.csv')

print (df.shape)

#df.dtypes

# In[51]:

# Load data
df.head(3)

# In[52]:

df.info()

# In[53]:

df.isnull().sum()

# In[54]:

df.duplicated().sum()
```

```

# In[55]:

df['Google_Index'].value_counts()

# In[56]:

df.mean()

# In[57]:

# filling na values with mean
data = df.fillna(df.mean())

data.head(3)

# In[58]:

data.isnull().any()

# In[59]:

y = df["Result"].value_counts()
#print (y)
sns.barplot(y.index, y.values)

# In[60]:

y_True = df["Result"][df["Result"] == 1]
print ("Result Percentage = "+str( (y_True.shape[0] /
df["Result"].shape[0]) * 100 ))

```

```
# In[61]:
```

```
df.describe()
```

```
# In[62]:
```

```
df.groupby(["having_IPhaving_IP_Address",  
"Result"]).size().unstack().plot(kind='bar', stacked=True, figsize=(30,10))
```

```
# In[63]:
```

```
df.groupby(["URLURL_Length", "Result"]).size().unstack().plot(kind='bar',  
stacked=True, figsize=(5,5))
```

```
# In[64]:
```

```
y = df['Result'].to_numpy().astype(np.int)  
y.size
```

```
# In[65]:
```

```
df.drop(["Result"], axis = 1, inplace=True)
```

```
# In[66]:
```

```
X = df.to_numpy().astype(np.float)
```

```
# In[67]:
```

```
X
```

```
# In[68]:
```

```
X.shape
```

```
# In[69]:
```

```
scaler = preprocessing.StandardScaler()  
X = scaler.fit_transform(X)
```

```
# In[70]:
```

```
X
```

```
# In[71]:
```

```
def stratified_cv(X, y, clf_class, shuffle=True, n_folds=10, **kwargs):  
    stratified_k_fold = cross_validation.StratifiedKFold(y, n_folds=n_folds,  
shuffle=shuffle)  
    y_pred = y.copy()  
    # ii -> train  
    # jj -> test indices  
    for ii, jj in stratified_k_fold:  
        X_train, X_test = X[ii], X[jj]  
        y_train = y[ii]  
        clf = clf_class(**kwargs)  
        clf.fit(X_train, y_train)  
        y_pred[jj] = clf.predict(X_test)  
    return y_pred
```

```
# In[72]:
```

```
from sklearn.model_selection import cross_val_score
```

```
from sklearn.model_selection import train_test_split
```

```
# In[73]:
```

```
xtrain, ytrain, xtest, ytest=train_test_split(X, y, test_size=0.25, random_state=123)
```

```
rf=ensemble.RandomForestClassifier(max_depth=8, n_estimators=5)
rf_cv_score=cross_val_score(estimator=rf, X=xtrain, y=xtest, cv=5)
print(rf_cv_score)
```

```
# In[74]:
```

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(xtrain, xtest)
LR_score=cross_val_score(estimator=lr, X=xtrain, y=xtest, cv=5)
print(LR_score)
```

```
# In[75]:
```

```
from sklearn.linear_model import LinearRegression
LR = LinearRegression()
LR_score=cross_val_score(estimator=LR, X=xtrain, y=xtest, cv=5)
print(LR_score)
```

```
# In[76]:
```

```
from sklearn.neighbors import KNeighborsClassifier
knc = KNeighborsClassifier()
knc.fit(xtrain, xtest)
knn_score=cross_val_score(estimator=knc, X=xtrain, y=xtest, cv=5)
print(knn_score)
```

```
# In[77]:
```

```
from sklearn.svm import LinearSVC
SVC= LinearSVC()
svc_score=cross_val_score(estimator=SVC,X=xtrain,y=xtest,cv=5)
print(svc_score)
```

```
# In[78]:
```

```
from sklearn.tree import DecisionTreeClassifier
decTree = DecisionTreeClassifier(max_depth=6, random_state=0)
dt_cv_score=cross_val_score(estimator=decTree,X=xtrain,y=xtest,cv=5)
print(dt_cv_score)
```

```
# In[79]:
```

```
print("DT", dt_cv_score)
print("RF", rf_cv_score)
print("SCM", svc_score)
print("kNN ", knn_score)
print("LR", LR_score)
```

```
# In[80]:
```

```
import pickle
with open('model_pkl_knc', 'wb') as files:
    pickle.dump(knc, files)
```

```
# In[81]:
```

```
with open('model_pkl_knc' , 'rb') as f:
    lr = pickle.load(f)
```

```
# In[82]:
```

```
X1 = df.to_numpy().astype(np.float)
t = X1[10]
lr.predict([t])
```

```
# In[ ]:
```

app.py

```
import os
from flask import Flask, render_template, request, redirect, url_for, flash,
session, jsonify
import pickle
#import inputScript
import numpy as np
app = Flask(__name__)
model = pickle.load(open('model_pkl_rf' , 'rb'))
ASSET=os.path.join('static', 'assets')
@app.route("/")
def index():
    filename=os.path.join(ASSET, 'work2.jpg')
    return render_template('index.html', filename=filename)

@app.route("/predict")
def predict():
    return render_template('final.html')

@app.route('/y_predict', methods=['POST'])
def y_predict():
    url = request.form['URL']

    checkprediction = [[-1,1,1,1,-1,-1,-1,-1,-1,1,1,-1,1,-1,1,-1,-1,-1,0,1,1,1,1,-1,-1,-1,-1,1,1,-1,1]]
    #inputScript.main(url)
    prediction = model.predict(checkprediction)
    print("prediction", prediction)
    output = prediction[0]
    if output==1:
        pred ="You are safe! This is a legitimate Website."
    else:
```



```

        pred = "You are on the wrong site. Be cautious!"
        return render_template('final.html', prediction_text = '{}'.format(pred),
url=url)

@app.route('/predict_api', methods=['POST'])
def predict_api():
    data = request.get_json(force=True)
    prediction = model.y_predict([np.array(list(data.values()))])
    output = prediction[0]
    return jsonify(output)

if __name__ == "__main__":
    port = int(os.environ.get('PORT', 5000))
    app.run(debug=True, host='0.0.0.0', port=port)

```

14.2 GITHUB AND PROJECT DEMO LINK

- Gunter Ollmann, “The Phishing Guide Understanding & Preventing Phishing Attacks”, IBM Internet Security Systems, 2007.
- <https://github.com>
- <https://smartinternz.com>
- www.google.com

Github & Project Demo link

<https://github.com/IBM-EPBL/IBM-Project-43532-1660717592>

<https://drive.google.com/file/d/1YCNE2j0rD8cCxuZu00rOe5oQmkRr8cmw/view?usp=drivesdk>

14.3 REFERENCE LINK

<https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5>

<https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-net.2020.0078>

