PHISHING URL DETECTION SYSTEM

A Project report submitted in partial fulfilment of the requirements
for the award of the Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE ENGINEERING
By

Sree Anulekha Muddamsetty
(319136410079)
Nagasetti Syamalatha
(319136410083)
Narava Amrutha Kesini
(319136410085)
Chandinikumari Patnana
(319136410090)

Under the esteemed guidance of
Mr.Syed Mujib Rahaman
Associate Professor
Department of Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Dr. L. BULLAYYA COLLEGE OF ENGINEERING
(Affiliated to Andhra University, Visakhapatnam)
New Resapuvanipalem, Visakhapatnam-530013

Year of submission:2023

Dr. L. BULLAYYA COLLEGE OF ENGINEERING

New Resapuvanipalem, Visakhapatnam-530013

Department of Computer Science Engineering

Bonafide Certificate

This is to certify that **Ms. Sree Anulekha Muddamsetty, Nagasetti Syamalatha, Narava Amrutha Kesini, Chandinikumari patnana** bearing register numbers **319136410079, 319136410083, 319136410085, 319136410090** students of final year B. Tech in Computer Science Engineering, has carried out the project work titled **“Phishing URL detection system”** at Dr. L. Bullayya College of Engineering, Visakhapatnam during the academic year **2022-23.**

Project Supervisor

Mr.Syed Mujib Rahaman

Associate Professor

Dept. of Computer Science Engineering

Head of the Department

Dr. D. Madhavi

Professor

Dept. of Computer Science Engineering

ABSTRACT

Phishing websites, which are nowadays in a considerable rise, have the same look as legitimate sites. However, their backend is designed to collect sensitive information that is inputted by the victim.

Phishing is a form of fraudulent attack or a social engineering attack where the attacker tries to gain sensitive information by posing as a reputable source. In a phishing attack, a victim opens a compromised link that poses as a credible website. The victim is then asked to enter their credentials, but since it is a fake website, the sensitive information is routed to the hacker and the victim may also get's hacked.

Phishing is popular since it is a low effort, high reward attack.

Most modern web browsers, antivirus software and email clients are pretty good at detecting phishing websites at the source, helping to prevent attacks.

To understand how they work, this is to show you how to build your own phishing URL prediction system using Python.

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to **Prof D. Deepak Chowdary,** Principal, Dr. L. Bullayya College of Engineering for giving us the opportunity to do this project.

We would like to thank **Dr. D. Madhavi,** HOD, Department of CSE, Dr. L. Bullayya College of Engineering, Visakhapatnam, for her guidance in producing this work.

We are deeply indebted to our project guide and the head of research cluster **"**Privacy Preserving Network Security Data Science**" and** project guide **Mr.Syed Mujib Rahaman** Associate Professor, Department of Computer Science Engineering, Dr. Lankapalli Bullayya College of Engineering,Visakhapatnam, for guiding us throughout the project in spite of his busy schedule.

Apart from the efforts of our, the success of this project depends largely on the encouragement of other faculty of CSE, Dr. Lankapalli Bullayya College of Engineering, Visakhapatnam. We take this opportunity to express my gratitude to the entire faculty who has been instrumental in the successful completion of this project.

Also deserving of thanks for our family and friends, for their support for their confidence in our achievements.

Sree Anulekha Muddamsetty
Nagasetti Syamalatha
Narava Amrutha Kesini
ChandiniKumari Patnana

DECLARATION

This is to declare that the Project work entitled "**Phishing URL Detection system"** is a bonafide work done by us under the research cluster group "Privacy Preserving Network Security Data Science" with the esteemed guidance of Mr.Syed Mujib Rahaman, Associate Professor, Department of CSE, Dr.L.Bullayya College of Engineering. This project report is being submitted in the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science Engineering during the academic year 2022-2023. This project possesses originality as it is not extracted from any source and it has not been submitted to any other institutions and university.

(Sree Anulekha Muddamsetty)
Reg.No:319136410079

(Nagasetti Syamalatha)
Reg.No:319136410083

(Narava Amrutha Kesini)
Reg.No:319136410085

(Chandinikumari Patnana)
Reg.No:319136410090

Visakhapatnam
Date:

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

1.INTRODUCTION

Phishing is a type of attack that is often used to steal user data, including login credentials and credit card numbers. Phishing is a popular form of cybercrime because of how effective it is.
The first phishing attacks happened in the mid-1990s, when a group of hackers posed as employees of AOL and used instant messaging and email to steal users' passwords and hijack their accounts.

Cybercriminals have been successful in using emails, text messages, and direct messages on social media or in video games, to get people to respond with their personal information.

Phishing URL's, which are nowadays in a considerable rise, have the same look as legitimate sites. However, their backend is designed to collect sensitive information that is inputted by the victim.

Phishing is an attack that attempts to steal your money, or your identity, by getting you to reveal personal information such as credit card numbers, bank information, or passwords on websites that pretend to be legitimate. Cybercriminals typically pretend to be reputable companies, friends, or acquaintances in a fake message, which contains a link to a phishing website.

In a phishing attack, a victim opens a compromised link that poses as a credible website. The victim is then asked to enter their credentials, but since it is a fake website, the sensitive information is routed to the hacker and the victim may also get's hacked.

Phishing and hacking are driven by similar intents as both are primarily used to defraud people in some way. However, phishing relies on people voluntarily providing information while hacking involves forcefully gaining unauthorized access to it, such as by disabling the security measures of a computer network.

Phishing is popular since it is a low effort, high reward attack. Most modern web browsers, antivirus software and email clients are pretty good at detecting phishing websites at the source, helping to prevent attacks.

One of the real life **classic example is a phishing email from Netflix that says "Your account has been suspended". It asks you to click a link and give your details to reactivate your account. The attackers then harvest those details and either use them to commit fraud, or sell them on the dark** web.

There are actually multiple types of phishing scams that businesses are targeted by on a daily basis ; like-

- Whaling
- Spear phishing
- Vishing
- Watering hole phishing
- Smishing, etc..

1. Whaling

   A whaling attack, also known as whaling phishing or a whaling phishing attack, is a specific type of phishing attack that targets high-profile employees, such as the chief executive officer or chief financial officer, in order to steal sensitive information from a company. In many whaling phishing attacks, the attacker's goal is to manipulate the victim into authorizing high-value wire transfers to the attacker.

2. Spear Phishing

   Spear phishing is a phishing method that targets specific individuals or groups within an organization.

   Spear phishers carefully research their targets, so the attack appears to be from trusted senders in the targets' life. A spear phishing emails in such a way to urge the victim to click on a suspicious link or attachment. Once the victim completes the intended action, the attacker can steal the credentials of a targeted legitimate user and enter a network undetected.

3. Vishing

   Vishing is a different attack that falls under the general phishing umbrella and shares the same goals as phishing. Vishers use fraudulent phone numbers, voice-altering software, text messages.Vishing generally uses voice to trick users. A vishing attack might also start with a text message and contain a phone number asking users to call, but vishing attacks could also use automated messages and robocalls.

4. Water hole phishing

   A watering hole attack is a targeted attack designed to compromise users within a specific industry or group of users by infecting websites they typically visit . The end goal is to infect the user's computer and gain access to the organizations network.

5.Smishing

Smishing is a closely related phishing attack that also uses smartphone numbers. But instead of voice mail, smishing uses text messages to trick users.

There is a lot of overlap between smishing and vishing. A vishing attack might also start with a text message and contain a phone number asking users to call, but vishing attacks could also use automated messages and robocalls. Smishing can also include a phone number in a text message, but many attacks focus mainly on tricking users into clicking links and opening a malicious website page.

Signs of Phishing

- Danger or Sense of urgency
- Message style
- Varieties in web addresses
- Language mistakes
- Interest for identification, payment, or other personal Information
- Misspelled
- Points to the wrong top-level domain
- A combination of a valid and a fraudulent URL
- Is incredibly long
- Is just be an IP address
- Has a low page rank
- All the above characteristics of a phishing URL can help us to distinguish it from a valid URL.

1.1 Background and Motivation

The first time someone used the term 'phishing' can be traced back to January 2nd, 1996. The 2000s and 2010s is when phishing has started evolving at a rapid pace.

In the early 2000s, people still didn’t know much about phishing. It wasn’t widespread knowledge that scammers pretend to be trusted authorities to score a jackpot.During this period, phishers started to turn their attention to online payment gateways, such as Paypal and E gold. For example, scammers sent an email to Paypal users and at the time there were already a lot of users telling them to update their credit card details but stole their details instead.

The late 2008 brought forth crypto currencies, untraceable payment methods that hackers use to collaborate with each other, extort their victims, or cash out on their most recent scams securely. Ransomware, which are mainly sent through phishing emails, runs rampant starting from the Cryptolocker ransomware in 2013, to various other worms.

The loss caused by a ransomware attack isn’t small either. Most lost millions of dollars, and that’s only from the ransom. There are still fines, operational costs, and restoration costs to consider.

In the early 2010s, you also see a shift on how hackers use phishing attacks, with more of them using it for a larger purpose than the usual financial goals. Today, while cybersecurity experts are catching up, it’s far from enough. Both security researchers and hackers are stuck in a never ending battle where they constantly try to one-up the other using new technologies, scenarios, and attack methods.

With the growth of social media like LinkedIn or Facebook, cybercriminals found a new treasure of information, where they can do research and make their phishing messages more specific and thus, convincing. Unrestricted access to sensitive information helps hackers build personalized spear phishing emails that rely on familiarity and make it harder for users to detect a phishing attempt.

The pandemic forced a lot of companies to go remote, improving the success rate of phishing campaigns over the past couple of years. While companies and employees are adapting to the new remote work security guidelines, hackers took this as an opportunity to attack more small businesses as they don’t have much security as larger companies for a bigger payout.

A report mentions that in March 2020, there were 589% more phishing attacks compared to February 2020. That’s a nearly 600% increase over a month, which just shows how much hackers are capitalizing on the panic caused by the pandemic.

Additionally, while emails have been dominating in phishing the past decade, 2020 marked the rise in scams done through phone calls (vishing) and SMS or text messages (smishing).

Despite almost three decades of dealing with phishing attacks, a lot of cybersecurity researchers are continuously looking for ways to combat hackers, malicious actors are also looking for more creative techniques to fool users. Fortunately, there are still ways to detect and prevent phishing attacks even when the method changes.

So, many of the individuals and companies who had been suffered and suffering from the phishing attacks made us the vision and to know deeper about this project called “ Phishing URL Detection”.

1.2 Problem Statement

Phishing is a type of fraud in which an attacker impersonates a reputable company or person in order to get sensitive information such as login credentials or account information via email or other communication channels.

URL is the first thing to analyse a website to decide whether it is a legitimate or not. URLs of phishing have some distinctive features.

Some of the common features of Phishing URL’s are

- Having IP address in its domain
- Long URL’s
- Shortening URL’s
- URL’s having ‘@’ symbols
- Redirecting using ‘//’
- Adding prefixes or suffixes separated by ‘-‘ in domain
- Mispelled ,etc..

The above features which are related are obtained when the URL is processed through which we have to decide whether it is a phishing URL or not.

2. REQUIREMENT ELICITATION AND ANALYSIS

2.1 Existing System

The security of personal data is crucial for a company or any individual. Phishing is one of the most common and dangerous cybercrime attacks. Even though there are several systems and solutions, the amount of personal information stolen continues to increase as cyberattacks become more difficult to detect. In existing systems, it consists of a broad review to study the work carried out in the fight against phishing and the identification of vulnerabilities in existing systems to achieve better efficiency. The authors focused on the social medium Twitter to study the phishing attacks passing through this medium, and they present their new design, which is based on new features. The classification of the approach includes 23 features and uses the logistic regression algorithm. Experiments show that the system is effective at detecting phishing URL's, with a 93% success rate using recent data.

2.2 Proposed System

In this proposed system we will be predicting the phishing URL's by means of building a prediction system with the records/data set that consists the features of various URL's having legitimate and also suspicious forms that helps to detect the phishing URL i.e., whether a valid URL is detected or an invalid URL . This can be done by a classifier model which helps to know about a particular URL is a phishing or a legitimate.

2.3 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

- It is simply an important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resources and time.

The different feasibilities that have to be analyzed and the main key consideration involved in the feasibility analysis is /are:

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

Operational Feasibility: Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which was previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

Economic Feasibility: Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer-based project. As software was installed from the beginning & for lots of purposes thus the cost on project of software is low. Since the system is network based, any number of employees connected to the LAN within that organization can use this tool from any time. The Virtual Private Network is to be developed using the existing resources of the organization. So, the project is economically feasible.

Technical feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4 System Requirements

The system requirements or software requirements is a listing of what software programs or hardware devices are required to operate the program or game properly. System requirements is a statement that identifies the functionality that is needed by a system in order to satisfy the user's requirements. They are the first and foremost important part of any project, because if the system requirements are not fulfilled, then the project is not complete.
A software requirements is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

A software requirement can be of 2 types:
1. Functional Requirements
2. Non-functional Requirements

2.4.1 Functional requirements:

The functional requirements for a system describe what the system should do. Those requirements depend on the type of software being developed, the expected users of the software. These are statement of services the system should provide, how the system should react to particular inputs andhow the system should behave in particular situation.

-First, we should import all the necessary datasets.
-We will read the data frame.
-The values in the class(target column) are -1 and 1.
-Then fit transform () function is used to fit and transforms data.
-Building a prediction system using machine learning model.

2.4.2 SYSTEM REQUIREMENT SPECIFICATION

Functional requirements table

S.no	Requirement	Requirement no.	Essential	Description
1	Input URL’s	RS1	essential	Dataset should be loaded without any errors
2	Data preprocessing	RS2	essential	Data should not have any null values or noisy data.
3	Feature Selection & Extraction	RS3	essential	Feature selection/ extraction is performed
4	Training model	RS4	essential	According to the given ratio the dataset should be split into train and test data
5	Decision tree classifier is called for given data set	RS5	essential	the model applied should work without any errors
6	Detection	RS6	essential	Detection should be done correctly

2.4.3 Non Functional Requirements

Non-Functional Requirement (NFR) specifies the quality attribute of a software systems. Non Functional requirements in Software Engineering allows you to impose constraints or restrictionson the design of the system

Hardware requirements:

The hardware requirements are the requirements of a hardware device.

1. Intel i3 processor or above
2. RAM 4GB or above
3. Hard disk 50GB

Software requirements:

The software requirements are the requirements of a software device.

2.3.1 Python:3.8.5.2

2.3.2 Numpy

3. pandas: 1.1.5
4. matplotlib: 3.2.2
5. Scikit-learn: 0.24.2
6. Seaborn
7. Minmax Scaler

3. SYSTEM DESIGN

System Design is a solution to how to approach to the creation of a system. System Design is the process of designing the architecture, components, and interfaces for a system so that it meets the end- user requirements. System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. System design could be seen as the application of the system theory to product development. There is some overlap with the disciplines of the system analysis, system architecture and system engineering. System modelling is the interdisciplinary study of the use of mobile to conceptualize and construct system in business and IT development. This important phase provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. The design step produces a data design, an architectural design and a procedural design. The data design transforms the information domain model created during analysis in to the data structures that will be required to implement the software. The architectural design defines the relationship among major structural components into a procedural description of the software.

Source code generated and testing is conducted to integrate and validate the software. From a project management point of view, software design is conducted in two steps.

Preliminary design is connected with the transformation of requirements into data and software architecture.

Detailed design focuses on refinements to the architectural representation that leads the detailed data structure and algorithmic representations of software.

Reliability in System Design

A system is Reliable when it can meet the end-user requirement. When designing it, we should have planned to implement a set of features and services in your system. If the system can serve all those features without wearing out then your System can be considered to be Reliable.

A Fault Tolerant system can be one that can continue to be functioning reliably even in the presence offaults. Faults are the errors that arise in a particular component of the system. An occurrence of fault doesn’t guarantee Failure of the System. Failure is the state when the system is not able to perform as expected. It is no longer able to provide certain services to the end-users.

Availability in System Design

Availability is a characteristic of a System which aims to ensure an agreed level of Operational Performance, also known as uptime. It is essential for a system to ensure high availability in order to serve the user’s requests. The extent of Availability varies from system to system.

“Suppose when we are designing a Social Media Application then high availability is not much of a need. A delay of a few seconds can be tolerated. Getting to view the post on Instagram with a delay of 5 to 10 seconds will not be much of an issue. But if you are designing a system for hospitals, Data center, or Banking, then you should ensure that your system is highly available. Because a delay in the service can lead to a huge loss.”

So now as we are designing a system regarding hospitals, we should make our system is highly available.

There are various principles you should follow in order to ensure the availability of your system:

- System should not have a Single Point of Failure. Basically, your system should not be dependent ona single service in order to process all of its requests.
- Because when that service fails then your entire system can be end up becoming unavailable.
- Detecting the Failure and resolving it at that point.

Scalability in System

Design Scalability refers to the ability of the System to cope up with the increasing load. While designing the system you should keep in mind the load experienced by it. It’s said that if you have to design a system for load X then you should plan to design it for 10X and test it for 100X. There can be a situation where your system can experience an increasing load.

3.1 Object Oriented Analysis and Design

"Object-oriented analysis is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain".
Object–Oriented Analysis (OOA) is the procedure of identifying software engineering requirements and developing software specifications in terms of a software system's object model, which comprises of interacting objects. The main difference between object-oriented analysis and other forms of analysis is that in object-oriented approach, requirements are organized around objects, which integrate both data and functions. They are modelled after real-world objects that the system interacts with. In traditional analysis methodologies, the two aspects - functions and data - are considered separately.

The primary tasks in object-oriented analysis (OOA) are:

♣ Identifying objects

♣ Organizing the objects by creating object model diagram

♣ Defining the internals of the objects, or object attributes

♣ Defining the behavior of the objects, i.e., object actions

♣ Describing how the objects interact

The common models used in OOA are use cases and object models.

Object–Oriented Design (OOD) involves implementation of the conceptual model produced during object-oriented analysis. In OOD, concepts in the analysis model, which are technology−independent, are mapped onto implementing classes, constraints are identified and interfaces are designed, resulting in a model for the solution domain, i.e., a detailed description of how the system is to be built on concrete technologies.
The implementation details generally include:

♣ Restructuring the class data (if necessary),

♣ Implementation of methods, i.e., internal data structures and algorithms,

♣ Implementation of control, and

♣ Implementation of associations.

Generally, object-oriented design as "a method of design encompassing the process of object-oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models of the system under design".

Object-oriented programming (OOP) is a programming paradigm based upon objects (having both dataand methods) that aims to incorporate the advantages of modularity and reusability. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs.

The important features of object–oriented programming are:

♣ Bottom–up approach in program design

♣ Programs organized around objects, grouped in classes

♣ Focus on data with methods to operate upon object's data

♣ Interaction between objects through functions

♣ Reusability of design through creation of new classes by adding features to existing classes Some examples of object-oriented programming languages are C++, Java, Smalltalk, Delphi, C#, Perl, Python, Ruby, and PHP.

And also, the Object-Oriented Modelling (OOM) technique visualizes things in an application by usingmodels organized around objects.

Any software development approach goes through the following stages:

♣ Analysis

♣ Design and

♣ Implementation.

3.1.1 Use-case

A use case diagram is a Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. A use case is a set of scenarios that describesan interaction between a user and a system.

A use case diagram displays the relationship among actors and use cases.

The two main components of a use case diagram are use cases and actors. The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose.

Use case diagrams are used to gather the requirements are mostly design requirements. So, when asystem is analyzed to gather its functionalities use cases are prepared and actors are identified.

So, in brief, the purpose of use case diagrams:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.

Importance of Use Case Diagram:

- To identify the functions and how roles interact with them-The primary purpose of use case diagrams
- For a high -level view of the system- Especially useful when presenting to managers or stakeholders. You can highlight the roles that interact with the system and the functionality provided by the system without going deep into the inner working of the system
- To identify internal and external factors – This might sound simple but in large complex projects a system can be identified as an external role in another use case.

Use Case Diagram objects:

Use case diagrams consists of 4 objects:

- Actor
- Use case
- System

The objects are further explained below.

Actor: Actor in a use case diagram is any entity that performs a role in one given system. This couldbe a person, organization, or an external system and usually drawn like a skeleton.

Use case: A use case represents a function or an action within the system. It is drawn as an oval andnamed with the function.

System: The system is used to define the scope of the use case and drawn as a rectangle. This is an optional element but useful when you’re visualizing large systems.
For example, you can create all the use cases and then use the system object to define the scope covered by your project. Or you can even use it to show the different areas covered in different releases.

Relationships in use case diagrams:
There are five types of relationships in use case diagrams.

They are:

- Association between an actor and a use case
- Generalization of an actor
- Extend relationship between two use cases
- Include relationship between two use cases

Use-case diagram

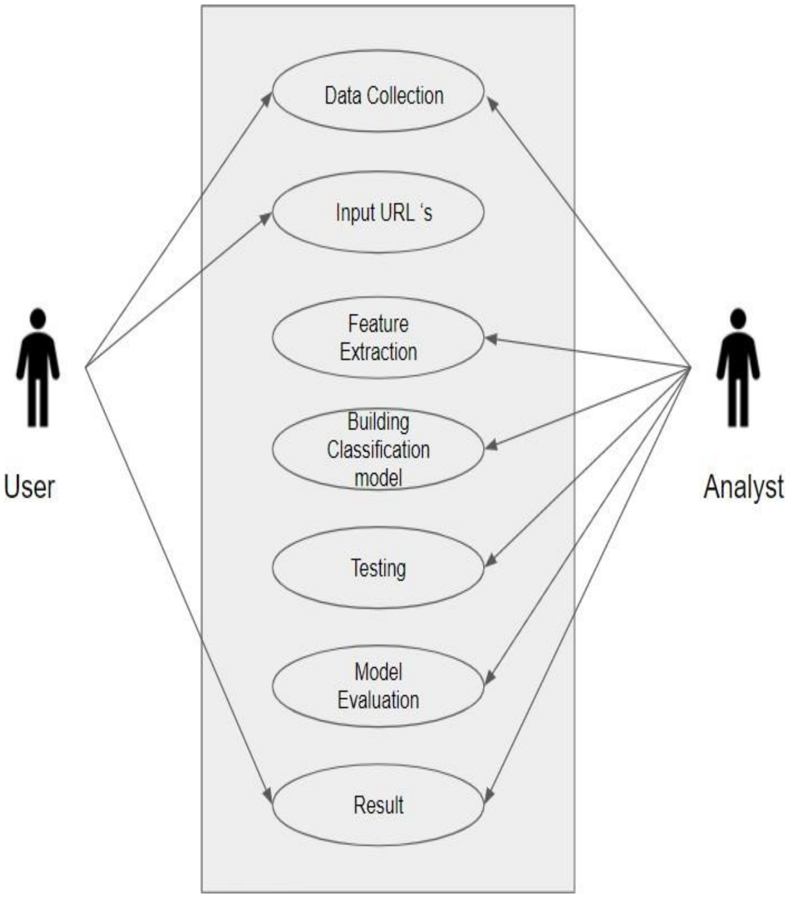


Fig-(1)

3.1.2 Sequence Diagram:

Sequence diagrams, commonly used by developers, model the interactions between the objects to the single use case. They illustrate how the different parts of a system interacts with each other to carry outa function and the order in which the interactions occur when a particular use case is executed in simplewords, a sequence diagram shows different parts of a system in sequence to get something done.
A sequence diagram describes an interaction among a set of objects participated and arranged in a chronological order. It shows the objects participating in the interaction and the messages that they send to each other.

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function,operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Sequence diagram for preprocessing

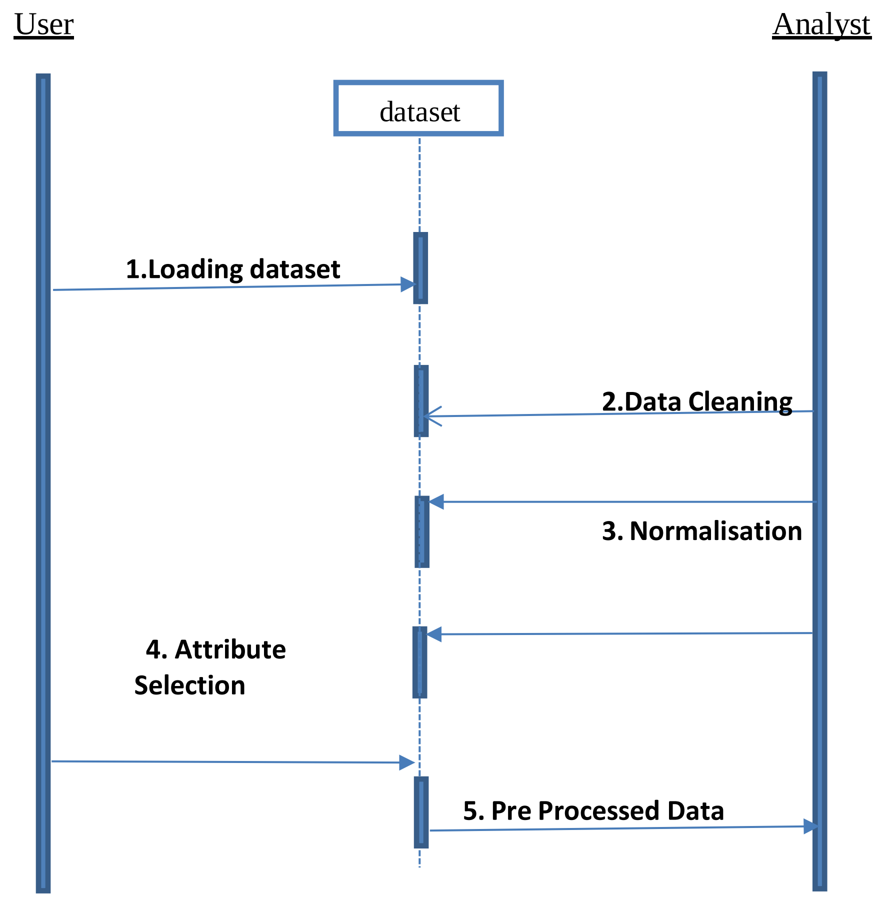


fig-(2)

3.1.3 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling.The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

The middle compartment contains the attributes of the class.

The bottom compartment contains the operations the class can execute.

Class Diagram-

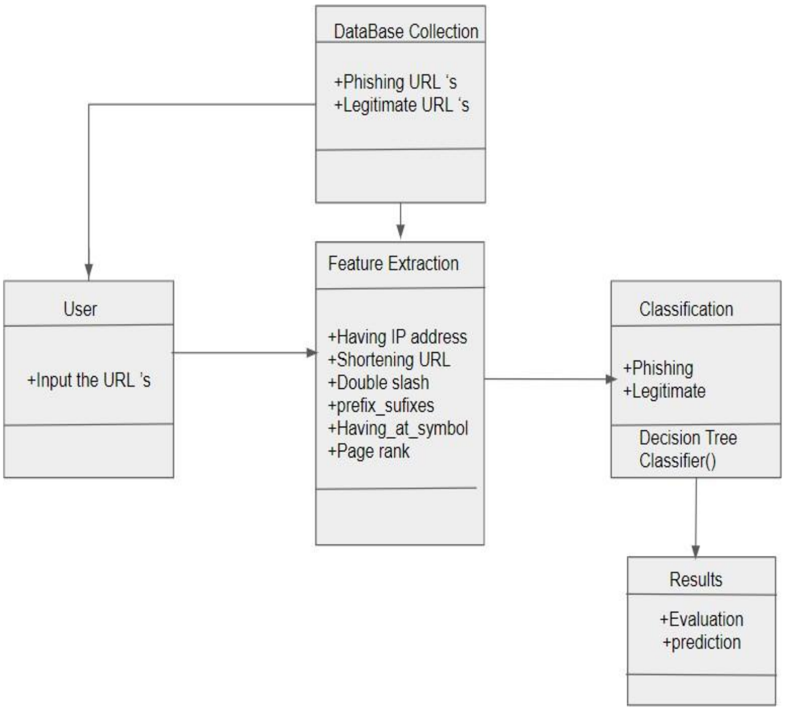
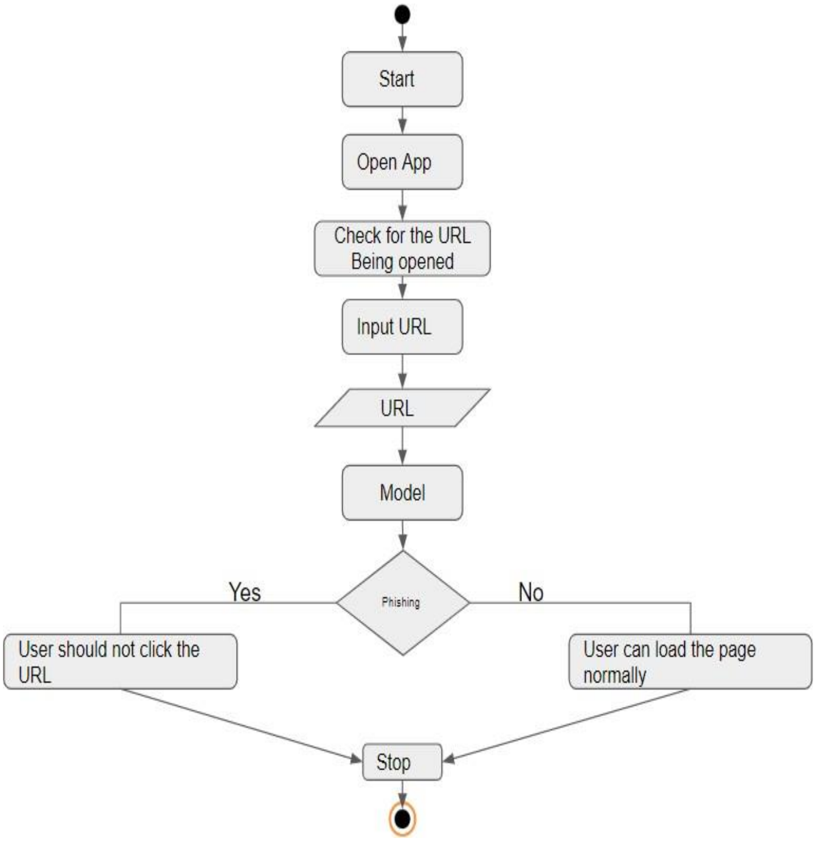


Fig-(3)

3.1.4 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data.

Activity diagram


Start
Open App
Check for the URL Being opened
Input URL
URL
Model
Yes
Phishing
No
User should not click the URL
User can load the page normally
Stop

Fig(4)

3.1.5 Swimlane Activity Diagram

The Activity diagrams in Object Oriented design are just like the flow carts that show the sequence of steps that make up a complex process, such as an algorithm or workflow. Activity diagram are most useful during the initial stages of the design phase.

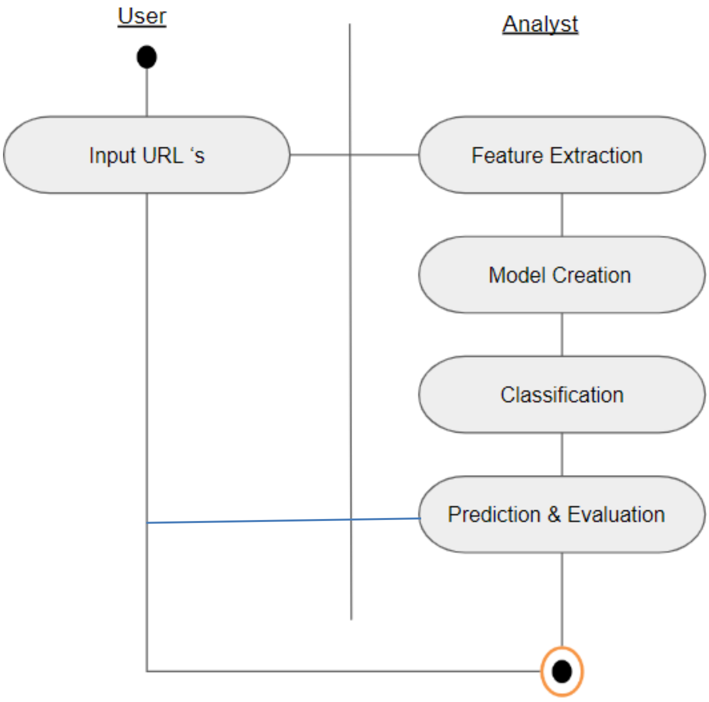
These can be both sequential and in parallel. They describe the objects used, consumed or produced by an activity and the relationship between the different activities.

Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The onlymissing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single. Before drawing an activity diagram, we must have a clear understanding about the elements used in the activity diagram. The main element of an activity diagram is the activity itself. Activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Swimlanes are used to show which activities are performed by which organization in the activity diagram. The lanes are boundaries are drawn and the activities of a particular organization are drawn in the same lane as that of the organization.

Swim lanes have to be ordered in a Logical Manner. It is suggested to have less than five swim lanes in an activity diagram. Swim lanes are good in that they combine the activity diagram’s depiction of logic with the interaction diagram’s depiction of responsibility.

Swimlane Activity diagram (usecase+activity):

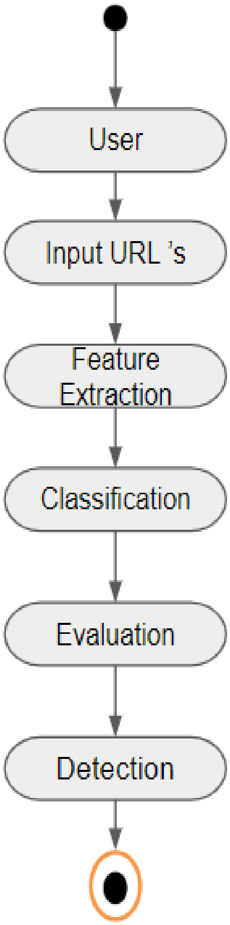


Fig(5)

3.1.6 State chart Diagram

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. As the State chart diagram defines the states, it is used to model the lifetime of an object. State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of a state chart diagram is to model the lifetime of an object from creation to termination

State chart diagram



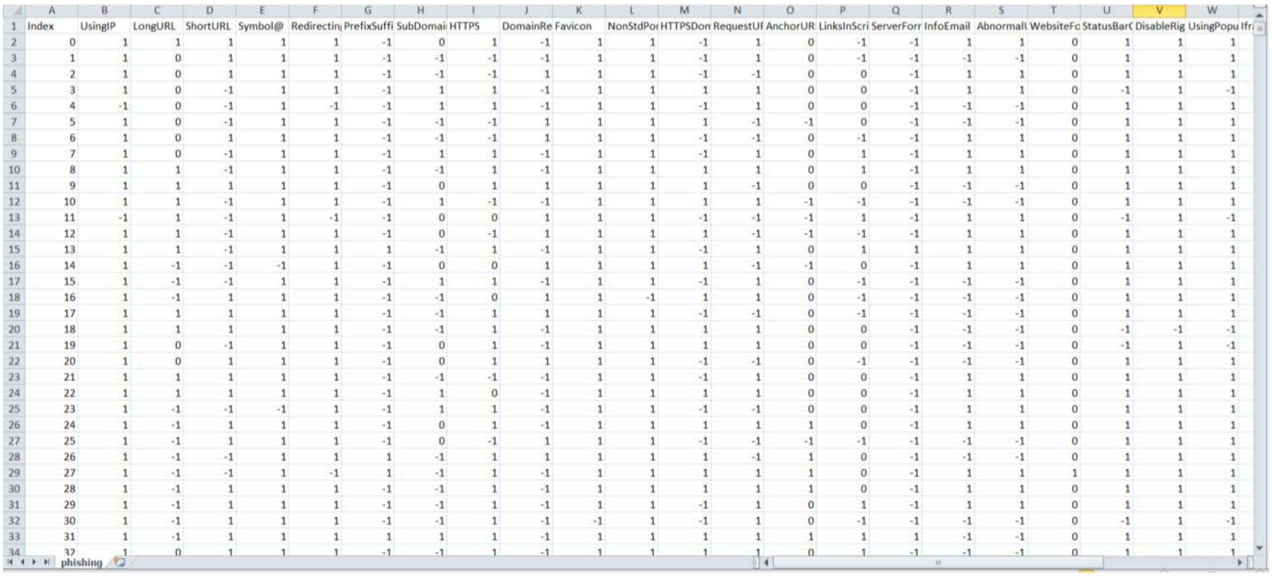
Fig(6)

3.2 Dataset

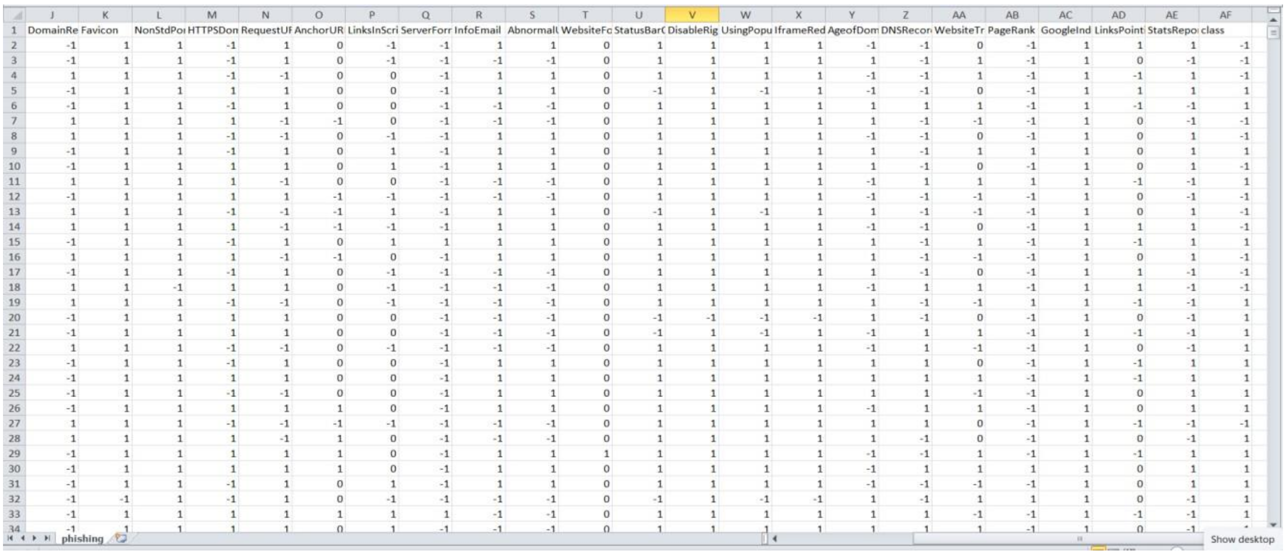
A data set is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set. The data set lists values for each of the variables and for each member of the data set. Data sets can also consist of a collection of comma separated values (csv),documents or files.

The data set for “phishing URL detection” considered here is a csv file consisting of 11055 rows and 33 columns; where each column representing the various features that are used for detecting the phishing URL by means of building a model

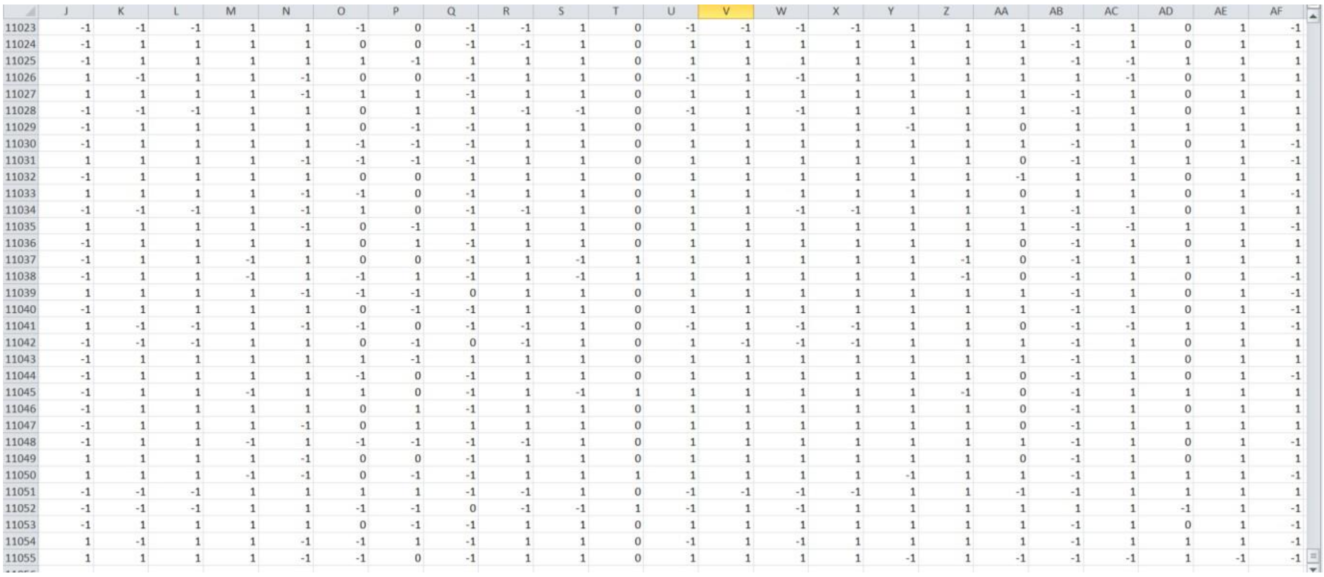
csv file:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirectin	PrefixSuffi	SubDomai	HTTPS	DomainRe	Favicon	NonStdPo	HTTPSDon	RequestUF	AnchorUR	LinksInScri	ServerForr	InfoEmail	AbnormalU	WebsiteFo	StatusBar(	DisableRig	UsingPopu	Ifr
2	0	1	1	1	1	1	-1	0	1	-1	1	1	-1	1	0	-1	-1	1	1	0	1	1	1	
3	1	1	0	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	-1	-1	-1	-1	0	1	1	1	
4	2	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	0	-1	1	1	0	1	1	1	
5	3	1	0	-1	1	1	-1	1	1	-1	1	1	1	1	0	0	-1	1	1	0	-1	1	-1	
6	4	-1	0	-1	1	-1	-1	1	1	-1	1	1	-1	1	0	0	-1	-1	-1	0	1	1	1	
7	5	1	0	-1	1	1	-1	-1	-1	1	1	1	1	-1	-1	0	-1	-1	-1	0	1	1	1	
8	6	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	-1	-1	1	1	0	1	1	1	
9	7	1	0	-1	1	1	-1	1	1	-1	1	1	-1	1	0	1	-1	1	1	0	1	1	1	
10	8	1	1	-1	1	1	-1	-1	1	-1	1	1	1	1	0	1	-1	1	1	0	1	1	1	
11	9	1	1	1	1	1	-1	0	1	1	1	1	1	-1	0	0	-1	-1	-1	0	1	1	1	
12	10	1	1	-1	1	1	-1	1	-1	-1	1	1	1	1	-1	-1	-1	-1	-1	0	1	1	1	
13	11	-1	1	-1	1	-1	-1	0	0	1	1	1	-1	-1	-1	1	-1	1	1	0	-1	1	-1	
14	12	1	1	-1	1	1	-1	0	-1	1	1	1	1	-1	-1	-1	-1	1	1	0	1	1	1	
15	13	1	1	-1	1	1	1	-1	1	-1	1	1	-1	1	0	1	1	1	1	0	1	1	1	
16	14	1	-1	-1	-1	1	-1	0	0	1	1	1	1	-1	-1	0	-1	1	1	0	1	1	1	
17	15	1	-1	-1	1	1	-1	1	1	-1	1	1	-1	1	0	-1	-1	-1	-1	0	1	1	1	
18	16	1	-1	1	1	1	-1	-1	0	1	1	-1	1	1	0	-1	-1	-1	-1	0	1	1	1	
19	17	1	1	1	1	1	-1	-1	1	1	1	1	-1	-1	0	-1	-1	-1	-1	0	1	1	1	
20	18	1	1	1	1	1	-1	-1	1	-1	1	1	1	1	0	0	-1	-1	-1	0	-1	-1	-1	
21	19	1	0	-1	1	1	-1	0	1	-1	1	1	1	1	0	0	-1	-1	-1	0	-1	1	-1	
22	20	1	0	1	1	1	-1	0	1	1	1	1	-1	-1	0	-1	-1	-1	-1	0	1	1	1	
23	21	1	1	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	0	-1	1	1	0	1	1	1	
24	22	1	1	1	1	1	-1	1	0	-1	1	1	1	1	0	0	-1	1	1	0	1	1	1	
25	23	1	-1	-1	-1	1	-1	1	1	-1	1	1	-1	-1	0	0	-1	1	1	0	1	1	1	
26	24	1	-1	1	1	1	-1	0	1	-1	1	1	1	1	1	0	-1	1	1	0	1	1	1	
27	25	1	-1	1	1	1	-1	0	-1	1	1	1	-1	-1	-1	-1	-1	-1	-1	0	1	1	1	
28	26	1	-1	-1	1	1	1	-1	1	1	1	1	1	-1	1	0	-1	-1	-1	0	1	1	1	
29	27	1	-1	-1	1	-1	1	-1	1	-1	1	1	1	1	1	0	-1	1	1	1	1	1	1	
30	28	1	-1	1	1	1	-1	-1	1	-1	1	1	1	1	1	0	-1	1	1	0	1	1	1	
31	29	1	-1	1	1	1	-1	-1	1	-1	1	1	-1	1	0	1	-1	1	1	0	1	1	1	
32	30	1	-1	1	1	1	-1	-1	1	-1	-1	1	-1	1	0	-1	-1	-1	-1	0	-1	1	-1	
33	31	1	-1	1	1	1	1	1	1	-1	1	1	1	1	1	1	1	-1	-1	0	1	1	1	
34	32	1	0	1	1	1	-1	-1	1	-1	1	1	1	1	0	1	-1	-1	-1	0	1	1	1	



	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1	DomainRe	Favicon	NonStdPoı	HTTPSDon	RequestUF	AnchorUR	LinksInScri	ServerForr	InfoEmail	Abnormal(	WebsiteFc	StatusBar(	DisableRig	UsingPopu	IframeRed	AgeofDom	DNSRecor	WebsiteTr	PageRank	GoogleInd	LinksPoint	StatsRepoı	class
2	-1	1	1	-1	1	0	-1	-1	1	1	0	1	1	1	1	-1	-1	0	-1	1	1	1	-1
3	-1	1	1	-1	1	0	-1	-1	-1	-1	0	1	1	1	1	1	-1	1	-1	1	0	-1	-1
4	1	1	1	-1	-1	0	0	-1	1	1	0	1	1	1	1	-1	-1	1	-1	1	-1	1	-1
5	-1	1	1	1	1	0	0	-1	1	1	0	-1	1	-1	1	-1	-1	0	-1	1	1	1	1
6	-1	1	1	-1	1	0	0	-1	-1	-1	0	1	1	1	1	1	1	1	-1	1	-1	-1	1
7	1	1	1	1	-1	-1	0	-1	-1	-1	0	1	1	1	1	1	-1	-1	-1	1	0	-1	-1
8	1	1	1	-1	-1	0	-1	-1	1	1	0	1	1	1	1	-1	-1	0	-1	1	0	1	-1
9	-1	1	1	-1	1	0	1	-1	1	1	0	1	1	1	1	1	-1	1	1	1	0	1	1
10	-1	1	1	1	1	0	1	-1	1	1	0	1	1	1	1	1	-1	0	-1	1	0	1	-1
11	1	1	1	1	-1	0	0	-1	-1	-1	0	1	1	1	1	-1	1	1	1	1	-1	-1	1
12	-1	1	1	1	1	-1	-1	-1	-1	-1	0	1	1	1	1	-1	-1	-1	-1	1	0	-1	-1
13	1	1	1	-1	-1	-1	1	-1	1	1	0	-1	1	-1	1	1	-1	-1	-1	1	0	1	-1
14	1	1	1	1	-1	-1	-1	-1	1	1	0	1	1	1	1	-1	-1	0	-1	1	1	1	-1
15	-1	1	1	-1	1	0	1	1	1	1	0	1	1	1	1	1	-1	1	-1	1	-1	1	1
16	1	1	1	1	-1	-1	0	-1	1	1	0	1	1	1	1	1	-1	-1	-1	1	0	1	-1
17	-1	1	1	-1	1	0	-1	-1	-1	-1	0	1	1	1	1	1	-1	0	-1	1	1	-1	-1
18	1	1	-1	1	1	0	-1	-1	-1	-1	0	1	1	1	1	-1	1	1	-1	1	1	-1	-1
19	1	1	1	-1	-1	0	-1	-1	-1	-1	0	1	1	1	1	1	-1	-1	1	1	-1	-1	1
20	-1	1	1	1	1	0	0	-1	-1	-1	0	-1	-1	-1	-1	1	-1	0	-1	1	0	-1	1
21	-1	1	1	1	1	0	0	-1	-1	-1	0	-1	1	-1	1	-1	1	1	-1	1	-1	-1	1
22	1	1	1	-1	-1	0	-1	-1	-1	-1	0	1	1	1	1	-1	1	-1	-1	1	0	-1	1
23	-1	1	1	-1	1	0	0	-1	1	1	0	1	1	1	1	1	1	0	-1	1	-1	1	1
24	-1	1	1	1	1	0	0	-1	1	1	0	1	1	1	1	1	1	1	-1	1	-1	1	1
25	-1	1	1	-1	-1	0	0	-1	1	1	0	1	1	1	1	1	1	-1	-1	1	0	1	1
26	-1	1	1	1	1	1	0	-1	1	1	0	1	1	1	1	-1	1	1	-1	1	0	1	1
27	1	1	1	-1	-1	-1	-1	-1	-1	-1	0	1	1	1	1	1	1	0	-1	1	-1	-1	-1
28	1	1	1	1	-1	1	0	-1	-1	-1	0	1	1	1	1	1	-1	0	-1	1	0	-1	1
29	-1	1	1	1	1	1	0	-1	1	1	1	1	1	1	1	-1	-1	1	-1	1	-1	1	1
30	-1	1	1	1	1	1	0	-1	1	1	0	1	1	1	1	-1	1	1	1	1	0	1	1
31	-1	1	1	-1	1	0	1	-1	1	1	0	1	1	1	1	-1	-1	-1	-1	1	0	1	1
32	-1	-1	1	-1	1	0	-1	-1	-1	-1	0	-1	1	-1	-1	1	-1	1	1	1	0	-1	1
33	-1	1	1	1	1	1	1	1	-1	-1	0	1	1	1	1	1	1	-1	-1	1	-1	-1	1
34	-1	1	1	1	1	0	1	-1	-1	-1	0	1	1	1	1	1	1	1	-1	1	0	-1	



	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
11023	-1	-1	-1	1	1	-1	0	-1	-1	1	0	-1	-1	-1	-1	1	1	1	-1	1	0	1	-1
11024	-1	1	1	1	1	0	0	-1	-1	1	0	1	1	1	1	1	1	1	-1	1	0	1	1
11025	-1	1	1	1	1	1	-1	1	1	1	0	1	1	1	1	1	1	1	-1	-1	1	1	1
11026	1	-1	1	1	-1	0	0	-1	1	1	0	-1	1	-1	1	1	1	1	1	-1	0	1	1
11027	1	1	1	1	-1	1	1	-1	1	1	0	1	1	1	1	1	1	1	-1	1	0	1	1
11028	-1	-1	-1	1	1	0	1	1	-1	-1	0	-1	1	-1	1	1	1	1	-1	1	0	1	1
11029	-1	1	1	1	1	0	-1	-1	1	1	0	1	1	1	1	-1	1	0	1	1	1	1	1
11030	-1	1	1	1	1	-1	-1	-1	1	1	0	1	1	1	1	1	1	1	-1	1	0	1	-1
11031	1	1	1	1	-1	-1	-1	-1	1	1	0	1	1	1	1	1	1	0	-1	1	1	1	-1
11032	-1	1	1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	-1	1	1	0	1	1
11033	1	1	1	1	-1	-1	0	-1	1	1	0	1	1	1	1	1	1	0	1	1	0	1	-1
11034	-1	-1	-1	1	-1	1	0	-1	-1	1	0	1	1	-1	-1	1	1	1	-1	1	0	1	1
11035	1	1	1	1	-1	0	-1	1	1	1	0	1	1	1	1	1	1	1	-1	-1	1	1	-1
11036	-1	1	1	1	1	0	1	-1	1	1	0	1	1	1	1	1	1	0	-1	1	0	1	1
11037	-1	1	1	-1	1	0	0	-1	1	-1	1	1	1	1	1	1	-1	0	-1	1	1	1	1
11038	-1	1	1	-1	1	-1	1	-1	1	-1	1	1	1	1	1	1	-1	0	-1	1	0	1	-1
11039	1	1	1	1	-1	-1	-1	0	1	1	0	1	1	1	1	1	1	1	-1	1	0	1	-1
11040	-1	1	1	1	1	0	-1	-1	1	1	0	1	1	1	1	1	1	1	-1	1	0	1	-1
11041	1	-1	-1	1	-1	-1	0	-1	-1	1	0	-1	1	-1	-1	1	1	0	-1	-1	1	1	-1
11042	-1	-1	-1	1	1	0	-1	0	-1	1	0	1	-1	-1	-1	1	1	1	-1	1	0	1	1
11043	-1	1	1	1	1	1	-1	1	1	1	0	1	1	1	1	1	1	1	-1	1	0	1	1
11044	-1	1	1	1	1	-1	0	-1	1	1	0	1	1	1	1	1	1	0	-1	1	0	1	-1
11045	-1	1	1	-1	1	1	0	-1	1	-1	1	1	1	1	1	1	-1	0	-1	1	1	1	1
11046	-1	1	1	1	1	0	1	-1	1	1	0	1	1	1	1	1	1	0	-1	1	0	1	1
11047	-1	1	1	1	-1	0	1	1	1	1	0	1	1	1	1	1	1	0	-1	1	1	1	1
11048	-1	1	1	-1	1	-1	-1	-1	-1	1	0	1	1	1	1	1	1	1	-1	1	0	1	-1
11049	1	1	1	1	-1	0	0	-1	1	1	0	1	1	1	1	1	1	0	-1	1	0	1	1
11050	1	1	1	-1	-1	0	-1	-1	1	1	1	1	1	1	1	-1	1	1	-1	1	1	1	-1
11051	-1	-1	-1	1	1	1	1	-1	-1	1	0	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1
11052	-1	-1	-1	1	1	-1	-1	0	-1	-1	1	-1	1	-1	1	1	1	1	1	1	-1	1	-1
11053	-1	1	1	1	1	0	-1	-1	1	1	0	1	1	1	1	1	1	1	-1	1	0	1	-1
11054	1	-1	1	1	-1	-1	1	-1	1	1	0	-1	1	-1	1	1	1	1	-1	1	1	1	-1
11055	1	1	1	1	-1	-1	0	-1	1	1	0	1	1	1	1	-1	1	-1	-1	-1	1	-1	-1

4. IMPLEMENTATION DETAILS

Implementation is the execution or practice of a plan, a method or any design, idea, model, specification, standard or policy for doing something. As such, implementation is the action that must follow any preliminary thinking for something to actually happen.

To provide as much flexibility as possible, the display application is implemented as multiple nearly- independent modules. Each module is responsible for the display of a particular kind of data and is implemented.

- analyzing requirements
- installation
- configuration
- customization
- testing
- running
- systems integrations
- user training
- delivery
- making changes

4.1 Software Environment

A software development environment (SDE) is the collection of hardware and software tools a system developer uses to build software systems. When you are developing software, you probably don't want your users to see every messy part of your application creation process.

There 4 different environments in a software development team are shown below:

Development environment.

Testing Environment

Staging environment

Production environment

4.1.1

Algorithm used-

Decision Tree Algorithm

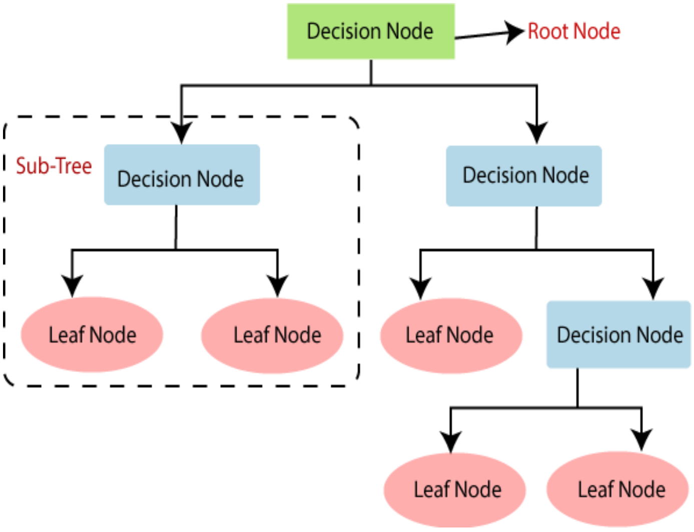
Decision Tree is a supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems.

It is a tree structured classifier where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

- **Parent/Child node**: The root node of the tree is called the parent node, and other nodes are called the child nodes.



- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.

✓ A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning. There are mainly two types of tree **pruning** technology used:

- **Cost Complexity Pruning**
- **Reduced Error Pruning.**

Adavntages of Decision Tree

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
- For more class labels, the computational complexity of the decision tree may increase.

4.1.2 Software Technologies

The software technology used in this project is python.

Python is the fastest growing programming language.It supports multiple programming paradigms,including structured,object-oriented and functional programming.And it is dynamically- typed and garbage collected.
It consistently ranks as one of the most popular programming languages.It can be also usedon a server to create web applications .
It has a huge number of libraries and frameworks.
Python frameworks are no different; they are a collection of modules and packages. These frameworks automate common processes and implementation. For instance, developers can focus on application logic rather than dealing with routinary processes.
The python libraries used are:

- numpy
- pandas
- matplotlib
- sklearn
- seaborn

Website designing-

- html
- bootstrap
- css
- Flask

Numpy-

The name “Numpy” stands for “Numerical Python”. It is the commonly used library. It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations. Even libraries like TensorFlow use Numpy internally toperform several operations on tensors. Array Interface is one of the key features of this library

Pandas-

pandas is a software library written for the Python programming language for data manipulation and analysis. When we have to work on Tabular data, we prefer the pandas module. The powerful tools of pandas are Data frame and Series. Pandas has a better performance when a number of rows is 500K or more

Matplotlib-

matplotlib() is a library function that is responsible for plotting numerical data. And that’s why it is used in data analysis. It is also an open-source library and plots high-defined figures like pie charts, histograms, scatterplots, graphs, etc.

Scikit-learn-

We use this as Scikit-learn (). It is a famous Python library to work with complex data. Scikit-learn isan open-source library that supports machine learning. It supports variously supervised and unsupervised algorithms like linear regression, classification, clustering, etc. This library works in association with NumPy and SciPy.

Seaborn-

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions. It is used for data visualization and exploratory data analysis. Seaborn works easily with data frames and the Pandas library. The graphs createdcan also be customized easily.

Flask-

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Html-

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page.

Bootstrap-

Bootstrap is a free, open source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap also provides a collection of syntax for template designs.Bootstrap framework is really amazing in creating beautiful and responsive websites, it makes your work easier and faster it has ready to use div, classes and every component and everything which is used to create beautiful and responsive websites that’s why we should use Bootstrap in our website.

CSS-

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML . CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Python Implementation of Decision Tree Algorithm

Here we will implement the Decision Tree Classifier algorithm using Python. For this, we will use the same dataset "phishing.csv", which we have used in classification models. By using the same dataset, we can compare the Decision Tree Classifier with other classification models such as KNN, SVM, Logistic Regression, etc.

Implementation Steps are given below:

- Data Pre-processing step
- Fitting the Decision Tree algorithm to the Training set
- Predicting the test result .
- Test accuracy of the result. (Creation of Confusionmatrix)
- Visualising the test result.
- Designing the interactive web page.

5.TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

The main purpose of testing is to discover errors and it is a process to check the functionality of components. It ensures that software systems meets its requirements and user expectations without fault .

Software Testing is an important element of quality assurance and represents ultimate view of specification and design.

Testing can also be defined as –

A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

When to Start Testing?

An early start to testing reduces the cost and time to rework and produce error-free software that is delivered to the client. However in Software Development Life Cycle (SDLC), testing can be started from the Requirements Gathering phase and continued till the deployment of the software.

It also depends on the development model that is being used. For example, in the Waterfall model, formal testing is conducted in the testing phase; but in the incremental model, testing is performed at the end of every increment/iteration and the whole application is tested at the end.

Testing is done in different forms at every phase of SDLC −

During the requirement gathering phase, the analysis and verification of requirements are also considered as testing.

Reviewing the design in the design phase with the intent to improve the design is also considered as testing

Testing performed by a developer on completion of the code is also categorized as testing.

When to Stop Testing?

It is difficult to determine when to stop testing, as testing is a never-ending process and no one can claim that a software is 100% tested. The following aspects are to be considered for stopping the testing process −

- Testing Deadlines
- Completion of test case execution
- Completion of functional and code coverage to a certain point

White Box Testing

White box testing is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. Implementation and impact of the code are tested.

It is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software.

- Code implementation is necessary for white box testing.
- It is mostly done by software developers.
- Knowledge of implementation is required.
- It is the inner or the internal software testing.
- It is the Structural test if the software.
- This type of testing is software is started after detail design document.

Black Box Testing

Black box testing is a software testing method in which the internal structure/design/implementationof the item being tested is not known to the tester. Only the external design and structure are tested.

It is a way of software testing in which the internal structure or the program or the code is hidden andnothing is known about it.

- Implementation of code is not needed for black box testing.
- It is mostly done by software testers.
- No knowledge of implementation is needed.
- It can be referred as outer or external software testing.
- It is functional test of the software.
- This testing can be initiated on the basis of requirement specifications document.

5.1 Test case table’s

Test case table-1

	Test	Requirements	Req no.	Essential	Description	Expected Output	Actual Output	Result
S.no	Case no.	(req)						
1	TC1	Input URl’s (dataset)	RS1	essential	Dataset is downloaded and then loaded	The dataset should be loaded without any errors	The dataset is loaded without any errors	SUCCESS

Test case table-2

	Test	Requirements	Req no.	Essential	Description	Expected Output	Actual Output	Result
S.no	Case no.	(req)						
2	TC2	Data Preprocessing	RS2	essential	Dataset is checked for null values, if any null values are found they should be handled	Data should be cleaned and should and should contain no null values	No null values are found	SUCCESS

Test case table-3

	Test	Requirements	Req no.	Essential	Description	Expected Output	Actual Output	Result
S.no	Case no.	(req)						
3	TC3	Feature Selection	RS3	essential	Feature selection is done using heatmap	Heatmap shows whether all columns are relevant to the target or not	Heatmap is generated and all the features are found to be significant	SUCCESS

Test case table-4

	Test	**Requirements**	**Req no.**	**Essential**	**Description**	**Expected Output**	**Actual Output**	**Result**
S.no	**Case no.**	**(req)**						
4	TC4	Training model	RS4	essential	Dataset is split into train and test data	According to the given ,the dataset should be split into train and test data.	The given dataset is split into train and test data .	SUCCESS

Test casetable-5

	Test	**Requirements**	**Req no.**	**Essential**	**Description**	**Expected Output**	**Actual Output**	**Result**
S.no	**Case no.**	**(req)**						
5	TC5	Decision Tree algorithm is called for given data set	RS5	essential	Decision Tree algorithm is applied	The model applied should work without any errors	Result is classified (yes/no)	SUCCESS

Test casetable-6

	Test	**Requirements**	**Req no.**	**Essential**	**Description**	**Expected Output**	**Actual Output**	**Result**
S.no	**Case no.**	**(req)**						
6	TC6	Evaluation of model	RS6	essential	Accuracy, precision are measured	evaluation properties are measured properly	All the properties are measured and compared	SUCCESS

Test case table-7

	Test	**Requirements**	**Req no.**	**Essential**	**Description**	**Expected Output**	**Actual Output**	**Result**
S.no	Case no.	(req)						
7	TC7	Detection system	RS7	essential	The detection system should be able to detect the valid and invalid URL’s correctly	Detection should be made correctly	Detection is made correctly for the given input	SUCCESS

5.CONCLUSION

1. The final take away from this project is to explore a machine learning model, performing exploratory Data Analysis on phishing dataset and understanding their features.
2. It is a user-friendly website that helps the user to check faster whether a particular URL is a legitimate URL or not by its attributes.
3. Creating this notebook helped us to learn a lot about the features affecting the model to detect whether URL is safe or not, also we came to know how to tune model and how they affect the model performance.
4. The final conclusion on the Phishing dataset is that the some feature like “HTTTPS”, “AnchorURL”, “Using @”, “IP address” ,etc have the importance to classify URL is phishing URL or not.
5. Decision Tree Classifier correctly classify URL upto 95.0% respective classes and hence reduces the chance of phishing attachments.

Future Scope

The days of basic phishing schemes have more or less passed. Attacks now rely on advanced forms of infiltration that better disguise fraud intent.

As Phishing has always had the aim of baiting users to take an action or share a piece of sensitive information by appearing as a non-threat, but awareness has since grown. Unprompted password reset emails, while once effective, no longer drive the same volume of user action and are often detected by spam filters.

No end for phishing that exists today, but research is ongoing and can often provide still more substantial improvements. It would be better if there is a still more easier way for the user to check the URL in a particular website.
Not only detecting through the limited features but also including all the possible features of a phishing URL as not every URL has every feature or only one.

References:

1. https://informationsecuritybuzz.com/the-future-of-phishing/

2. https://www.opensourceforu.com/2022/04/detect-a-phishing-url-using-machine-learning-in- python/

3. https://www.activestate.com/blog/phishing-url-detection-with-python-and-ml/

4.https://www.researchgate.net/publication/355263255_Detecting_phishing_websites_using_m achine_learning_technique

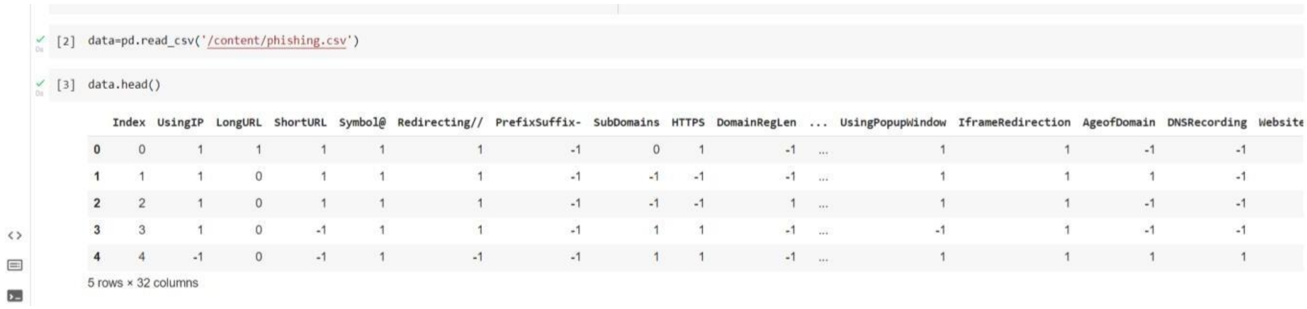
5. https://www.techtarget.com/searchnetworking/definition/URL

6 .https://csu-csus.esploro.exlibrisgroup.com/esploro/outputs/graduate/Detecting-malicious- shortened-URLs-using-machine/99257831032201671

7. https://ieeexplore.ieee.org/document/6565224

8. https://www.w3schools.com/bootstrap/

APPENDIX

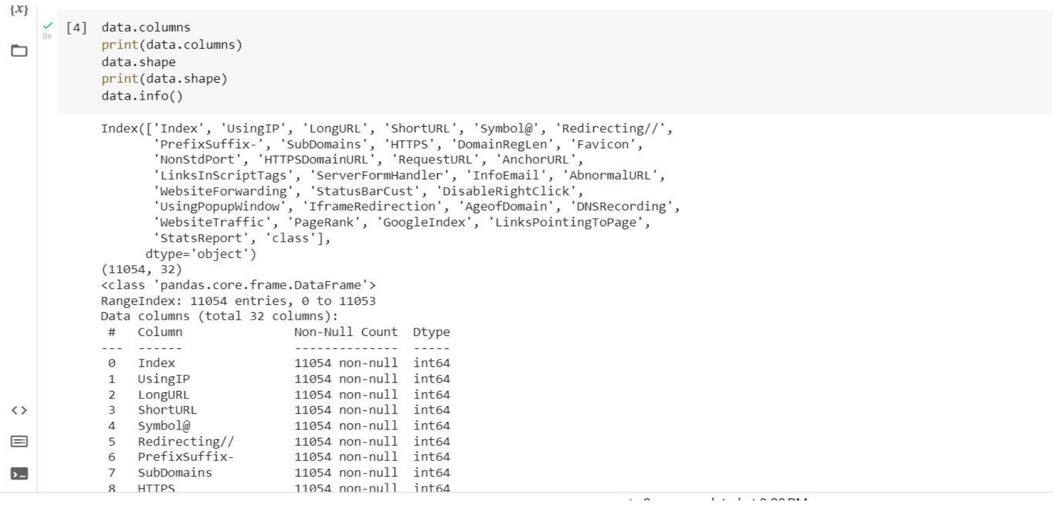


```
[2] data=pd.read_csv('/content/phishing.csv')
```

```
[3] data.head()
```

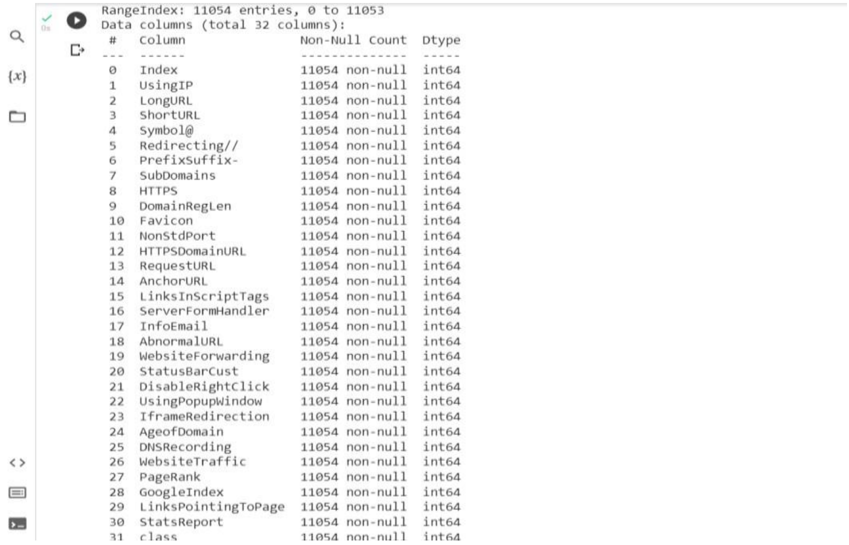
	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegLen	...	UsingPopupWindow	IframeRedirection	AgeofDomain	DNSRecording	Website
0	0	1	1	1	1	1	-1	0	1	-1	...	1	1	-1	-1	
1	1	1	0	1	1	1	-1	-1	-1	-1	...	1	1	1	-1	
2	2	1	0	1	1	1	-1	-1	-1	1	...	1	1	-1	-1	
3	3	1	0	-1	1	1	-1	1	1	-1	...	-1	1	-1	-1	
4	4	-1	0	-1	1	-1	-1	1	1	-1	...	1	1	1	1	

5 rows × 32 columns

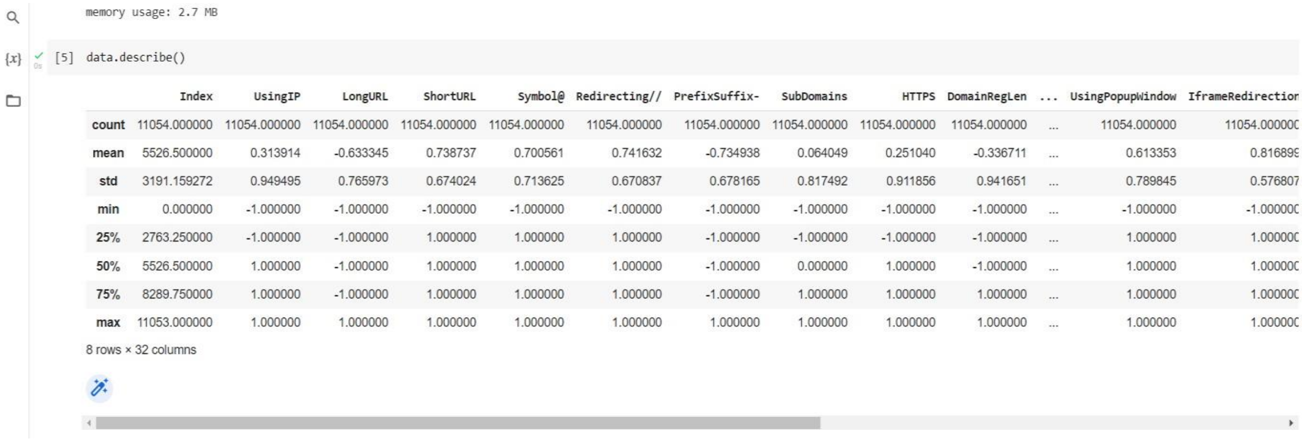


```
[4] data.columns
    print(data.columns)
    data.shape
    print(data.shape)
    data.info()
```

```
Index(['Index', 'UsingIP', 'LongURL', 'ShortURL', 'Symbol@', 'Redirecting//',
       'PrefixSuffix-', 'SubDomains', 'HTTPS', 'DomainRegLen', 'Favicon',
       'NonStdPort', 'HTTPSDomainURL', 'RequestURL', 'AnchorURL',
       'LinksInScriptTags', 'ServerFormHandler', 'InfoEmail', 'AbnormalURL',
       'WebsiteForwarding', 'StatusBarCust', 'DisableRightClick',
       'UsingPopupWindow', 'IframeRedirection', 'AgeofDomain', 'DNSRecording',
       'WebsiteTraffic', 'PageRank', 'GoogleIndex', 'LinksPointingToPage',
       'StatsReport', 'class'],
      dtype='object')
(11054, 32)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11054 entries, 0 to 11053
Data columns (total 32 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Index               11054 non-null  int64
 1   UsingIP             11054 non-null  int64
 2   LongURL             11054 non-null  int64
 3   ShortURL            11054 non-null  int64
 4   Symbol@             11054 non-null  int64
 5   Redirecting//       11054 non-null  int64
 6   PrefixSuffix-       11054 non-null  int64
 7   SubDomains          11054 non-null  int64
 8   HTTPS               11054 non-null  int64
```



```
RangeIndex: 11054 entries, 0 to 11053
Data columns (total 32 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Index                11054 non-null  int64
 1   UsingIP              11054 non-null  int64
 2   LongURL              11054 non-null  int64
 3   ShortURL             11054 non-null  int64
 4   Symbol@              11054 non-null  int64
 5   Redirecting//        11054 non-null  int64
 6   PrefixSuffix-        11054 non-null  int64
 7   SubDomains           11054 non-null  int64
 8   HTTPS                11054 non-null  int64
 9   DomainRegLen         11054 non-null  int64
 10  Favicon              11054 non-null  int64
 11  NonStdPort           11054 non-null  int64
 12  HTTPSDomainURL       11054 non-null  int64
 13  RequestURL           11054 non-null  int64
 14  AnchorURL            11054 non-null  int64
 15  LinksInScriptTags    11054 non-null  int64
 16  ServerFormHandler    11054 non-null  int64
 17  InfoEmail            11054 non-null  int64
 18  AbnormalURL          11054 non-null  int64
 19  WebsiteForwarding    11054 non-null  int64
 20  StatusBarCust        11054 non-null  int64
 21  DisableRightClick    11054 non-null  int64
 22  UsingPopupWindow     11054 non-null  int64
 23  IframeRedirection    11054 non-null  int64
 24  AgeofDomain          11054 non-null  int64
 25  DNSRecording         11054 non-null  int64
 26  WebsiteTraffic       11054 non-null  int64
 27  PageRank             11054 non-null  int64
 28  GoogleIndex          11054 non-null  int64
 29  LinksPointingToPage  11054 non-null  int64
 30  StatsReport          11054 non-null  int64
 31  class                11054 non-null  int64
```

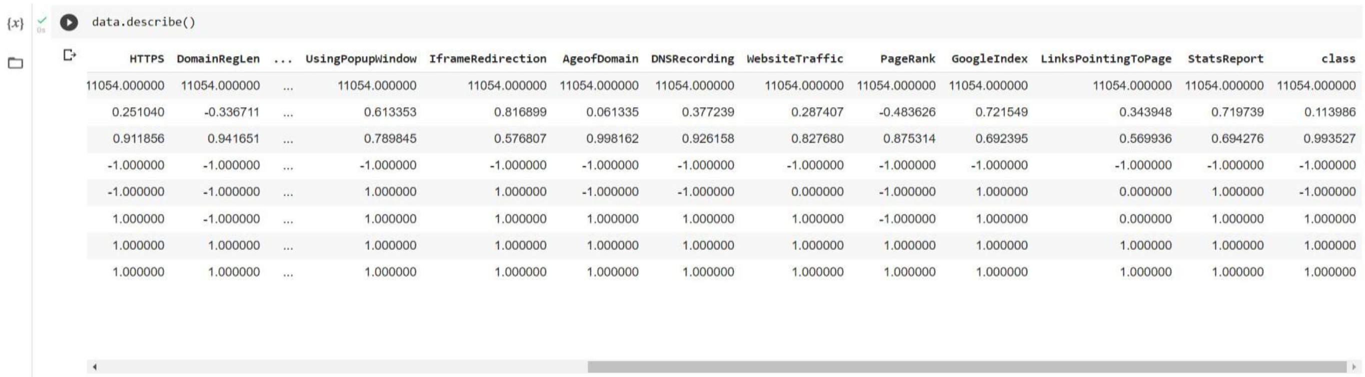


```
memory usage: 2.7 MB
```

```
[5] data.describe()
```

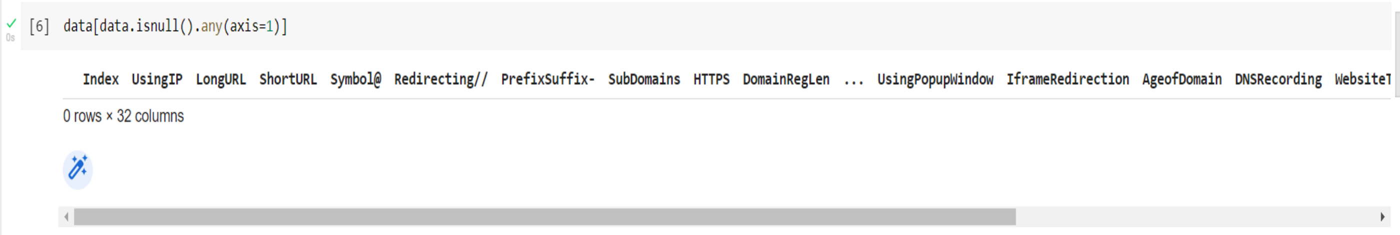
	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegLen	...	UsingPopupWindow	IframeRedirectior
count	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	...	11054.000000	11054.00000C
mean	5526.500000	0.313914	-0.633345	0.738737	0.700561	0.741632	-0.734938	0.064049	0.251040	-0.336711	...	0.613353	0.816899
std	3191.159272	0.949495	0.765973	0.674024	0.713625	0.670837	0.678165	0.817492	0.911856	0.941651	...	0.789845	0.576807
min	0.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	...	-1.000000	-1.00000C
25%	2763.250000	-1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	-1.000000	-1.000000	-1.000000	...	1.000000	1.00000C
50%	5526.500000	1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	0.000000	1.000000	-1.000000	...	1.000000	1.00000C
75%	8289.750000	1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	1.000000	1.000000	1.000000	...	1.000000	1.00000C
max	11053.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.00000C

8 rows × 32 columns



```
data.describe()
```

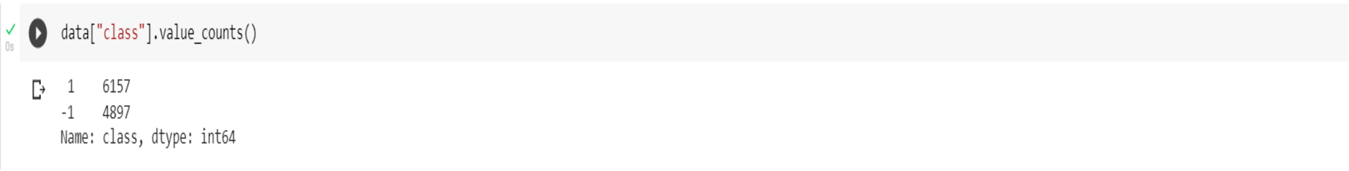
HTTPS	DomainRegLen	...	UsingPopupWindow	IframeRedirection	AgeofDomain	DNSRecording	WebsiteTraffic	PageRank	GoogleIndex	LinksPointingToPage	StatsReport	class
11054.000000	11054.000000	...	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000
0.251040	-0.336711	...	0.613353	0.816899	0.061335	0.377239	0.287407	-0.483626	0.721549	0.343948	0.719739	0.113986
0.911856	0.941651	...	0.789845	0.576807	0.998162	0.926158	0.827680	0.875314	0.692395	0.569936	0.694276	0.993527
-1.000000	-1.000000	...	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
-1.000000	-1.000000	...	1.000000	1.000000	-1.000000	-1.000000	0.000000	-1.000000	1.000000	0.000000	1.000000	-1.000000
1.000000	-1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000	-1.000000	1.000000	0.000000	1.000000	1.000000
1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000



```
[6] data[data.isnull().any(axis=1)]
```

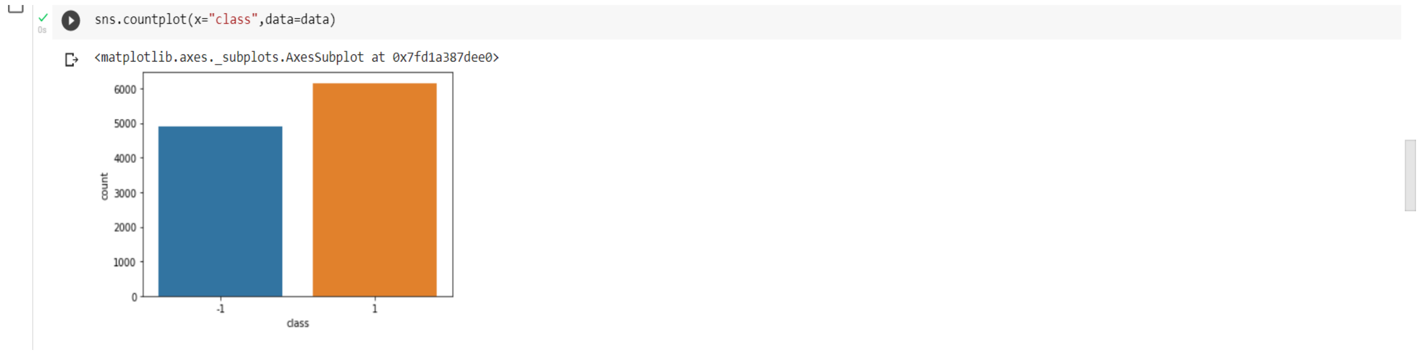
Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegLen	...	UsingPopupWindow	IframeRedirection	AgeofDomain	DNSRecording	WebsiteT

0 rows × 32 columns



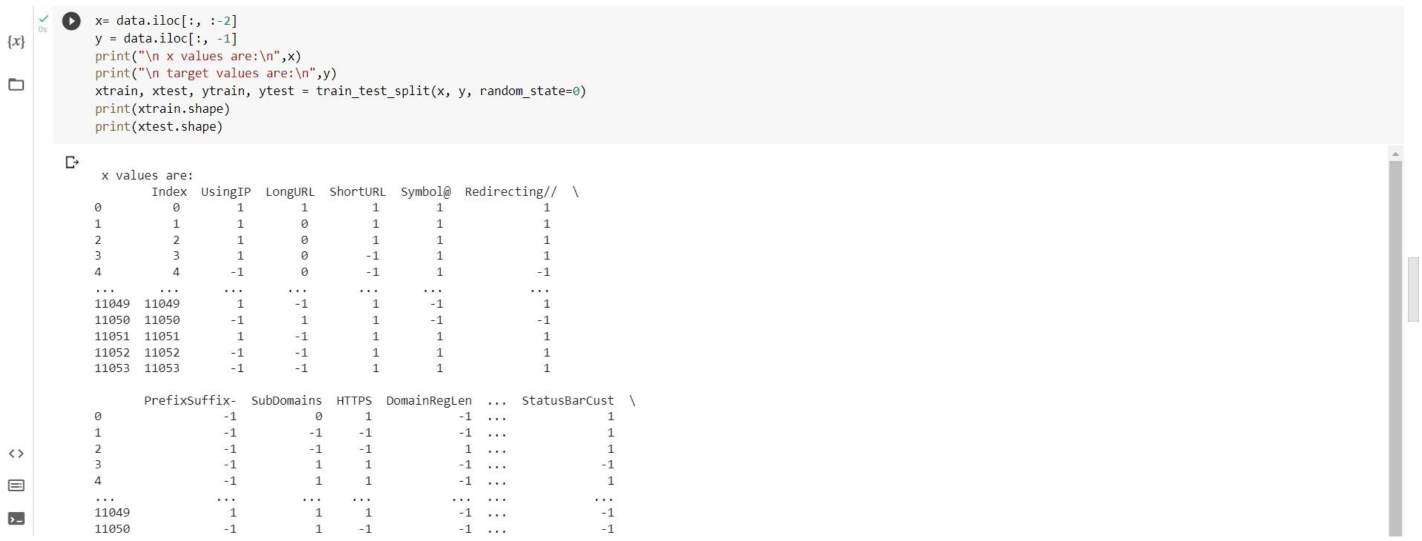
```
data["class"].value_counts()
```

```
 1    6157
-1    4897
Name: class, dtype: int64
```



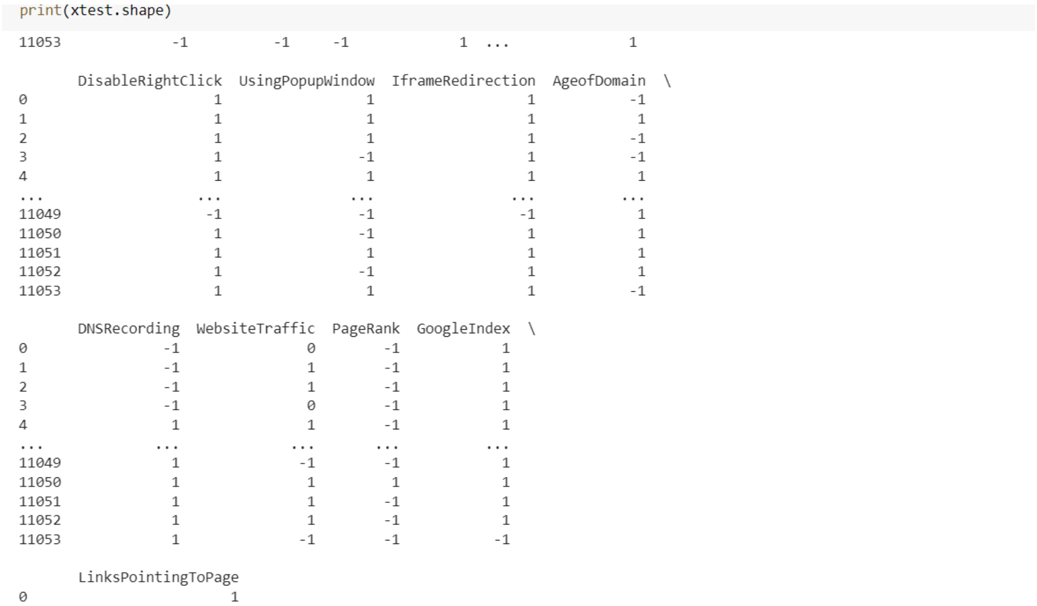
```
sns.countplot(x="class",data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd1a387dee0>
```



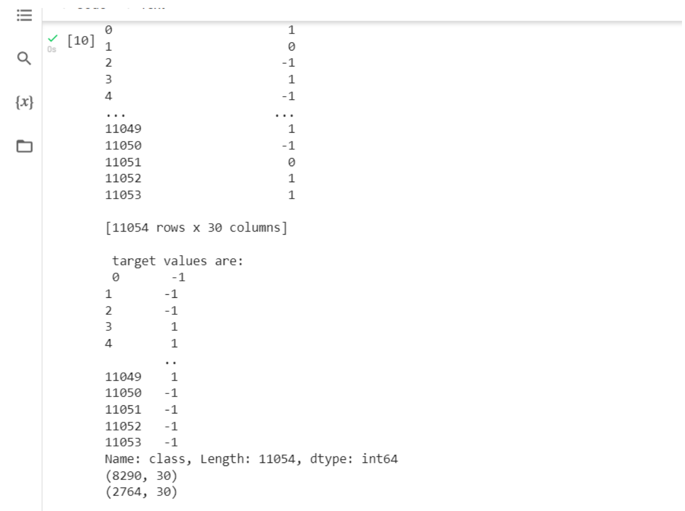
```
x= data.iloc[:, :-2]
y = data.iloc[:, -1]
print("\n x values are:\n",x)
print("\n target values are:\n",y)
xtrain, xtest, ytrain, ytest = train_test_split(x, y, random_state=0)
print(xtrain.shape)
print(xtest.shape)
```

```
 x values are:
        Index  UsingIP  LongURL  ShortURL  Symbol@  Redirecting//  \
0          0        1        1         1        1              1
1          1        1        0         1        1              1
2          2        1        0         1        1              1
3          3        1        0        -1        1              1
4          4       -1        0        -1        1             -1
...      ...      ...      ...       ...      ...            ...
11049  11049        1       -1         1       -1              1
11050  11050       -1        1         1       -1             -1
11051  11051        1       -1         1        1              1
11052  11052       -1       -1         1        1              1
11053  11053       -1       -1         1        1              1

       PrefixSuffix-  SubDomains  HTTPS  DomainRegLen  ...  StatusBarCust  \
0                 -1           0      1            -1  ...              1
1                 -1          -1     -1            -1  ...              1
2                 -1          -1     -1             1  ...              1
3                 -1           1      1            -1  ...             -1
4                 -1           1      1            -1  ...              1
...              ...         ...    ...           ...  ...            ...
11049              1           1      1            -1  ...             -1
11050             -1           1     -1            -1  ...             -1
```

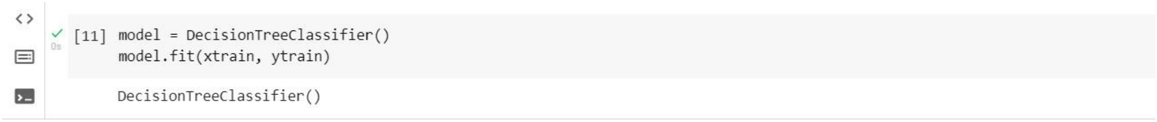


```
print(xtest.shape)
```

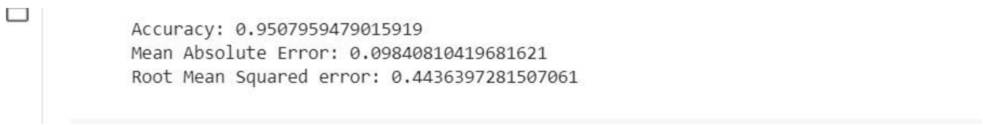
```
11053             -1          -1     -1             1  ...              1

       DisableRightClick  UsingPopupWindow  IframeRedirection  AgeofDomain  \
0                      1                 1                  1           -1
1                      1                 1                  1            1
2                      1                 1                  1           -1
3                      1                -1                  1           -1
4                      1                 1                  1            1
...                  ...               ...                ...          ...
11049                 -1                -1                 -1            1
11050                  1                -1                  1            1
11051                  1                 1                  1            1
11052                  1                -1                  1            1
11053                  1                 1                  1           -1

       DNSRecording  WebsiteTraffic  PageRank  GoogleIndex  \
0                -1               0        -1            1
1                -1               1        -1            1
2                -1               1        -1            1
3                -1               0        -1            1
4                 1               1        -1            1
...             ...             ...       ...          ...
11049             1              -1        -1            1
11050             1               1         1            1
11051             1               1        -1            1
11052             1               1        -1            1
11053             1              -1        -1           -1

       LinksPointingToPage
0                        1
```



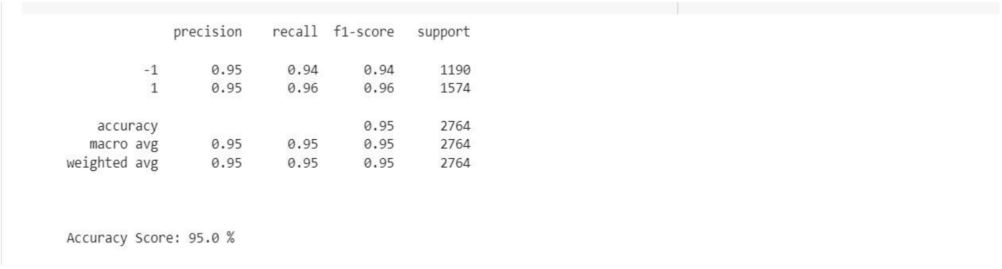
```
[10] 0                       1
     1                       0
     2                      -1
     3                       1
     4                      -1
     ...                   ...
     11049                   1
     11050                  -1
     11051                   0
     11052                   1
     11053                   1

     [11054 rows x 30 columns]

      target values are:
      0        -1
     1        -1
     2        -1
     3         1
     4         1
              ..
     11049     1
     11050    -1
     11051    -1
     11052    -1
     11053    -1
     Name: class, Length: 11054, dtype: int64
     (8290, 30)
     (2764, 30)
```



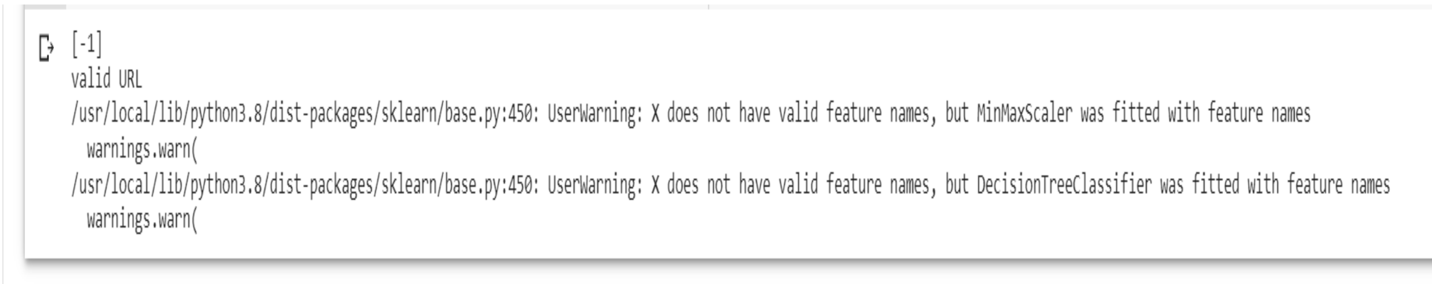
```
[11] model = DecisionTreeClassifier()
     model.fit(xtrain, ytrain)

     DecisionTreeClassifier()
```



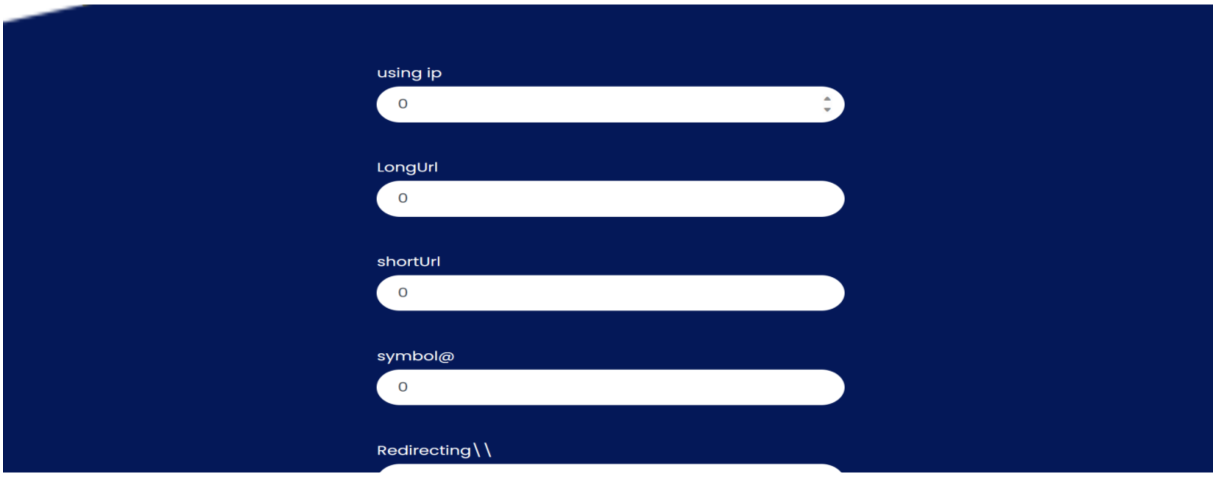
```
Accuracy: 0.9507959479015919
Mean Absolute Error: 0.09840810419681621
Root Mean Squared error: 0.4436397281507061
```



```
              precision    recall  f1-score   support

          -1       0.95      0.94      0.94      1190
           1       0.95      0.96      0.96      1574

    accuracy                           0.95      2764
   macro avg       0.95      0.95      0.95      2764
weighted avg       0.95      0.95      0.95      2764



Accuracy Score: 95.0 %
```



```
[-1]
valid URL
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but MinMaxScaler was fitted with feature names
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



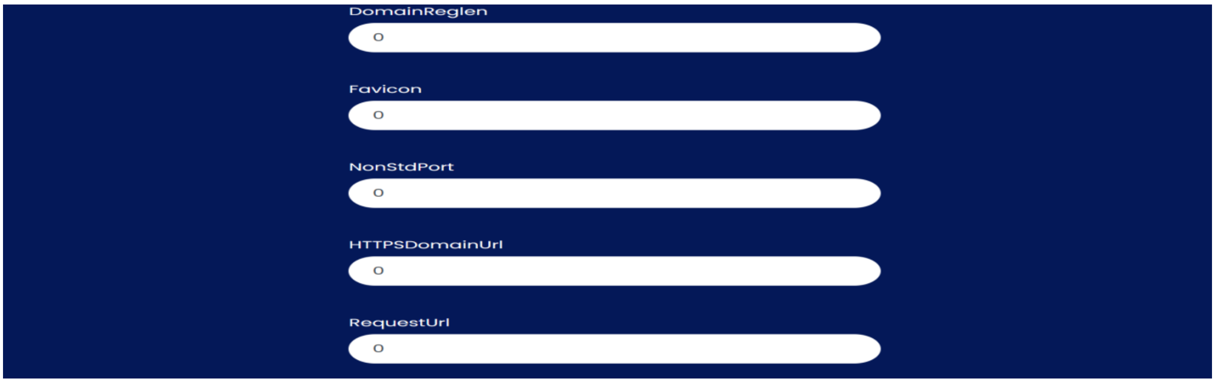
using ip
0

LongUrl
0

shortUrl
0

symbol@
0

Redirecting\\



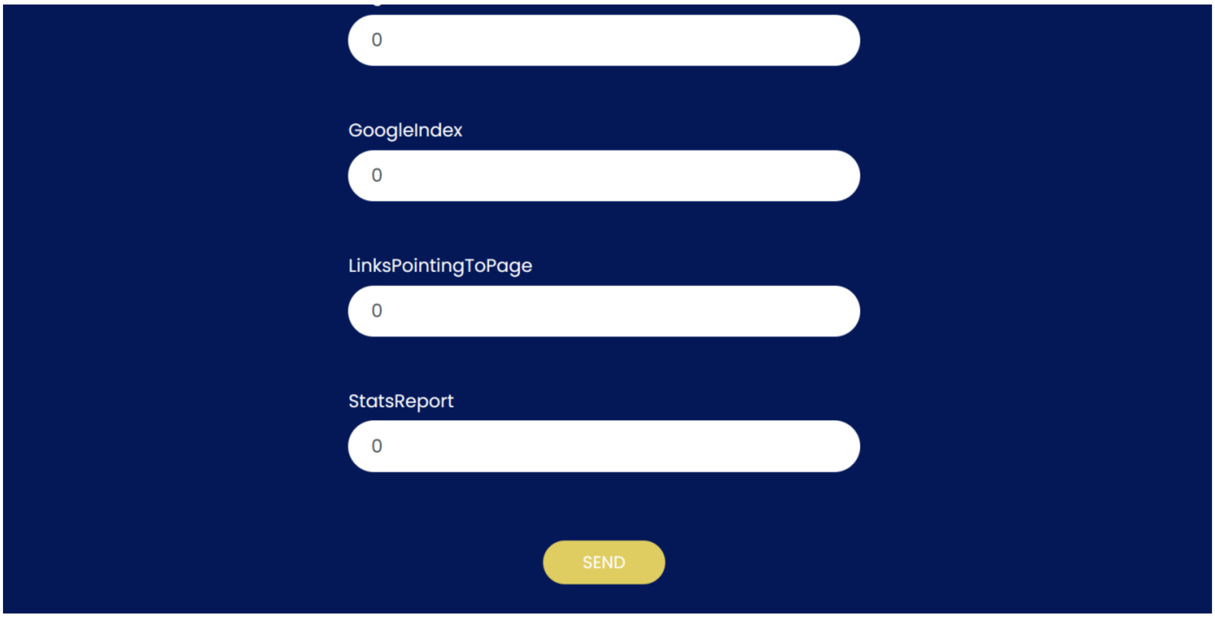
DomainReglen
0

Favicon
0

NonStdPort
0

HTTPSDomainUrl
0

RequestUrl
0

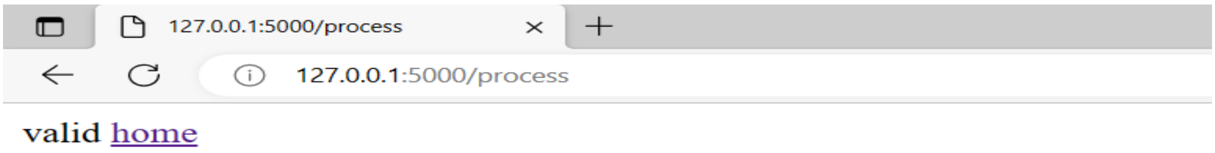


0

GoogleIndex
0

LinksPointingToPage
0

StatsReport
0



127.0.0.1:5000/process

valid home

Sample code:

```
from flask import *
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.metrics import accuracy_score,mean_absolute_error,mean_squared_error
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import tree
from numpy import array
from sklearn.preprocessing import MinMaxScaler

app = Flask(__name__)

@app.route('/')
def index():
        return render_template('index.html')

@app.route('/process',methods = ['POST','GET'])
def process():
        # all data comes here
        # ui=request.form('ui')
        ui=request.form['ui']
        lu=request.form['lu']
        su=request.form['su']
        sa=request.form['sa']
        rs=request.form['rs']
        ps=request.form['ps']
        sd=request.form['sd']
        ht=request.form['ht']
        dl=request.form['dl']
        Fn=request.form['Fn']
        nsp=request.form['nsp']
        htdou=request.form['htdou']
        Ru=request.form['Ru']
        Au=request.form['Au']
        lsct=request.form['lsct']
        sfh=request.form['sfh']
        inem=request.form['inem']
        abu=request.form['abu']
        wf=request.form['wf']
        sbc=request.form['sbc']
        Drc=request.form['Drc']
        upopw=request.form['upopw']
```

```
    ifdi=request.form['ifdi']
    ado=request.form['ado']
    DNSre=request.form['DNSre']
    WT=request.form['WT']
    pr=request.form['pr']
    gi=request.form['gi']
    lptpa=request.form['lptpa']
    SR=request.form['SR']
    print(ui, wf)
    data=pd.read_csv(r"C:\Users\hp\Downloads\phishing.csv")
    data.head()
    data.columns
    print(data.columns)
    data.shape
    print(data.shape)
    data.info()

    data.describe()
    print (data.describe() )
    print( 'nulls => ', data[data.isnull().any(axis=1)])

    data["class"].value_counts()

    sns.countplot(x="class",data=data)

    features=data.drop(['Index','class'],axis=1)
    target=data.loc[:,'class']
    x= data.iloc[:, :-2]
    y = data.iloc[:, -1]
    print("\n x values are:\n",x)
    print("\n target values are:\n",y)
    xtrain, xtest, ytrain, ytest = train_test_split(x, y, random_state=0)
    print(xtrain.shape)
    print(xtest.shape)
    model = DecisionTreeClassifier()
    model.fit(xtrain, ytrain)
    scaler=MinMaxScaler((-1,1))
    features_c=scaler.fit_transform(features)

input_data=(ui,lu,su,sa,rs,ps,sd,ht,dl,Fn,nsp,htdou,Ru,Au,lsct,sfh,inem,abu,wf,sbc,Drc,upopw,ifdi,ado,DNSre,WT,pr,gi,lptpa,SR)
    print(input_data)
    input_data_as_numpy_array=np.asarray(input_data)
    input_data_reshaped=input_data_as_numpy_array.reshape(1,-1)
    std_data=scaler.transform(input_data_reshaped)
    prediction=model.predict(std_data)
    print(prediction)
```

```
          if (prediction[-1]==-1):
                  print("valid URL")
                  return "valid <a href='/'>home</a>"
          else:
                  print("Invalid URL")
                  return "Invalid <a href='/'>home</a>"
                  # predict logic

  app.run()
```