BESANT TECHNOLOGIES

DATA ANALYSIS PROJECT

Topic: Phishing Detection Using Machine Learning

SUBMITTED BY:

NAME: S R LEKHANA

PH.NO: 8971818272

EMAIL ID: lekhanasr1811@gmail.com

UNDER THE GUIDANCE OF

PRIYANKA B G

CONTENTS

1. Introduction

2. Problem Statement

3. Objectives of the Analysis

4. Tools and Technologies

5. Dataset /overview

6. Methodology

    6.1: Data Collection and Import

    6.2: Data Cleaning and Preparation

    6.3: Exploratory Data Analysis (EDA)

    6.4: Data Visualization

7. Data Visualisation & Insights

8. Conclusion

9. References

1. Introduction

Phishing is a cyber-crime where attackers create fake websites or links to steal sensitive information like passwords and banking details. These phishing pages often look similar to trusted websites, so users may not easily identify the risk.

Machine Learning helps detect phishing more accurately by learning patterns from website features. In this project, a phishing-legitimate URL dataset is used, which includes different characteristics of websites such as redirects, number of digits in the URL, and usage of suspicious external resources.

The goal of this project is to build a machine learning model that can classify websites as phishing or legitimate and help protect users from online attacks.

2. Problem Statement

Phishing attacks are one of the most common cybercrimes where attackers trick users into sharing sensitive information by disguising fraudulent webpages as legitimate ones. Even trained users can be deceived due to sophisticated phishing techniques. To strengthen cybersecurity defences, this study aims to analyze phishing website characteristics and build an automated classification model to detect phishing webpages.

Using the **Phishing Websites Dataset** (10,000 samples: 5000 phishing + 5000 legitimate, with 48 extracted features), we will determine which features are most predictive and design a model capable of identifying phishing attempts in real-time.

3. Objective of the Analysis

- To analyze the behavioural and structural differences between legitimate and phishing websites
- To identify the most influential features that indicate phishing attacks
- To build a robust machine learning classifier that distinguishes phishing webpages from legitimate ones
- To evaluate multiple ML algorithms and select the best performing model
- To support cybersecurity policies and help build automated anti-phishing systems

Key Questions / KPIs

1. Which features of a webpage most strongly indicate phishing behaviour? (URL length, HTTPS usage, Domain age, use of special characters, etc.)
2. Which class is more common: Phishing or Legitimate?
3. Does URL length correlate with phishing attacks?
4. Are HTTPS certificates always safe?
5. How well does the model detect phishing without many false alarms?

Business Outcomes

- Automatically detect phishing websites to prevent cyber-fraud
- Protect user data such as banking and login information
- Reduce financial losses caused by phishing attacks
- Improve security systems by identifying risky website features

4. Tools and Technologies

Software and Tools

Tool	**Purpose**
Python	The main programming language used for analysis and visualization.
Jupyter Notebook	Provides an interactive environment to write, execute, and visualize Python code.
Microsoft Word	Used for preparing and formatting the final project report.
CSV File (Dataset Format)	Contains the dataset used for analysis, stored in tabular form.

Python Libraries Used

Library	**Description**
pandas	Used for data manipulation, cleaning, and analysis (handling missing values, filtering, grouping).
numpy	Provides mathematical and numerical operations essential for data computation.

matplotlib	A visualization library for creating static, animated, and interactive graphs.
seaborn	Built on top of matplotlib, it simplifies complex data visualization with enhanced styling and statistical graphics.

5. Dataset Overview

This dataset consists of **10,000 website URLs**, including **5,000 phishing** websites and **5,000 legitimate** websites. The data was collected during two periods: January–May 2015 and May–June 2017 to ensure variation in phishing techniques over time.

A total of 48 engineered features is extracted from each webpage, covering URL structure, domain properties, content behaviour, JavaScript activities, and security indicators. Feature extraction was performed using a browser automation-based approach (Selenium WebDriver), which provides more reliable and realistic results compared to traditional string parsing methods. Selenium ensures that dynamic webpage elements, scripts, and redirections are accurately captured.

Key Features:

Column Name	**Description**
CLASS_LABEL	Indicates whether the URL is Phishing (1) or Legitimate (0)
UrlLength	Total length of the entire URL string
NumDots	Number of dot (".") characters in the URL — indicates subdomain usage
SubdomainLevel	Number of subdomain segments before the main domain
NumDash	Count of dashes ("-") in the URL — often used to mimic legitimate domains
NumAtSymbol	Count of "@" symbols — used to mislead users into fake redirects
NumNumericChars	Number of digits in the URL — excessive use may indicate phishing
HTTPSecure	Shows if the page uses HTTPS security (1 = Yes, 0 = No)

RightClickDisabled	Whether right-click is blocked by JavaScript (phishing tactic)
PopupWindow	Checks if pop-up windows are triggered — often used for credential theft
IframeOrFrame	Presence of iframe/frame elements — can embed malicious content
NumQueryComponenets	Number of parameters after “?” in the URL — high count indicates manipulation
DomainInSubdomains	If the legitimate brand name is only in the subdomain (phishing sign)
NumRedirect	How many times the webpage redirects before loading — suspicious behaviour
UrlOfAnchor	Checks if anchor tags (<a>) point to external or deceptive URLs

6. Methodology

The methodology adopted for the *Phishing Website Detection* project focuses on applying **Exploratory Data Analysis (EDA)**, **feature selection**, and **machine learning model development** to classify webpages as *phishing* or *legitimate*. The workflow includes structured steps from importing the dataset to evaluating the model performance using Python-based tools such as Pandas, Scikit-learn, Matplotlib, and Seaborn.

Step 1: Data Collection and Import

The dataset used in this project contains **10,000 URL records** with **48 extracted features** representing webpage characteristics. The data was imported into Python using the pandas library for further analysis.

```
import pandas as pd
data = pd.read_csv('Phishing_Legitimate_full.csv')
data.head()
```

Step 2: Data Cleaning and Preparation

Data preprocessing operations were performed to ensure quality and consistency:

- Handling missing or null values using imputation or removal
- Converting data types where necessary
- Removing duplicate records
- Normalizing/Scaling numerical features for modelling
- Encoding categorical variables for ML compatibility
- Splitting the dataset into **training** and **testing** subsets

```
data.info()
data.isnull().sum()
```

Step 3: Exploratory Data Analysis (EDA)

EDA was conducted to understand data distribution and detect patterns between features and phishing behaviour.

Key areas analysed:

- URL structure indicators (length, dots, subdomains)
- Special characters in URLs ("@", "-", "_")
- Security indicators (HTTPS usage, redirects)
- Suspicious behaviours (disabled right-click, popups, iframe usage)

Statistical and visual techniques used:

```
data.describe()
data['CLASS_LABEL'].value_counts()
```

Step 4: Data Visualization

To interpret feature influence effectively, multiple plots were used:

- **Box Plots**: Outlier detection
- **Count Plots**: Binary behaviour features
- **Correlation Heatmap**: Multivariate relationships
- **Scatter/Strip Plots**: Numeric feature interaction

Libraries Used:

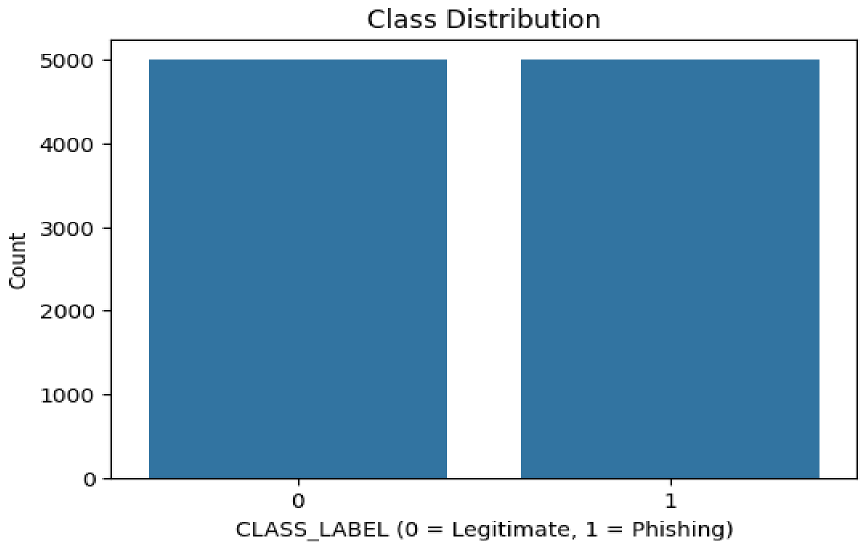
```
import matplotlib.pyplot as plt
import seaborn as sns
```

7. Data Visualization & Insights

1.What is the class distribution (Phishing vs Legitimate)?

Insight:

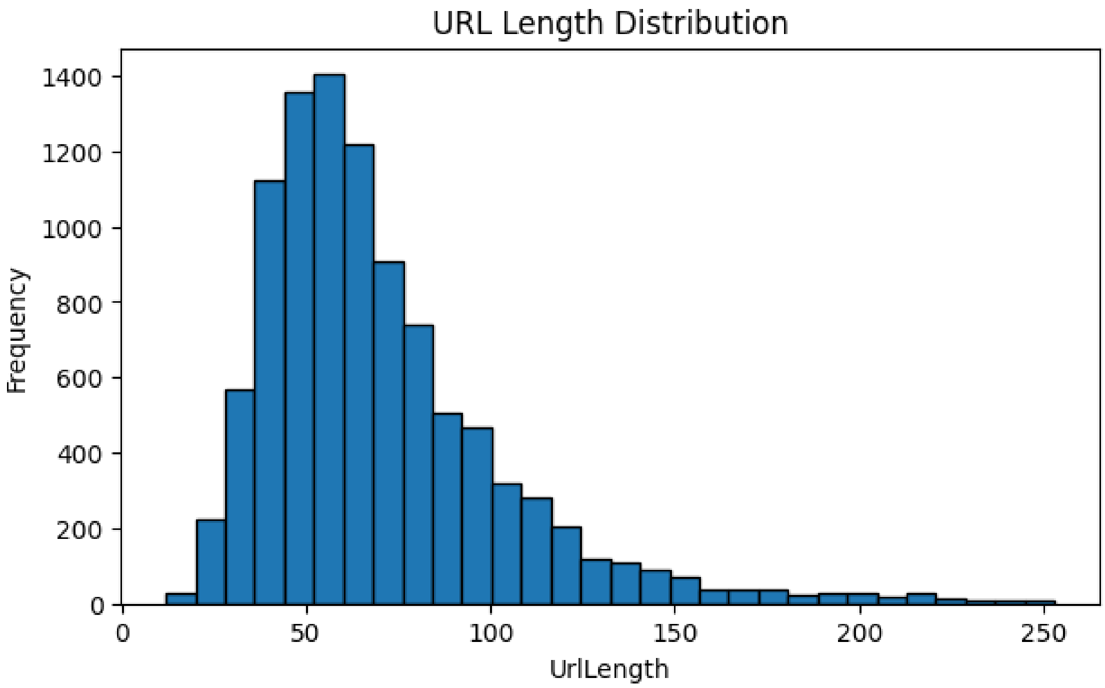
Phishing and legitimate websites are almost equally represented in this dataset.
A balanced dataset helps machine learning models learn both behaviours fairly without bias toward one class.



2.What is the distribution of URL Length?

Insight:

Most URLs in the dataset have shorter lengths, showing that both phishing and legitimate sites usually avoid long URLs. A few URLs are extremely long, which may indicate phishing attempts where attackers add extra hidden components.

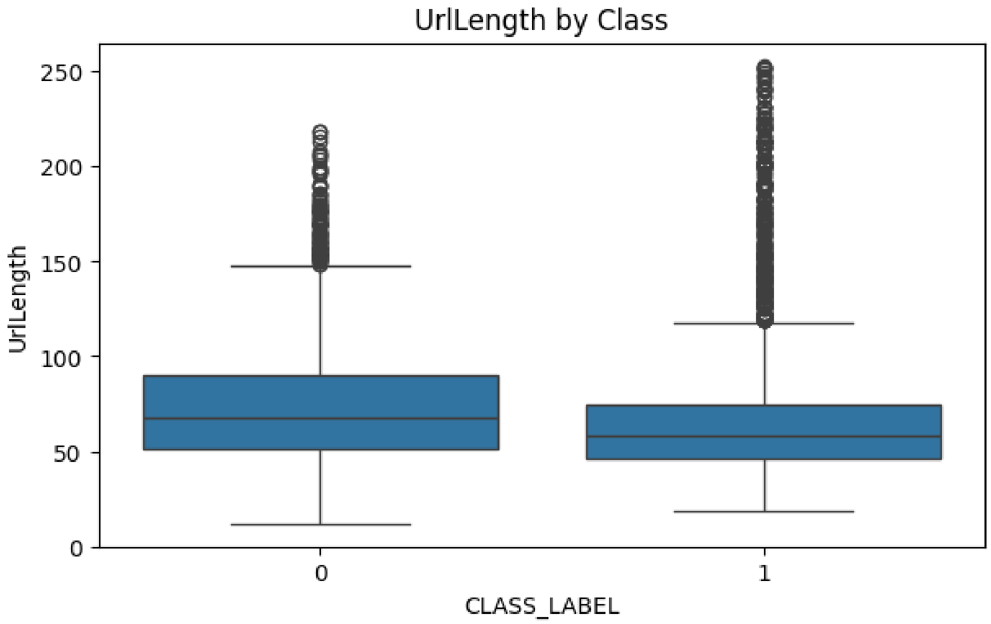


3. How does URL Length vary between classes?

Insight:

Phishing URLs are generally longer because attackers add extra paths, subdomains, or parameters to hide malicious intentions.

Legitimate URLs stay shorter and more structured, making them easier for users to recognize.

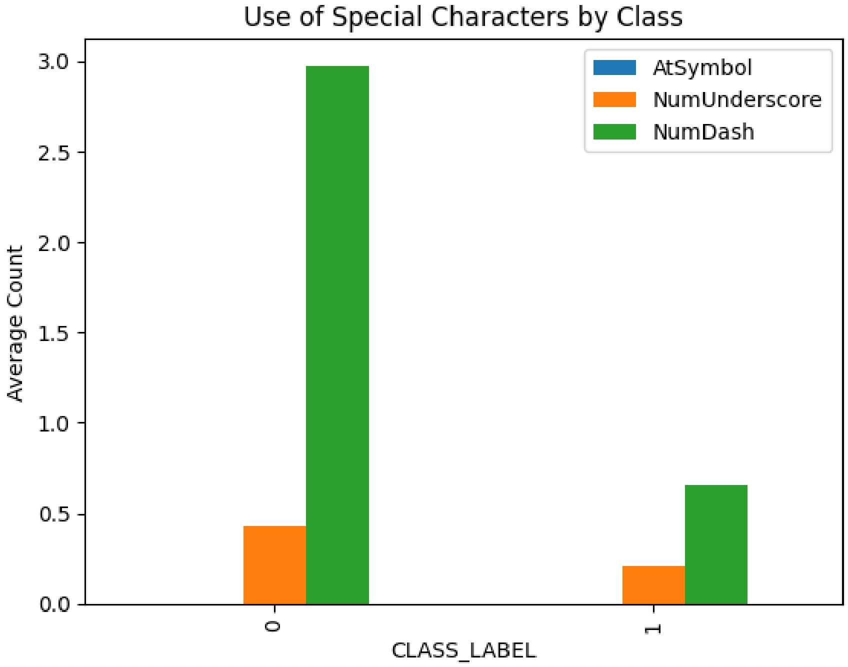


4. How do special characters appear in both classes?

Insight:

Phishing URLs tend to include more special characters like symbols and dashes to create misleading or complex links.

Legitimate sites keep their URLs simple because recognizable structure builds trust for users.

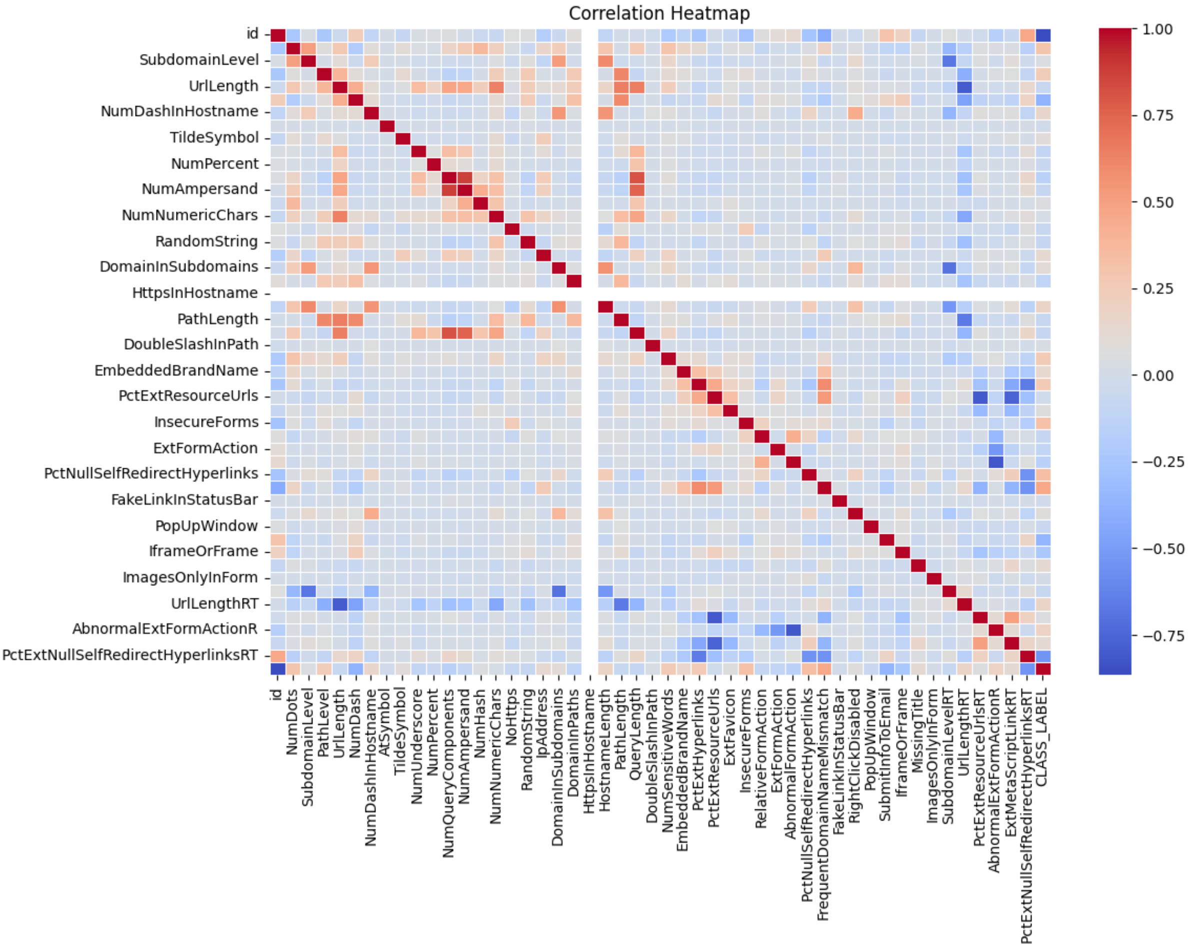


5.Which features are highly correlated with CLASS_LABEL?

Insight:

Several URL-based features show a strong correlation with the phishing class, meaning they play a key role in distinguishing malicious links. These correlated attributes improve model learning by capturing consistent phishing patterns.

The heatmap helps identify which features are strongly linked with phishing behaviour. Features that show stronger color intensity can be considered more important for training accurate ML models.

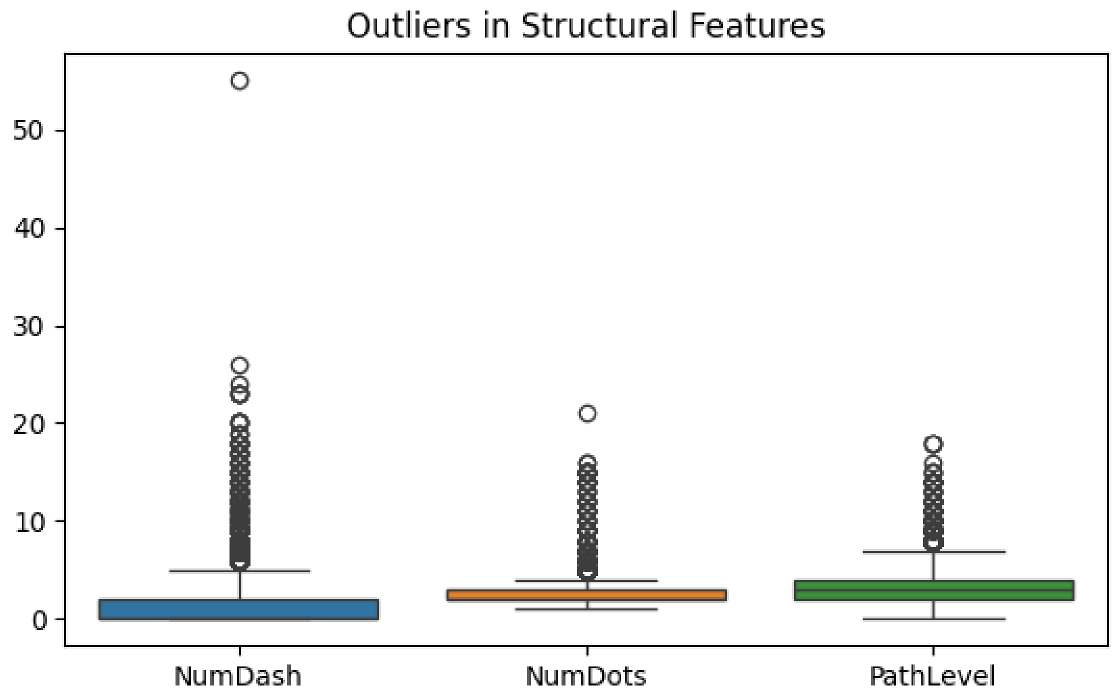


6. Which structural features have noticeable outliers?

Insight:

Some URLs show unusually high numbers of dashes, dots, and deeper path levels, indicating suspicious structural manipulation.

These outliers are often crafted to confuse users and hide the true destination of phishing links.

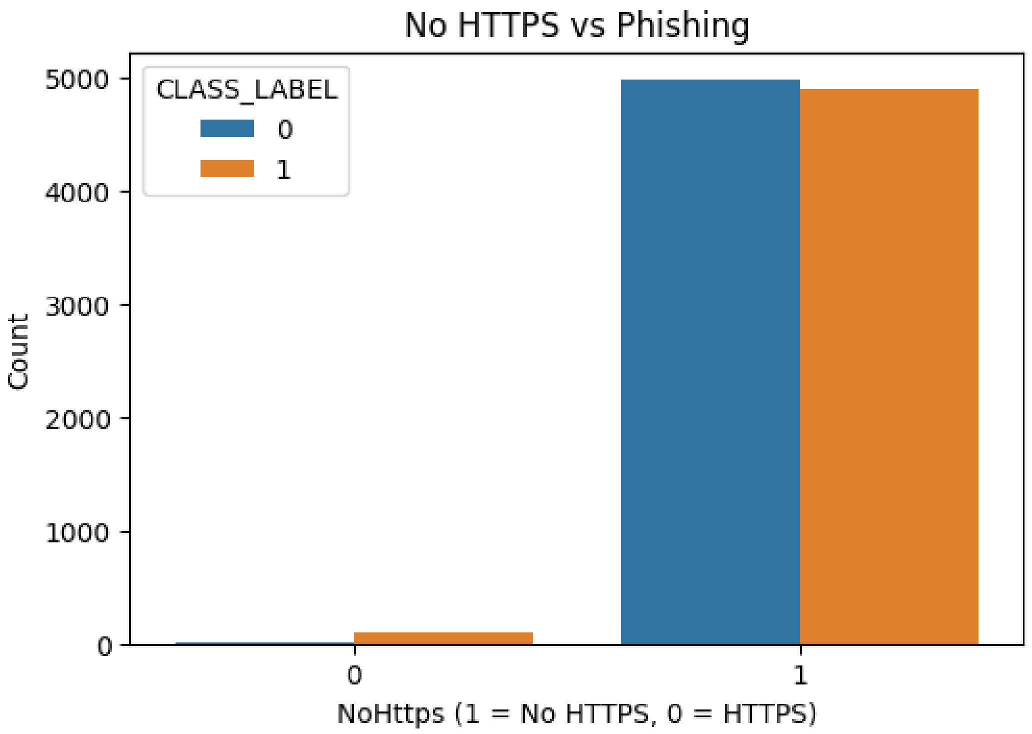


7. Is HTTPS more common in legitimate URLs?

Insight:

Websites without HTTPS are more frequently associated with phishing since attackers skip SSL certificates to reduce cost and effort.

Legitimate websites typically use HTTPS to ensure security and user trust during communication.

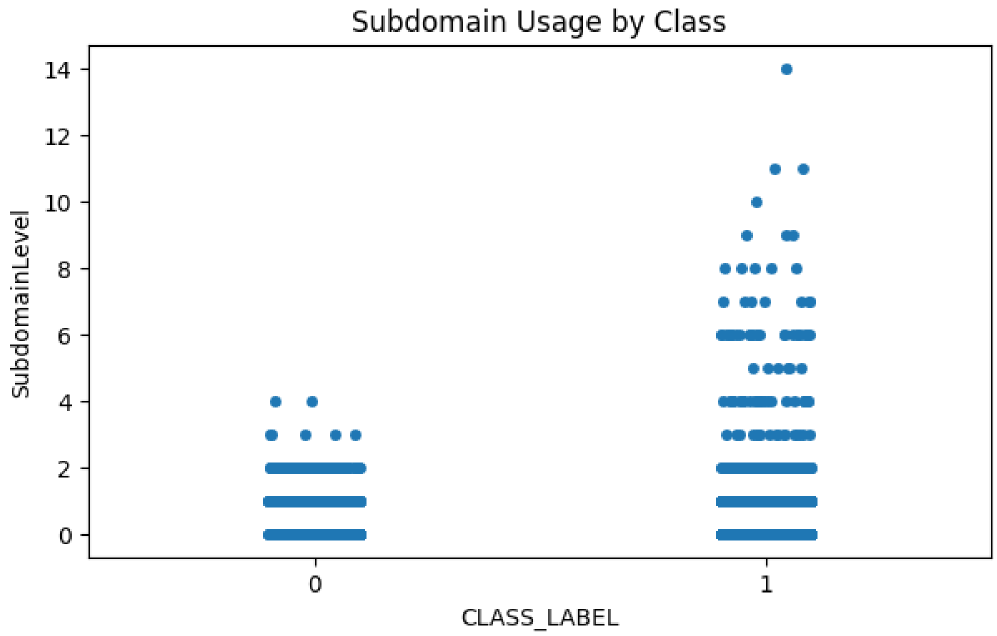


8.Compare distribution of number of subdomains in classes

Insight:

Phishing URLs tend to contain a higher number of subdomains, creating long and confusing structures to mislead users.

Legitimate websites usually maintain simple and recognizable domain formats.

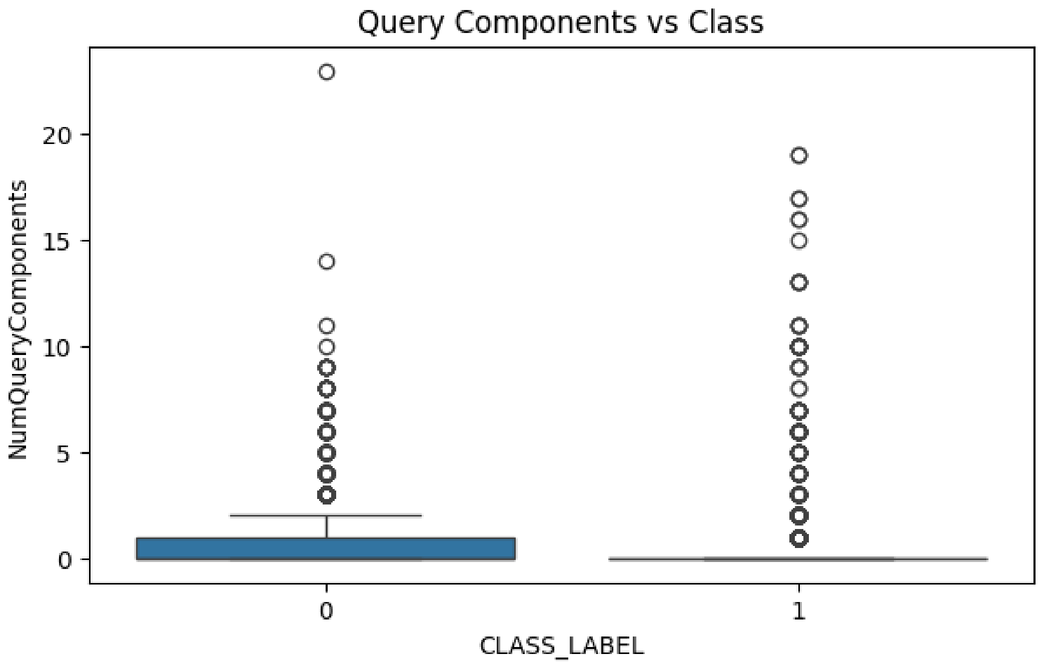


9. Do phishing websites use more query components?

Insight:

Phishing URLs often include more query components to encode redirects or hide malicious parameters.

Legitimate websites keep query structures minimal and straightforward for clarity and usability.



8. Conclusion

The analysis of the phishing website dataset has provided significant insights into how various URL characteristics, domain behaviours, and web interaction features contribute to identifying phishing threats. Through Exploratory Data Analysis (EDA) and machine learning techniques, important patterns were discovered that distinguish phishing websites from legitimate ones.

The results clearly indicate that phishing webpages often exhibit suspicious URL structures such as excessive subdomains, longer URL lengths, abnormal usage of special characters like “-” and “@”, and a higher number of redirects or query parameters. Additional behavioural indicators, including the presence of pop-ups, disabled right-click functionality, or insecure protocols (non-HTTPS), were strongly associated with malicious activity.

Machine learning classification models applied in this project demonstrated the effectiveness of automated phishing detection, achieving strong performance in correctly identifying phishing sites. These findings reinforce the importance of leveraging feature-based detection systems to enhance online security and reduce cyber-attack risks.

Overall, this project highlights the crucial role of data analytics and predictive modelling in cybersecurity. The outcomes not only show how phishing patterns can be analysed effectively but also provide a foundation for developing real-time phishing detection systems that can protect users from fraudulent websites and safeguard sensitive information on the internet.

9. References

- Phishing Legitimate Websites Dataset (UCI Repository / Kaggle Source)
- pandas Documentation – https://pandas.pydata.org
- Matplotlib Documentation – https://matplotlib.org
- Seaborn Documentation – https://seaborn.pydata.org
- Research Articles on Circadian Rhythm and Sleep Patterns – National Sleep Foundation.