A dissertation submitted to the **University of Greenwich**
in partial fulfilment of the requirements for the Degree of

Master of Science
in
Computer Forensics and Cyber Security

A Compare Productive Algorithms of Machine Learning models for Phishing Detection and Prevention.

Name: Parth D. Vashishth

Student ID: 001232207

Supervisor: Dr Tuan Vuong
Submission Date: 16th August 2023
Word count: 11302

ABSTRACT

Phishing attacks have grown to be a big cybersecurity concern, posing serious risks to people and businesses everywhere. This research focuses on creating a reliable phishing detection system utilizing machine learning training modules to address these emerging threats. The objective is to achieve high accuracy in phishing attempt detection, improving security and shielding users from such deceptive attacks. As part of the project's methodology, machine learning techniques are used for analysing huge datasets that contain both legal and phishing incidents. The detection model is trained using a variety of supervised learning methods, including Decision Trees, Random Forests, and Support Vector Machines (SVM). Furthermore, feature engineering approaches are used to extract pertinent and discriminative traits that can separate legitimate communications from phishing attempts. Anti-phishing describes measures taken to thwart phishing attempts. Phishing is a type of cybercrime in which attackers contact people by email, text, or phone and request critical information from them while posing as well-known or reliable businesses. Additionally, users might be asked to submit sensitive data such as bank accounts or credit card information. Attackers may exploit this data once it has been gathered to access accounts, steal data and identities, and install malware on a user's machine. With the aid of technological tools, we may gather extensive data that will be useful in recognising every phishing activity. The objective of this project is to recognize and monitor the phishing activity is being performed by various machine learning techniques. In this project data will collecting from open-source website like Kaggle and Phish Tank. This dataset also serves as an input for project functional and non-functional requirements. The model here applied are Logistic Regression, K-nearest Neighbours, Support Vector Machines, Naïve Bayes, Random Forest, Decision Tree, Gradient Boost is used. The result from the research shows that in order to find phishing activity all the machine learning model works well; however, Gradient Boost give us the highest result of 97% compared to other models.

Keywords:

Phishing Detection, Machine learning, Cybersecurity, Unsupervised Learning, decision tree, random forest, support vector machines, accuracy, precision, recall, F1-score, training modules, data analysis, cyber defence, cyber threats.

ACKNOWLEDGEMENT

Without their help and support, several people would not have been able to complete this dissertation project. My dissertation advisor, DR. Tuan Vuong, has been a huge help in the preparation and execution of this research, and I would like to start by expressing my deep gratitude for all of his wise counsel. His insightful counsel, dynamism, vision, and honesty motivated me. His methods, concept, and numerous hours of work helped me effectively complete the research. He is a specialist in the field like Cyber Physical system security with 9 publications and over 10 years of international experience in software development, IT Operation and Cyber Security. Throughout my MSc in Computer Forensics and Cyber Security, Greenwich University's experience was a great help to me.

I want to express my heartfelt gratitude to the friends and coworkers that I studied and exchanged information with. They encouraged me to think differently about a variety of things that push me to a better project outcome through engagement, debate, and teamwork.

I want to thank both Dr Tuan Vuong and Dr Anatolij Bezemskij for agreeing to have the project demonstration on the schedule day.

Therefore, I believe I will be able to successfully complete the dissertation and MSc Computer Forensics and Cyber Security programme with the help of the folks I've already mentioned.

Table of Contents

List of Figure

List of Table

Chapter 1: Introduction

1.1 Overview

The internet has become a necessary component of our lives, but it also offers several chances to engage in harmful acts like phishing discreetly. Phishers attempt to trick their victims by employing social engineering techniques or building fake websites in order to obtain information such as account IDs, usernames, and passwords from people and businesses. Although several strategies have been put out to identify phishing websites, phishers have developed ways to circumvent these strategies. Machine learning is one of the best techniques for spotting these dangerous behaviours. This is so that machine learning techniques can recognise the common traits shared by the majority of phishing assaults. Cybersecurity is of utmost significance in today's digital world. Cyber threats like phishing assaults are constantly becoming more sophisticated and prevalent. In order to steal sensitive information including login passwords, financial information, and personal information, these assaults frequently target individuals and organisations. There is a need for creative solutions to address this problem. In order to recognise and prevent phishing attempts, the Phishing Detection using Machine Learning project makes use of cutting-edge machine learning techniques. The goal of the project is to create a reliable and precise system that can improve computer performance and help all users determine if a website is real or fraudulent.

The research intends to analyse numerous properties and patterns inside emails and URLs using machine learning algorithms to differentiate between authentic communications and phishing efforts. The system will be taught to recognise subtle signals and signs that are suggestive of phishing by training the model on a diversified dataset comprising both genuine and phishing occurrences. This initiative is of utmost importance since it advances ongoing cybersecurity efforts. Successful adoption might greatly lower the dangers brought on by phishing attempts, shielding people and businesses from possible financial and data breaches. An adaptive and machine learning-based strategy gives a proactive response to a constantly shifting threat landscape as phishing assaults continue to develop.

The development of phishing attempts raises serious concerns in a digital environment rife with cyber risks. Our study offers a novel solution to this problem by employing machine learning to thwart phishing efforts. Our goal was to create a strong phishing detection module that could tell the difference between honest and dishonest online activity.

Our project's main component was a large dataset made up of both proprietary and open-source data. This dataset, which has 30 dimensions, serves as the basis for our module. We started with

thorough data preparation in our process. To assure the quality and dependability of following studies, we were aware of how crucial it was to refine the dataset. The data has to be cleaned, pre-processed, and organised at this step-in order to retrieve insightful information while removing noise.

We used a combination of machine learning techniques to develop a reliable detection system. We aimed to identify the most efficient strategy for our particular assignment by utilising the capabilities of algorithms like Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, and Gradient Boosting.
Enhancing the effectiveness of our module required feature preprocessing and selection in addition to the Information Gain module. In order to improve the module's ability to discriminate, this stage was designed to reduce the dataset to its most important features.
The crucial next stage in our strategy was visualising the outcomes. We aimed to identify developments and patterns that might not be visible in raw data by converting complicated data into graphical representations.

In conclusion, our approach is a bold attempt to combat phishing attacks using a highly developed machine learning-based detection module. We wanted to develop a comprehensive and efficient defence against phishing attempts by integrating data preprocessing, a set of machine learning algorithms, feature preprocessing and selection, and cutting-edge visualisation techniques. In-depth discussion of our methodology, findings, and consequences will be provided in the sections that follow. The methodology, dataset, feature selection, model design, training procedure, and result assessment will all be covered in more detail in the next sections of this. The final objective is to create a phishing detection system that is trustworthy and effective and can be included into current security architecture to provide a greater degree of protection against online attacks.

1.2 Aim and Conclusion:

1.2.1 Aim:

In order to successfully distinguish between genuine communications and phishing efforts, the project's objective is to construct a reliable and accurate phishing detection module utilising machine learning technique. The goal of this project focused on the industry is to use machine learning to create a sophisticated and flexible phishing detection module. The main goal is to provide a cutting-edge solution that smoothly fits into industrial cybersecurity frameworks,

enabling accurate identification of phishing attempts while minimising false positives and guaranteeing unhindered company operations.

1.2.2 Objective:

Data Collection and Preparation:

- Assemble a broad dataset including both self-compiled and open-source data.
- To eliminate noise and assure data quality, the dataset should be cleaned, pre-processed, and organised.

Algorithm Selection and Implementation:

- Use a variety of machine learning methods, such as Gradient Boosting, Naive Bayes, K-Nearest Neighbours, Support Vector Machine, Logistic Regression, Decision Tree, and Random Forest.
- Implement classification techniques, then use the prepared dataset to train models.

Feature Preprocessing and Selection:

- Preprocess features to standardise and normalise the properties of the data.
- To find useful characteristics for precise categorization, use feature selection approaches, such as Information Gain.

Model Evaluation and Comparison:

- To evaluate the model, divide the dataset into training and testing subsets.
- Analyse each algorithm's performance using measures like accuracy, precision, recall, and F1-score.
- Analyse the efficacy of various phishing detection techniques.

Data Visualization and Interpretation:

- Utilise graphical representations to see algorithmic outcomes.
- Understand the effects of various algorithms on the dataset by analysing and interpreting visualisations.

Dimensionality Reduction using PCA:

- Apply Principal Component Analysis (PCA) to lessen the dimensionality of the dataset.
- For better understanding, convert the 30-dimensional data into a 2-dimensional representation.

Ethical and Legal Considerations:

- Discuss the moral issues raised by user privacy and data usage.
- Make that the project is implemented in accordance with legal requirements and data protection legislation.

Documenting and Reporting:

- Record every step of the project's execution, including the collection of the data, the application of the algorithm, the outcomes of the assessment, and the visualizations.
- Create a thorough project report that includes the process, results, and consequences.

Contribution to Cybersecurity:

- By showcasing the effectiveness of machine learning in phishing detection and make a contribution to the field of cybersecurity in terms of insights and conclusions.
- Make suggestions for more advancements and modifications to the created module.

Chapter 2: Literature Review

2.1 Introduction:

Attacks using phishing techniques have become one of the most common cyber hazards, affecting people, businesses, and institutions all over the world. In these attacks, consumers are tricked into disclosing private information, like login passwords or financial information, using phoney emails, websites, or social engineering techniques. Researchers and cybersecurity professionals have investigated a variety of phishing detection methods, including rule-based heuristics, blacklists, and more recently, machine learning-based approaches, in order to address this expanding threat. This study of the literature offers an overview of the body of knowledge regarding phishing detection techniques, with a focus on machine learning-based algorithm to get good accuracy at after performing all task.

The changing security issues We need to identify any risks that prey on data and directly or indirectly harm that person. Attackers or phishers disguise their identity as legitimate internet users, nevertheless. (Said Salloum, 14 June 2022).

2.2 Heuristic and rule-based approaches: To find suspicious features in emails or URLs, early phishing detection methods used rule-based heuristics. These guidelines were developed based on well-established phishing attack patterns and telltale indicators, like misspellings, odd domain names, and dubious links. Although these methods had some success, they frequently lacked adaptability and found it difficult to keep up with the attackers' constantly shifting strategies. A quick and efficient method to identify phishing assaults that adhere to a pattern is to use rule-based heuristics. But as attackers develop their skills, they might produce phishing emails and webpages that are more challenging to spot using rule-based heuristics. Furthermore, a significant number of false positives produced by rule-based heuristics may cause users to become less sensitive to phishing warnings. Machine learning-based phishing detection methods have gained popularity in recent years. From a huge dataset of known phishing emails and websites, machine learning techniques can understand the patterns of phishing attacks. This makes rule-based heuristics less flexible than machine learning techniques when it comes to novel phishing attacks. The number of false positives can be decreased by using machine learning techniques to produce more accurate phishing warnings.

2.3 Blacklists and URL Analysis: Two methods that are frequently using to identify phishing attempts are blacklists and URL analysis. Blacklists are databases of well-known phishing URLs

and domains that are kept up to date by companies like Phish Tank and Kaggle. These blacklists are regularly updated with the latest phishing URLs and domains. The browser will typically block access to a URL that is on a blacklist in Phish Tank and alert the user that the URL may be hazardous and require action regarding their actions and that website's statement regarding website behaviour. As opposed to this, Kaggle is a Google community for data scientists and machine learning engineers where you can find datasets that can aid in the development of AI and machine learning models as well as the resolution of data science difficulties. Another method for identifying phishing assaults is URL analysis. In order to assess whether a URL is likely to be malicious, it is necessary to perform a URL analysis. These traits include the URL structure, the domain name, and the usage of dubious keywords. A URL is probably a phishing URL if it displays several suspicious traits. Though more dynamic than blacklists, URL analysis can also be more complicated and prone to mistakes. Furthermore, URL analysis frequently fails to identify zero-day phishing attempts, which employ cutting-edge methods that have not yet been shown to be dangerous.

2.4 Machine Learning modules: After gathering data and carrying out feature extraction. The next crucial step is to incorporate machine learning models into our dataset and train our model using the information provided. We may train our model using a variety of techniques, including logistic regression, random forest, and many others. Finally, based on the findings, we must analyse which model produces the best and most accurate results.

2.5 Recent Phishing Attack:

According to McAfee report of cyber security in 2020, a prominent phishing incident targeted PayPal user, a popular online payment network. The ingenuity and flexibility of the phishing techniques used by hackers are demonstrated by this instance. (McAfee, 2023)

2.5.1 Attack Description:

Users in this assault got emails that seemed to be real and imitated PayPal's official correspondence. In the expertly designed emails, recipients were informed of a purported "security breach" or "unauthorised transaction" on their PayPal accounts. The emails closely resembled correspondence from PayPal's official account, down to the convincing logos, style, and wording.

The email included a link that instructed recipients to promptly fix the problem by clicking on it. The link took users to a phoney website made to look a lot like PayPal's login page. After then, users were invited to provide their login details and other personal data.

2.5.2 Objective:

The primary aim of this initiative is to perform a machine learning task this literature review's primary goal is to thoroughly analyse the cutting-edge methods and approaches used in the field of phishing detection with machine learning algorithms. In addition to exploring the important features and datasets used, the review attempts to highlight the trends and breakthroughs in this subject, as well as the strengths and weaknesses of existing methodologies. The review aims to provide useful insights that can aid the creation of more precise and reliable machine learning-based phishing detection systems by analysing a wide range of academic papers and publications. The review also attempts to highlight potential research gaps and difficulties, opening the door for future research paths to improve the intended result and efficiency of phishing detection methods in actual situations.

2.6 Challenges in Phishing Detection:

Every person in this contemporary day visits a number of different URLs. According to Forbes, there are 1.13 billion websites in the globe, but only a small portion of them is regularly updated and utilised. As opposed to that, a new website is being created every three seconds. By the time you finish reading, 200 new websites will have been developed.

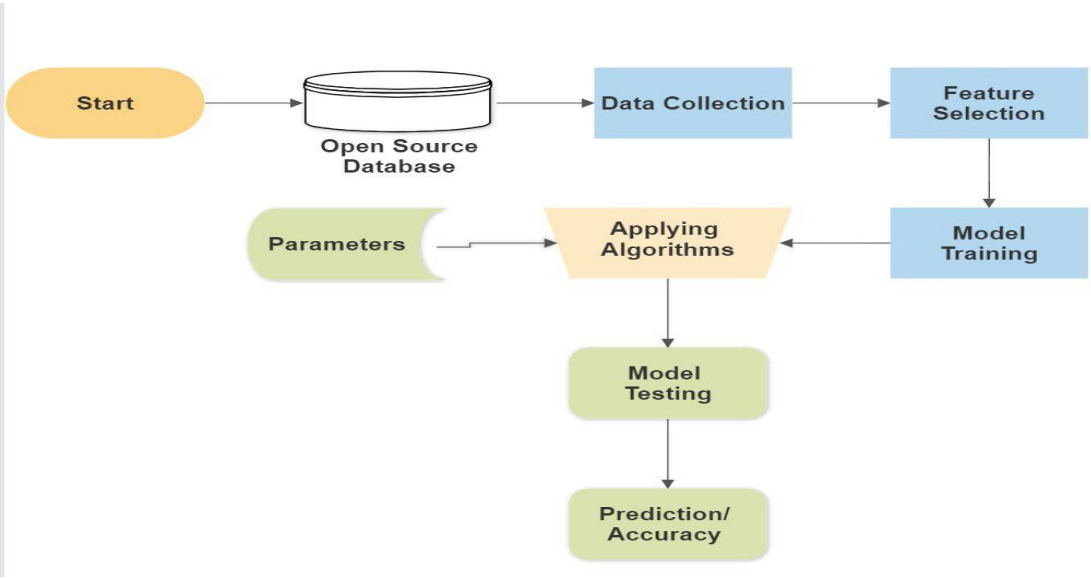


Figure 1 Different type of process

Figure 1 describe different type of process phase which we have to perform in this. According to vedesecure.com selection of candidate URLs the reputation of the URL domain is considered first. The proportion of malicious URLs that have been connected to and discovered in the past is the major factor that determines a domain's reputation; simply because a domain is well-known does not imply that it is reliable.

These domains are regarded as suspect due to the lack of past reputation data associated with unfamiliar domains. Moreover, as it is a common tactic used in cyberattacks, each instance of "burst" activity displayed by these domains is subject to close careful examination. A randomised sample strategy will be used in this situation because it is not practical to check every URL manually. This routine sampling procedure seeks to speed up the early detection of phishing websites.

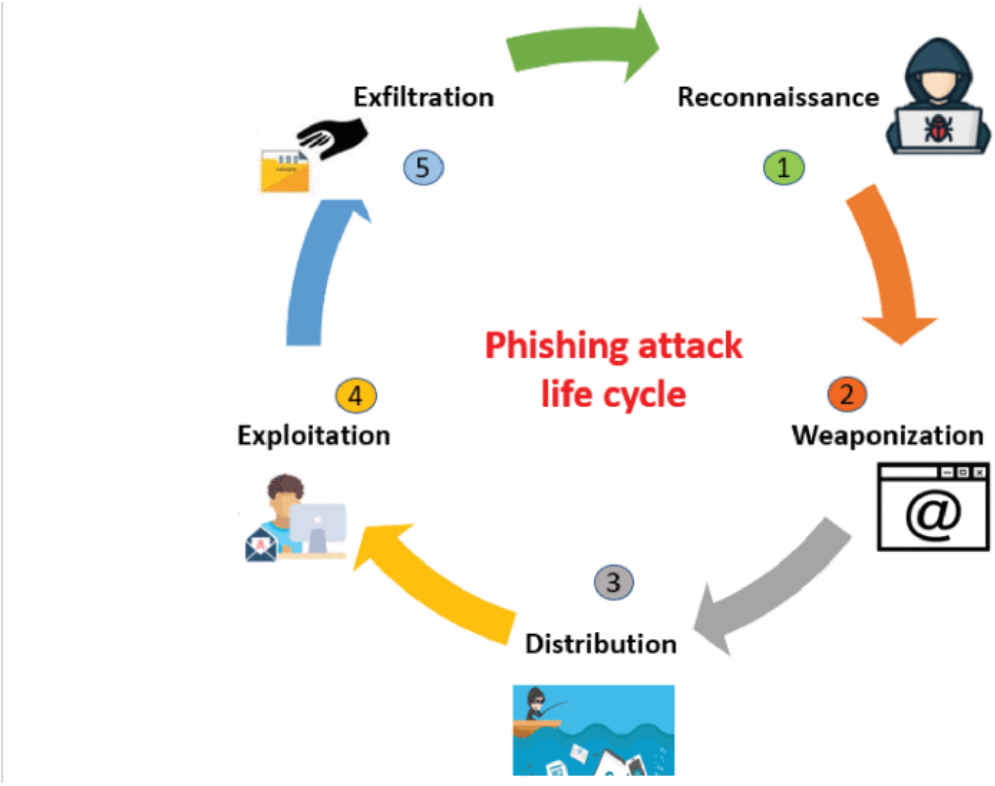


Figure 2 Phishing cycle

2.6.1 Reconnaissance:

The process of acquiring data and intelligence to spot possible phishing assaults before they can do any damage is known as reconnaissance in phishing detection. Phishing is a dishonest practise in which hostile actors try to mislead people into disclosing sensitive information, including login passwords, personal information, or financial data, by posing as reliable organisations.

These false materials frequently appear as emails, messages, or webpages that closely look like reliable sources.

Security experts and systems actively look for and examine various indicators and trends that could point to the existence of phishing attempts during the reconnaissance phase of phishing detection. This proactive method seeks to spot possible hazards before they become problems, reducing the risks of falling for phishing scams.

2.6.2 Weaponization:

When referring to phishing detection, the term "weaponization" refers to the phase of the attack lifecycle where the attacker creates malicious content, such as emails or websites, with the goal of taking advantage of flaws in software or human behaviour. This is the stage where the attacker crafts the phishing material to trick and influence the targets into carrying out predetermined actions, including clicking on harmful links, downloading malware, or disclosing sensitive data.

2.6.3 Distribution:

When discussing phishing detection, the term "distribution" refers to the stage of an attack where the harmful content, such as phishing emails or deceptive websites, is spread to the targeted individuals. At this point, the attackers use a variety of techniques and channels to distribute the false content in an effort to reach a large audience or a specific target audience. The distribution phase's success has a big impact on the phishing campaign's effectiveness because it directly affects the number of prospective victims who might interact with the harmful information.

2.6.4 Exploitation:

In phishing detection, the term "exploitation" refers to the stage of a phishing attack where the attackers take advantage of the trust, curiosity, fear, or other emotions of the targets to individuals them to do a certain action that is advantage to the attackers. Following the description of the phishing content, this phase entails tricking the receivers into interacting with the malicious elements therein, potentially resulting in breach or data theft.

2.6.5 Exfiltration:

When discussing phishing detection, the term "exfiltration" refers to hostile actors' unauthorised removal or transfer of sensitive or confidential information from a victim's system or network. Exfiltration happens in the context of phishing assaults after the victim has been effectively abused and important data has been obtained. With the intention of exploiting it for evil deeds like identity theft, financial fraud, or selling the stolen information on the dark web, the attackers

try to relocate this data away from the victim's surroundings, usually to a location under their control.

The research on phishing detection emphasises the need for novel strategies to combat emerging cyberthreats. Similar to the dataset used in this study, open-source and self-compiled datasets are frequently used for their variety and portrayal of real-world circumstances.

Machine learning techniques like Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, and Gradient Boosting will be used in this project to train our machine learning module due to their adaptability in spotting intricate patterns present in phishing attempts.

The inclusion of Information Gain, feature preprocessing, and selection procedures are proven techniques for enhancing model performance. These methods help to identify characteristics that are essential for precise categorization.

Techniques for displaying data clearly fit with current research trends and make complex findings more understandable. The properties of the dataset and the performance of the algorithms are revealed by the graphics.

Innovative techniques are supported by the use of Principal Component Analysis (PCA) to reduce the complexity of high-dimensional data. PCA improves comprehension while keeping key information by transforming data into a 2-dimensional representation.

2.7 Critical Analysis of the Project:

It is admirable that this project uses machine learning to produce a module for phishing detection. However, a rigorous review indicates that different project components have both strong points and room for development.

2.7.1 Strength:

- Comprehensive strategy: This project takes a comprehensive strategy that includes data preparation, multiple machine learning techniques, dimensionality reduction, and result visualisation.
- Algorithm Diversity: A thorough investigation of alternative solutions is demonstrated by the use of a variety of machine learning algorithms, such as Logistic Regression, K-

Nearest Neighbours, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, and Gradient Boost.

- Dimensionality Reduction: Using PCA to reduce dimensions improves the model's interpretability while retaining a significant amount of the variation from the original data. This choice improves the production of insights and the visual portrayal.
- Visual Representation: The use of ROC curves, confusion matrices, and graphs to visualise data adds a crucial level of insight to model performance and error analysis.

2.7.2 Areas of Improvement:

- The explanation does not provide enough information about the feature engineering process. The project would be strengthened by a more thorough justification of the selected criteria and how they relate to phishing detection.
- Data Source and Quality: It is acknowledged that the project relies on open-source data, but nothing is said about the data's quality, representativeness, or any biases. For the project to be credible, it is essential to discuss data source restrictions.
- Model Interpretability: The project addresses model interpretability using PCA, although more research into model interpretability techniques would improve transparency, especially for complicated models like Gradient Boost.
- Hyperparameter Tuning: The performance of the selected algorithms could be enhanced by investigating hyperparameter tuning methods. It would be more in-depth to go into the selection and tuning of the hyperparameters.

Chapter 3: Analysis of System

3.1 Legal Issue:

Data Usage and Ownership: Making use of both public domain and private data poses issues of ownership and copyright. To prevent legal issues, it is essential to ensure that data usage agreements are followed and to secure all relevant licences.

Data Privacy: Working with sensitive data necessitates respect to data privacy laws like GDPR, even when doing so for research reasons. To prevent breaking privacy rules, proper anonymization and data protection methods are crucial.

3.2 Social Issue:

User Privacy: Examining email content is a step in creating a phishing detection system. To prevent eavesdropping on people's conversations, it's crucial to strike a balance between email security monitoring and user privacy.

Fairness and Bias: Biases existing in the training data might be inherited by machine learning models. Maintaining fairness requires making sure the model does not bias against particular groups or communication styles.

3.3 Ethical Issue:

Informed Consent: Obtaining informed permission is essential if actual user data is used. Users should be informed that their privacy is protected and that research may utilise the messages they send.

Algorithmic Transparency: It is morally right to know how the system decides what to do. Users and stakeholders should be aware of the reasoning for the classification of emails as possible phishing attempts.

3.4 Professional Issue:

Accurate Representation: It's important to appropriately convey the system's capabilities and constraints. Overestimating the system's efficiency might result in a misuse of its use for important security choices.

Continuous Monitoring: It is crucial to continuously monitor and update the system as phishing techniques change. If this isn't done, security may be jeopardised by false positives or missing phishing efforts.

Collaboration and Attribution: If the project involves a component of collaborative research, it is the responsibility of the professional to ensure accurate attribution of contributions. It's crucial to acknowledge the contributions of your partners and to communicate clearly.

To sum up, the project's success and influence depend greatly on its legal, social, ethical, and professional components. Consciously addressing these problems guarantees that the phishing detection module complies with legal requirements, societal conventions, ethical standards, and professional obligations in addition to being technically successful.

Chapter 4: Data Pre-Processing:

4.1 Understanding Data:

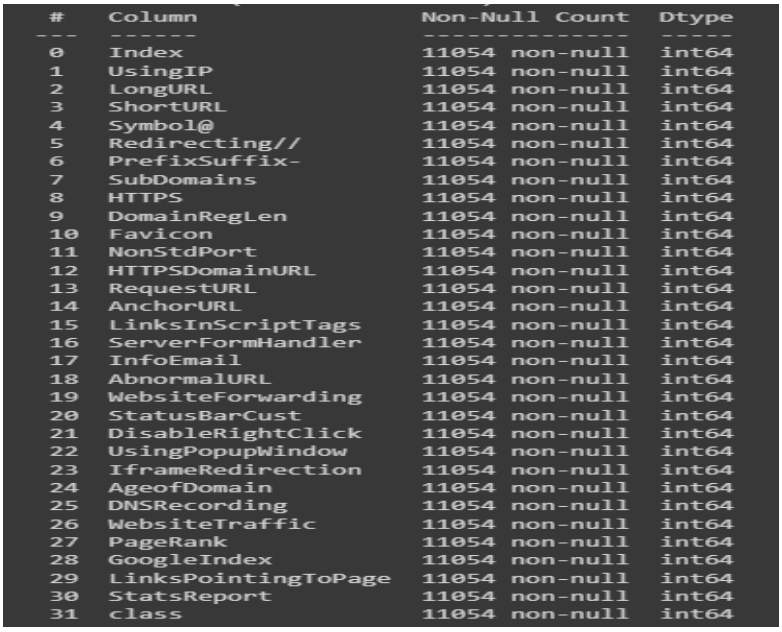
The dataset for this project was found from the (Anon., 2023). The offered dataset, which includes crucial materials to support the creation of a machine learning model, is accessible in text and CSV forms. Website URLs are included in the offered data, which also includes 11000 examples of website URLs, each of which is distinguished by 30 different factors, and one is for class module. Every website URL has a class label that is either 1 (identifying a phishing website) 0 (identifying a suspicious website) or -1 (representing a legitimate website) and is connected with it. (CHAND, n.d.)

Table 1 Data Understanding

Illustrious Data	Data Labelled
Using IP	2
Long URL	3
Short URL	2
Symbol @	2
Redirecting //	2
Prefix Suffix -	2
Sub Domains	3
HTTPS	3
Domain Reg Len	2
Favicon	2
Non Std Port	2
HTTPS Domain URL	2
Request URL	2
Anchor URL	3
Links In Script Tags	3
Server Form Handler	3
Info Email	2
Abnormal URL	2
Website Forwarding	2
Status bar Cust	2

Disable Right Click	2
Using Popup Window	2
I frame Redirection	2
Age of Domain	2
DNS Recording	2
Website Traffic	3
Page Rank	2
Google Index	2
Links Pointing to Page	3
Stats Report	2
Class	2

The project includes a well-structured code template that is integral to the development process. This template is divided into two fundamental sections, each serving a distinct purpose. The initial section is exclusively focused on the importation of imperative Python modules. This serves as the foundational step, establishing the prerequisites necessary for the subsequent coding endeavours. Following this, the second segment of the code template takes centre stage. Here, a pivotal function is meticulously crafted. This function is responsible for seamlessly loading the dataset, a core component of the project, into the working environment. Additionally, this segment offers succinct yet comprehensive descriptions, elucidating the inherent characteristics of the input and output fields associated with the dataset. Through these strategic divisions, the code template acts as a roadmap, facilitating a systematic approach to data manipulation and model construction in this machine learning endeavour.



#	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

Figure 3 understanding data.

The dataset's use in establishing the project's parameters and expected results is included in the project's scope in addition to its use as a training resource. The dataset has two uses: it supplies the data needed for model training, but it also acts as a foundation for the project, defining its parameters and goals. Aspects that are both functional and non-functional are included in this expansive scope. On the one hand, it explains the actual conditions needed to put the code into practise. On the other hand, it includes extraneous factors like project objectives, the anticipated accuracy of model projections, and an overall assessment of the project's effectiveness. In essence, the project scope is a dynamic blueprint that navigates the course of the project, incorporating technical and strategic dimensions to ensure a holistic and impactful outcome.

	count	mean	std	min	25%	50%	75%	max
UsingIP	11054.0	0.313914	0.949495	-1.0	-1.0	1.0	1.0	1.0
LongURL	11054.0	-0.633345	0.765973	-1.0	-1.0	-1.0	-1.0	1.0
ShortURL	11054.0	0.738737	0.674024	-1.0	1.0	1.0	1.0	1.0
Symbol@	11054.0	0.700561	0.713625	-1.0	1.0	1.0	1.0	1.0
Redirecting//	11054.0	0.741632	0.670837	-1.0	1.0	1.0	1.0	1.0
PrefixSuffix-	11054.0	-0.734938	0.678165	-1.0	-1.0	-1.0	-1.0	1.0
SubDomains	11054.0	0.064049	0.817492	-1.0	-1.0	0.0	1.0	1.0
HTTPS	11054.0	0.251040	0.911856	-1.0	-1.0	1.0	1.0	1.0
DomainRegLen	11054.0	-0.336711	0.941651	-1.0	-1.0	-1.0	1.0	1.0
Favicon	11054.0	0.628551	0.777804	-1.0	1.0	1.0	1.0	1.0
NonStdPort	11054.0	0.728243	0.685350	-1.0	1.0	1.0	1.0	1.0
HTTPSDomainURL	11054.0	0.675231	0.737640	-1.0	1.0	1.0	1.0	1.0
RequestURL	11054.0	0.186720	0.982458	-1.0	-1.0	1.0	1.0	1.0
AnchorURL	11054.0	-0.076443	0.715116	-1.0	-1.0	0.0	0.0	1.0
LinksInScriptTags	11054.0	-0.118238	0.763933	-1.0	-1.0	0.0	0.0	1.0
ServerFormHandler	11054.0	-0.595712	0.759168	-1.0	-1.0	-1.0	-1.0	1.0
InfoEmail	11054.0	0.635788	0.771899	-1.0	1.0	1.0	1.0	1.0
AbnormalURL	11054.0	0.705446	0.708796	-1.0	1.0	1.0	1.0	1.0
WebsiteForwarding	11054.0	0.115705	0.319885	0.0	0.0	0.0	0.0	1.0
StatusBarCust	11054.0	0.762077	0.647516	-1.0	1.0	1.0	1.0	1.0
DisableRightClick	11054.0	0.913877	0.406009	-1.0	1.0	1.0	1.0	1.0

Figure 4 Mean, Stdiv of the data

The table presented here provides a partial overview of our data description, with further details available subsequently.

4.2 Descriptive Data Analysis:

The dataset includes data on websites divided into three classes: phishing (class 1), suspicious (class 0), and legal (class 1). The objective is to create a classification model that accurately predicts the class of a website based on its qualities. Each website is characterised by a collection of attributes. Now that we are aware of how much information was gathered and how many behaviours and attributes were included in the dataset, it is time to examine the information and get some understanding of it. It becomes challenging to display all 14 of the behaviours in this dataset together due to the sheer volume of observations it includes. Although the qualities for each website are not stated directly, it appears that they are numerical values because of the class names

given (-1, 0, 1). There are 30 qualities for each website, therefore it's critical to comprehend what each one means. Are they connected to the structure, traffic, or other elements of the website? Understanding the nature of these qualities is essential for developing models. Where it changed after we completed the activities depicted in the supplied image respectively.

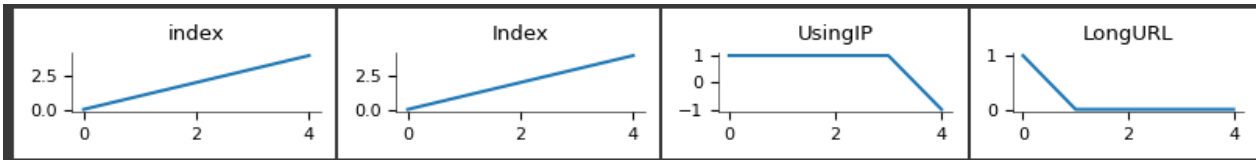


Figure 5 Data Analysis 1

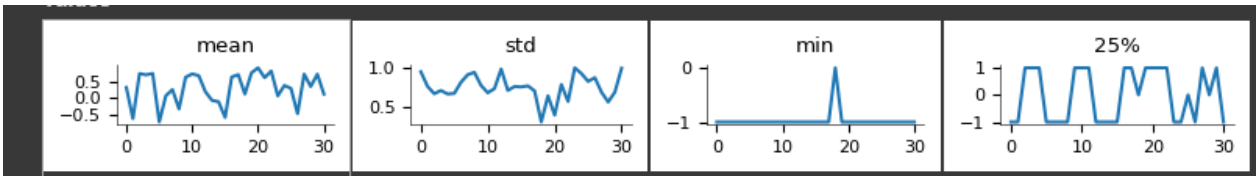


Figure 6 Data Analysis 2

The statistics in the graph above are averaged, have a standard deviation, a minimum value, and a 25% percentile, respectively. It represents value by linear method above. Each entry in the dataset appears to represent a website and any related information in comma-separated values (CSV) format.

4.3 Distribution of Data:

4.3.1Normal Distribution (Univariate):

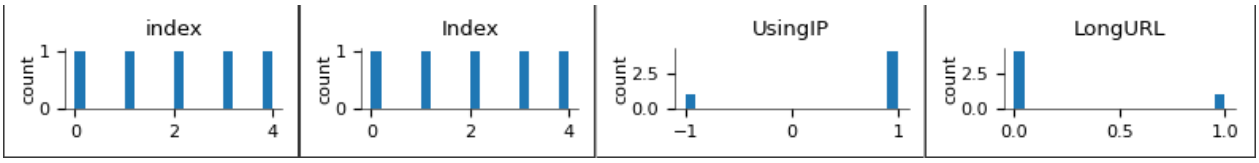


Figure 7 Data Distribution 1

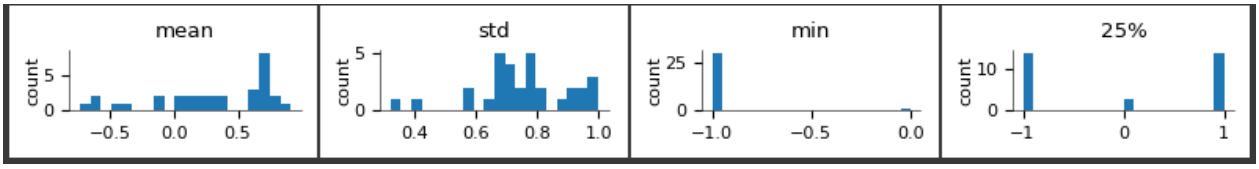


Figure 8 Data Distribution 2

The symmetric bell-shaped curve of the normal distribution, also known as the Gaussian distribution, sets it apart from other distributions. The distribution's mean, median, and mode all fall within this range and are positioned in its centre. The balance between values below and above the mean is indicated by the alignment at the centre. It is noteworthy that many natural occurrences follow the normal distribution, supporting the importance of the normal distribution in statistical research. The central limit theorem, which asserts that the distribution of the sum (or average) of a large number of independently distributed random variables closely resembles a normal distribution, is responsible for its ubiquity. The normal distribution serves as a fundamental idea in statistical theory and practice is commonly used to describe different real-world data sets because of its features.

4.3.2 2-Dimensional Distribution (Bivariate):

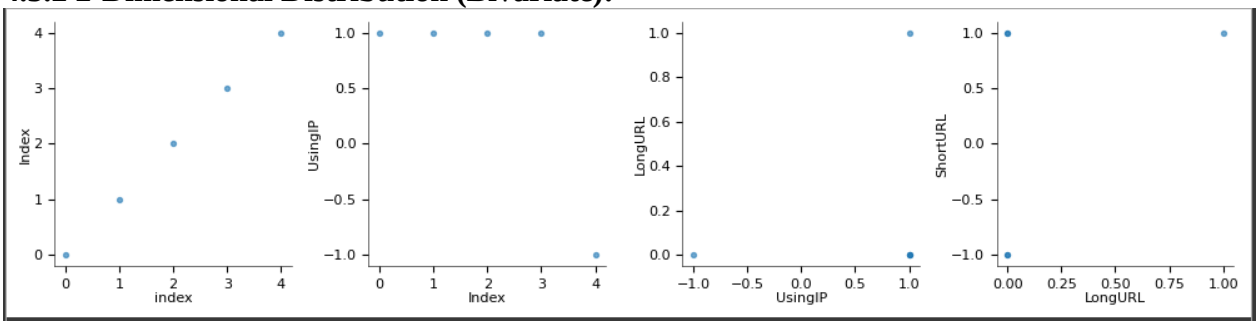


Figure 9 2-Dimensional Data 1

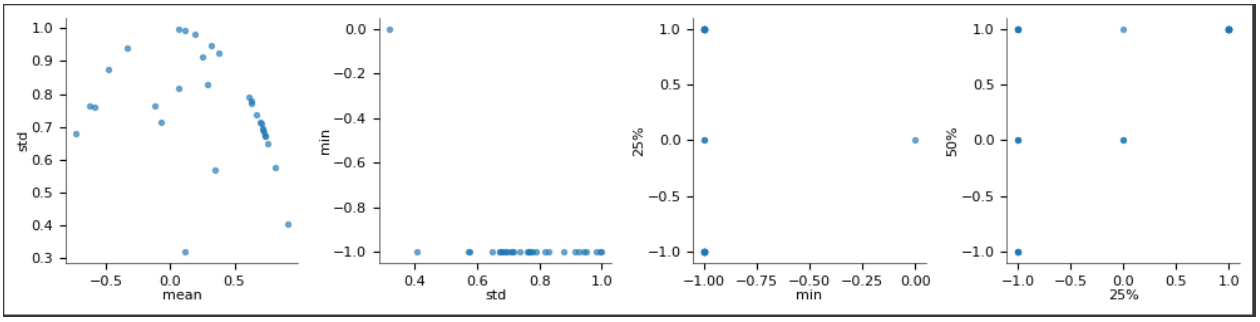


Figure 10 2-Dimensional Data 2

The organisation of data points within a two-dimensional geographic environment is referred to as a 2-dimensional distribution, sometimes known as a bivariate distribution. Two different variables, each of which contributes to the coordinates of a data point in the space, are used in this framework to construct data instances. Scatter plots, a graphical representation where each data point is located according to its respective values along the two variables, are frequently used to aid the visualisation of bivariate distributions. A visual representation of the link between the two variables is provided by each plotted point, which contains the pairing of values related to the two

variables. Scatter plots reveal patterns that shed light on potential dependencies, correlations, and interactions between the two variables. One can gain important insights into the underlying relationships of the data by carefully examining the scatter plot to determine whether changes in one variable are related to systematic changes in the other.

4.3.3 Time Series Distribution:

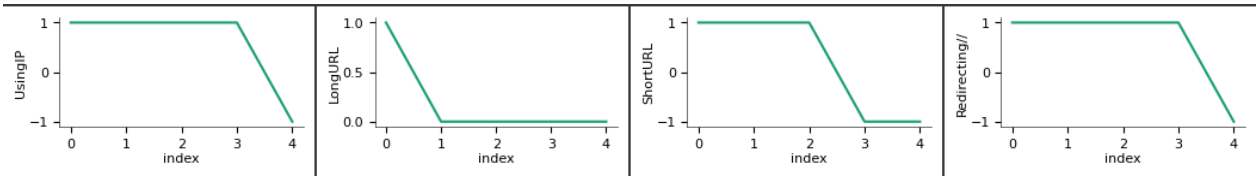
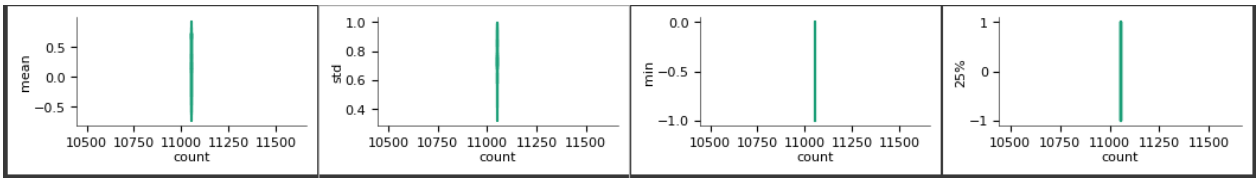


Figure 11 Time Data Distribution



Time series distributions, in which observations are methodically collected at successive time intervals, capture the temporal dynamics inherent in data. Time series data, which are often acquired at regular intervals, offer a thorough understanding of how a variable changes over time. Time series distributions shed light on a variable's trajectory and fluctuation as it moves across the temporal dimension, as opposed to traditional distributions, which represent the static frequency of individual values. Line charts are a key tool for enhancing the understanding of such data. With the horizontal axis indicating the time and the vertical axis designating the variable's value, these plots create a visual story of the variable's journey over time.

4.4 Heatmap Representation:

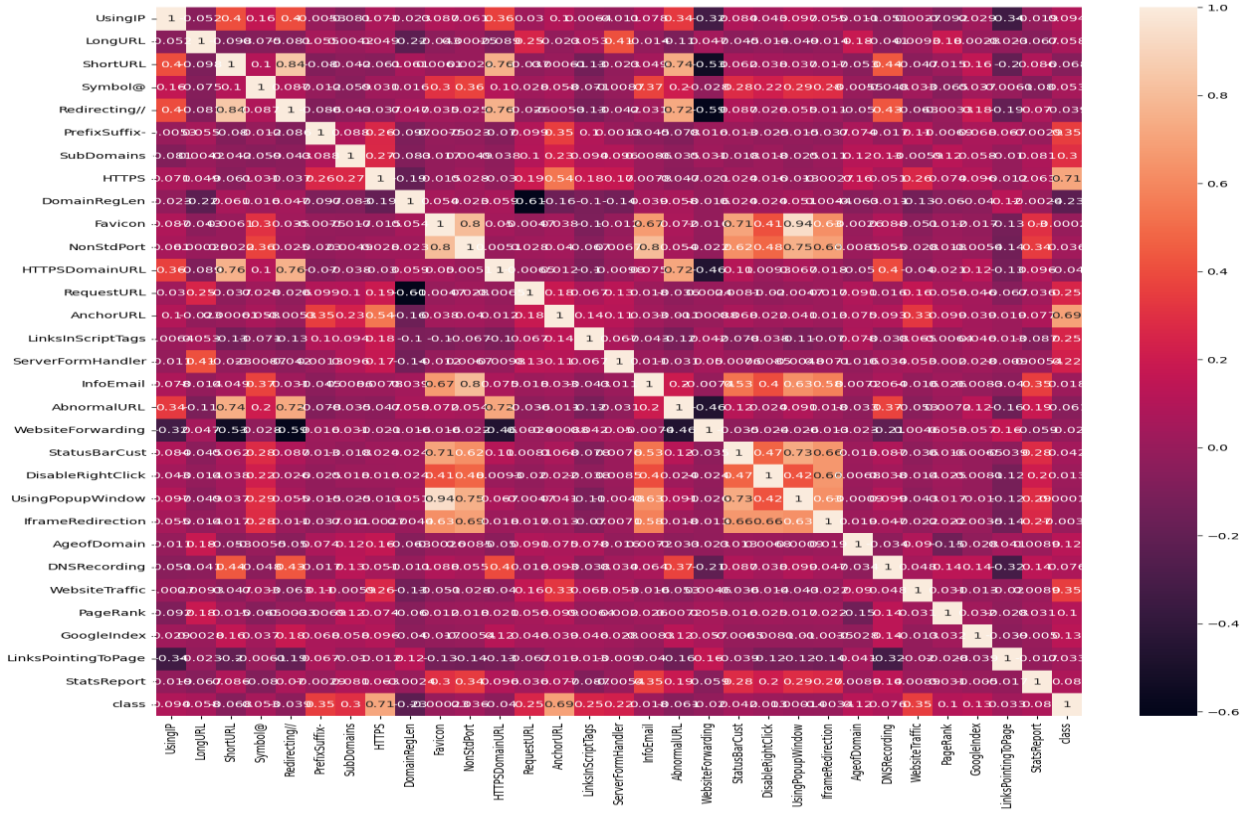


Figure 12 Heatmap Representation

A heatmap is a type of graphic that utilises colour to show a dataset or matrix's values. In data analysis and visualisation, heatmaps are very useful because they offer a simple means of comprehending intricate patterns, connections, and changes in the data. The value of heatmaps in data comprehension rests in their capacity to reduce enormous amounts of data into a visual format that is simple to grasp and draw conclusions from.

- Correlation Analysis: Heatmaps emphasise the strength and direction of correlations between pairs of qualities by visualising the correlation matrix of your data. One colour intensity stands in for positive correlations, another for negative correlations, and neutral colours for no correlation. With the help of this insight, you may spot characteristics that often change together and gain a deeper knowledge of the underlying dynamics of your data.
- Feature Selection: Heatmaps can help in feature selection by pointing out characteristics that are highly associated. To enhance model performance and decrease dimensionality, redundant characteristics with significant correlations can be eliminated.
- Identifying Patterns: Finding Patterns: Clusters of similar behaviours or characteristic groups may be shown by patterns in colour intensity. These patterns can make invisible data structures visible that are not always visible in tabular form.

- Detecting Anomalies: Weird colour patterns may point to outliers or other irregularities in your data. These are situations that dramatically depart from the usual and could call for more research.

4.5 Pie-chart Representation:

A pie chart is a circular graphic that shows how the various categories are distributed and how they compare over the whole dataset. A pie chart might be used in your project to show how the classes (-1 and 1) in your dataset are distributed. The use of a pie chart for your project is explained in detail below:

A pie chart offers a simple and understandable representation of the distribution of the various classes in your collection. The size of each slice of the pie, which represents each class, reflects the percentage of data points that fall within that category. The classes in your situation are -1 for legitimate, 0 for questionable, and 1 for phishing.

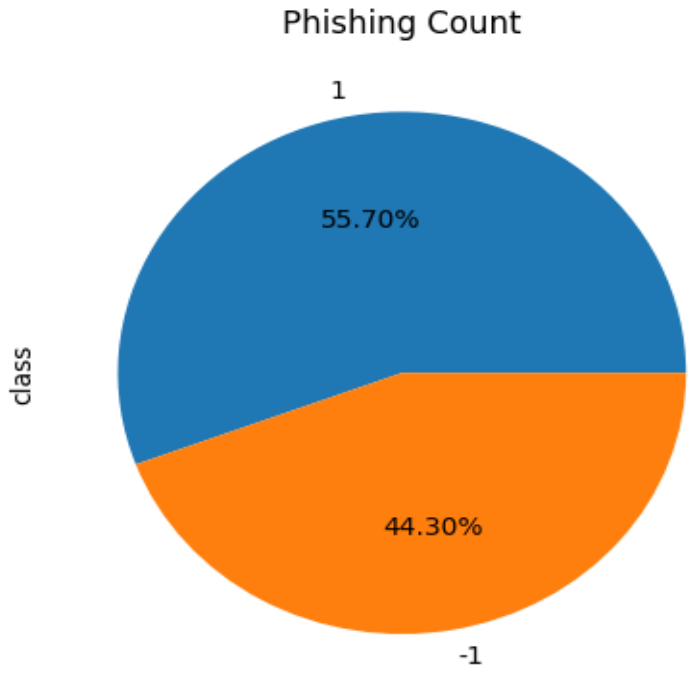


Figure 13 Phishing Pie chart Representation

4.6 Normalization:

In order to standardise the range of characteristics within a dataset, normalisation is a key data preprocessing technique used in phishing detection. This procedure is essential to eliminate biases brought on by different data scales and to guarantee that no one feature has an excessive impact on the detection model. By transforming the characteristics of the data into a common scale, usually between 0 and 1 or -1 and 1, fair comparison and analysis are made possible. Normalisation is to

scale down features to a similar scale. This enhances the model's functionality and training stability.

In basic terms, normalization is the process of converting the data into the range [0,1] or, alternatively, into a unit sphere. As the model only needs to process a restricted range of inputs, this method aids in obtaining the answer more quickly. Not all datasets require this technique, but it is important when the features in the dataset have varying ranges.

Mathematically, the process of normalization can be represented as follows for an attribute 'x': (Anon., 2022)

$$x' = (x - x_{min})/(x_{max} - x_{min})$$

Were,

X' is the value of maximization.

$Xminimum$ is the minimum value of the feature.

$Xmaximum$ is the maximum value of the feature.

There are several normalisation methods, however a few are covered below:

4.6 Min Max Normalization:

With this method, a feature's values are scaled to fall between 0 and 1. This is accomplished by dividing each value by the feature's range, then removing the feature's minimum value from each value. The linkages between the original data values are preserved through min-max normalization. We will have reduced standard deviations as a result of this restricted range, which can reduce the impact of outliers. (Anon., 2023)

$$\mathbf{v' = \frac{v - min(A)}{max(A) - min(A)}(new\_max(A) - new\_min(A)) + new\_min(A)}$$

Min(A) and max(A) are the minimum and maximum values of A.

Where A is an attribute of data

v is the old value of every entry in the data

new_min (A) and new_max(A) are the minimum and maximum values of the range.

4.7 Z – score normalization:

Data are transformed into a conventional normal distribution with a mean of 0 and a standard deviation of 1 using the Z-score normalisation procedure, commonly known as standardisation.

When the data distribution is unknown or there are outliers in the data, this strategy is especially helpful.

$$z = \frac{x - \mu}{\sigma}$$

Where:

X is the original value of the attribute

μ is the attribute's average throughout the dataset.

σ is the attribute's average standard deviation throughout the dataset. (ZACH, 2021)

Here, we employ a normalised dataset with a min-max scaler that must be used immediately for testing purposes.

Chapter 5: Feature Extraction

A crucial step in machine learning and data analysis, feature extraction entails reducing a large number of meaningless characteristics from raw or high-dimensional data. These characteristics minimise noise and redundancy while capturing the most important information from the original data. The efficiency and efficacy of subsequent modelling and analytic activities are improved by this procedure. The process of extracting numerical characteristics from raw data while preserving the original dataset's information is known as feature extraction. This initial dataset is changed into a smaller, easier-to-manage group, which would ultimately simplify the procedure. This method is very helpful when we wish to minimise the number of observations in a huge dataset without sacrificing the dataset's uniqueness. In here we have data which already converted into min, median and standard deviation so that it is easy to understand data without compromise with their uniqueness.

5.1 Mean:

It is calculated by adding up each value in the set, dividing the result by the total number of values, and then computing the result. A dataset's mean or usual value may be determined by using the mean, which is a fundamental measure of central tendency. It is often applied to a variety of statistical studies, data preparation, and model assessment. The mean is the mathematical average of the given numbers. It is nothing but the total sum of all the given numbers divided by the number of given values. We have 11054 websites in this dataset that we checked for the 30 main phishing reasons. This data gives us all we need in advance for each category that is represented in the supplied image.

5.2 Median:

Compared to the mean, it is a measure of central tendency that is less impacted by outliers. You must align every value in the dataset and choose the middle one to determine the median. The median is the average of the two middle values when the dataset has an even number of items. When extreme values or outliers can have a major influence on the mean, the median offers a reliable depiction of the usual value in a collection.

5.3 Standard Deviation:

It shows how far the individual data points deviate from the dataset's mean (average). While a smaller standard deviation denotes that the data points are more closely spaced from the mean, a greater standard deviation says that the data points are more dispersed from the mean. (Frost, 2023)

The standard deviation is derived mathematically by calculating the square root of the sum of the squared deviations between each data point and the dataset's mean. It gives important details about the data's distribution and is frequently utilised in a variety of analysis, including ones that evaluate the spread of values, spot outliers, and comprehend the degree of uncertainty in a dataset.

$$s = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \overline{x})^2}{N - 1}}$$

s = the sample Standard Deviation
N = number of observations
Xi = value of each observation
$\bar{x}$ = the sample mean

Now that we have all the data that was collected from the dataset and watched as the behaviour changed, we can calculate the mean and standard deviation by looking at each of the 30 possible scenarios. The table below provides further information on this.

Table 2 Table of Mean and Standard Deviation

Illustrious Data	Mean	Standard Deviation
Using IP	0.313914	0.949495
Long URL	-0.633345	0.765973
Short URL	0.738737	0.674024
Symbol @	0.700561	0.713625
Redirecting //	0.741632	0.670837
Prefix Suffix -	-0.734938	0.678165
Sub Domains	0.064049	0.817492
HTTPS	0.251040	0.911856
Domain Reg Len	-0.336711	0.941651
Favicon	0.628551	0.777804
Non Std Port	0.728243	0.685350
HTTPS Domain URL	0.675231	0.737640
Request URL	0.186720	0.982458
Anchor URL	-0.076443	0.715116
Links In Script Tags	-0.118238	0.763933
Server Form Handler	-0.595712	0.759168
Info Email	0.635788	0.771899
Abnormal URL	0.705446	0.708796

Website Forwarding	0.115705	0.319885
Status bar Cust	0.762077	0.647516
Disable Right Click	0.913877	0.406009
Using Popup Window	0.613353	0.789845
I frame Redirection	0.816899	0.576807
Age of Domain	0.061335	0.998162
DNS Recording	0.377239	0.926158
Website Traffic	0.287407	0.827680
Page Rank	-0.483626	0.875314
Google Index	0.721549	0.692395
Links Pointing to Page	0.343948	0.569936
Stats Report	0.719739	0.694276
Class	0.113986	0.993527

As a result, the following table defines mean and standard deviation and reduces the number of observations for easier understanding while also speeding up the model and increasing accuracy.

5.4 PCA:

Principal Component Analysis the principal component analysis (PCA) is a dimensionality reduction approach that converts the data into a new coordinate system with orthogonal dimensions that capture the greatest amount of variation. It is frequently used to lower the dimensionality of datasets with large dimensions. The observations of correlated characteristics are changed in this method into a collection of linearly uncorrelated features with the use of orthogonal transformations. By lowering the variance, PCA is a well-liked method for identifying key patterns in the dataset. By reducing the dimension, this approach aims to retain the most data from the original dataset possible.

The following are PCA's primary goals:

PCA is frequently used to overcome problems associated to the "curse of dimensionality" and to increase computing effectiveness in a variety of domains, including image processing, signal processing, and machine learning.

It may be used for many things, including feature selection, noise reduction, and visualisation. In order to better comprehend underlying patterns and enhance model performance, PCA helps to reveal the data's intrinsic structure by converting the data into a new space where the axes are orthogonal and aligned with the direction of highest variance. (Jolliffe, 2002)

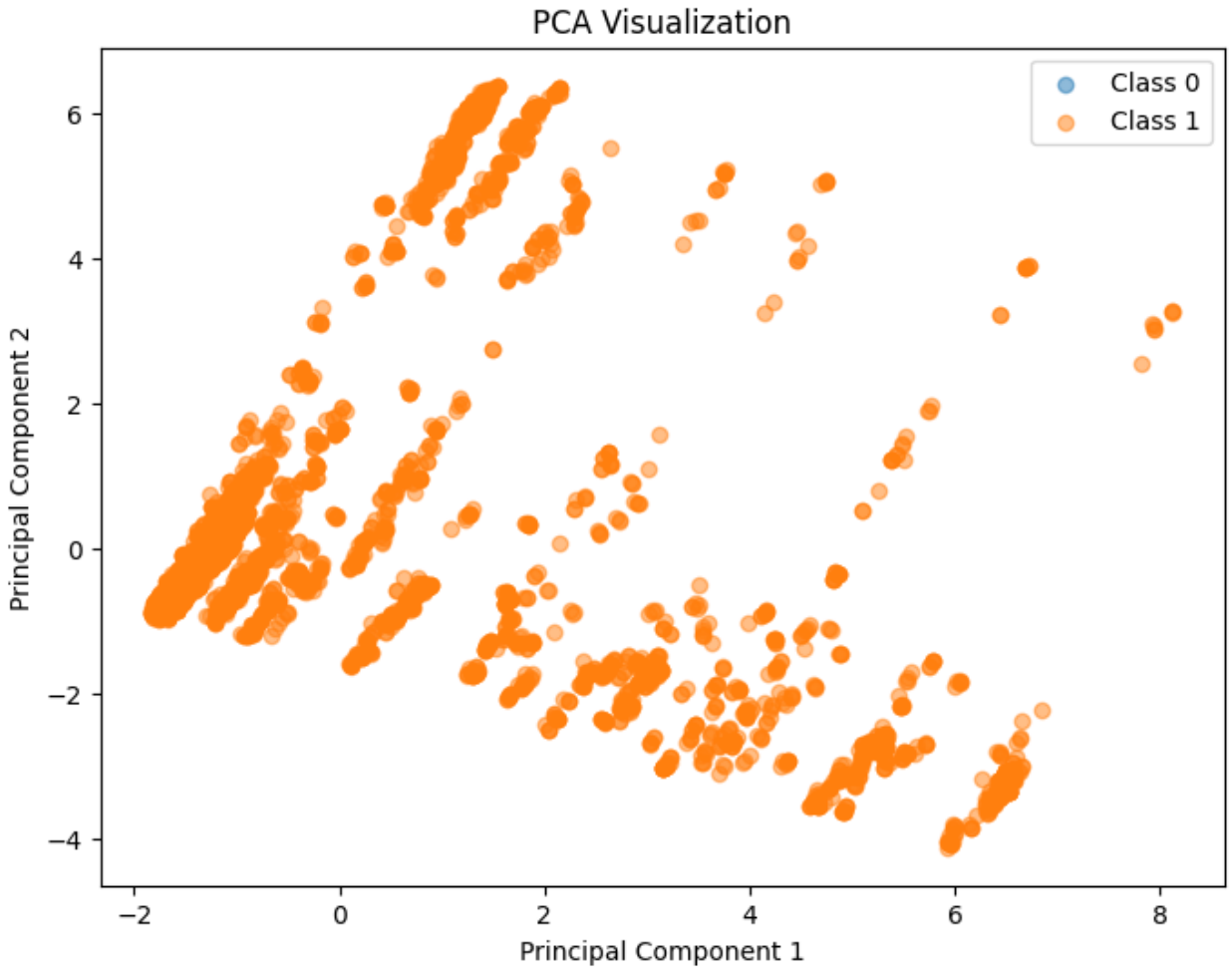


Figure 14 PCA Representation

The pre-processing task leaves a large number of observations in the data frame, so to create a PCA plot, the first observations of each behaviour are taken. The PCA representations of these observations are shown in the above figure, and the corresponding Python code is in the appendix (access to the appendix is provided).

In given PCA1 and PCA2 we calculate variance ratio it means that measures the percentage of the original data's overall variation that is accounted for by each major component. Understanding how much data each primary component retains from the original dataset is made easier by this.

The primary components are ranked according to how well they can explain variation when using PCA. The most variance is explained by the first principal component, followed by the second and so on. Each primary component's explained variance ratio is determined, which sheds light on the importance of each component in terms of capturing the variability of the data. In our PCA model shows PC1 17.28% meanwhile PC2 13.16%

Explained Variance Ratio = Variance explained by the ith component. / Total Variance in the data

In scikit-learn, a widely used machine learning library, the explained_variance_ratio_ attribute of the PCA object provides an array of explained variance ratios for each principal component.

5.4.1 Advantage of Principal Component Analysis (PCA):

- Dimensionality reduction:

PCA aids in the reduction of high-dimensional dataset's dimensionality while preserving the most crucial data. This can increase computational effectiveness and lessen the chance of machine learning models overfitting.

- Noise Reduction:

PCA tends to emphasise signal and reduce noise. By minimising the effects of random fluctuations in the data, it can result in more reliable models.

- Feature Interpretability:

Principal components are combinations of unique traits, yet they can occasionally shed light on the underlying causes that affect the data.

- Visualization:

Data may be transformed via PCA into a simpler-to-see, lower-dimensional space. Understanding data patterns, clusters, and linkages is made easier by doing this.

5.4.2 Disadvantages of Principal Component Analysis (PCA):

- Information Loss:

Some information is unavoidably lost during dimensionality reduction when less significant components are eliminated. Finding a balance between dimensionality reduction and information retention is crucial.

- Normalization is required.

It is strongly advised to normalise the dataset before using principal component analysis to it; otherwise, it would be challenging to locate suitable principal components.

5.5 Computational Requirements:

There are certain computational criteria that must be taken into account for the phishing detection module to be implemented using machine learning successfully:

5.5.1 Hardware:

Processing Power: The project's usage of a variety of machine learning algorithms, particularly ensemble learning and deep learning, can be very demanding. A multi-core CPU or perhaps a GPU with enough computing capacity might hasten model training and assessment.

Memory: To handle the big dataset, preprocessing procedures, and model training, enough RAM is required. Processing breakdowns or speed bottlenecks might result from a lack of memory.

5.5.2 Software:

Python: For data processing, modelling, and visualisation, the project uses Python and its machine learning libraries (Scikit-Learn, Pandas, and NumPy).

Machine learning libraries are necessary for a number of techniques, including Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, and Gradient Boost.

Visualisation libraries: For making different visualisations, such as graphs, scatter plots, ROC curves, and confusion matrices, libraries like Matplotlib or Seaborn are required.

5.5.3 Storage:

Disc space: A sizable quantity of disc space is needed to store the dataset, intermediate results, trained models, and visualisations. For efficient project execution, make sure there is enough storage space.

5.5.4 Time:

Disc space: The dataset, intermediate findings, trained models, and visualisations must all be stored on a large amount of disc space. Make sure there is enough storage space for effective project implementation.

Chapter 6: Machine Learning Module:

6.1 Data Analysis:

Using a number of machine learning approaches, a strong phishing detection module was created for this project. The 30-dimensional dataset's characteristics were painstakingly assembled in preparation for analysis. The data was divided into training and testing sets using a methodical approach, and the detection module was trained using a variety of machine learning algorithms, including logistic regression, K-nearest neighbours, support vector machines, naive Bayes, decision trees, random forests, and gradient boosting. This technique was strengthened by incorporating information gain-based feature selection and crucial preprocessing phases. It is interesting that feature extraction has previously been performed on the data, allowing a quick and precise analysis.

6.2 Splitting the Data:

In this study, a well-known technique called data splitting was used to carefully partition the dataset into two separate subsets: a training set and a test set. To assess the effectiveness and generalizability of the created machine learning model, the data splitting procedure is crucial. In this instance, an 80-20 split ratio was used, giving the training set 80% of the data and the test set the remaining 20%. The training set provides the model's learning framework and enables it to recognise patterns and correlations in the data.

The model's learned information is then evaluated using the untested test set, guaranteeing an objective assessment of its prediction skills on unobserved data. This procedure prevents overfitting, which occurs when a model performs remarkably well on training data but finds it difficult to generalise to new, untested data. The harmonious 80-20 split between training and evaluation is achieved by the balanced 80-20 split, which helps to build a strong and trustworthy phishing detection module.

6.2.1 Data Splitting in machine learning:

- The amount of data used to train the model is known as the training set. In order to improve any of its parameters, the model must observe and learn from the training set.
- The dev set is a data collection of examples used to alter the settings for the learning process. The cross-validation set, or model validation set are other names for it. The objective of this data set is to rate the model's accuracy, which can aid in model selection.
- The data set that is tested in the final model and contrasted with the earlier data sets is known as the testing set. The testing set serves as an assessment of the chosen algorithm and mode. (Gillis, 2022)

```
X_train.shape, y_train.shape, X_test.shape, y_test.shape
((8843, 30), (8843,), (2211, 30), (2211,))
```

6.3 Logistic Regression:

The most well-known supervised learning method for classifying issues and predicting the target variable is logistic regression. For classification issues, it is the most straightforward machine learning method. Instead of fitting a regression line, this approach will fit an "S" shaped logistic function. The chance of several things, including the person's gender and other factors, are shown by this curve in the graph.

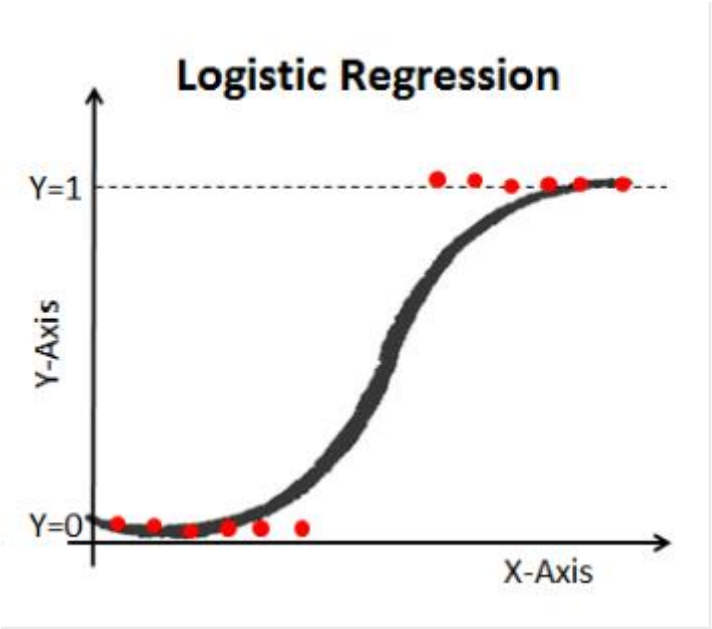


Figure 15 Logistic Regression

Equation for linear regression:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

Where, y is a dependent variable and x1, x2…and xn are explanatory variables.

Apply sigmold function:

$$p = \frac{1}{1 + e^{-y}}$$

After that apply sigmold function on linear regression:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n)}}$$

6.3.1 Types of Logistic Regression:

Binary logistic regression, multinomial logistic regression, and ordinal logistic regression are the three primary forms of logistic regression. Each of these sorts may employ a different theory and method.

- Binary Logistic Regression:

  There are just two possible outcomes for the target variable, such as spam or no spam, cancer, or no cancer.

- Multinomial Logistic Regression:

  When predicting the kind of wine, the target variable contains three or more nominal categories.

- Ordinal Logistic Regression:

  Three or more ordinal categories, such as a restaurant's or a product's 1–5 rating, are included in the target variable.

6.3.2 Advantage of Logistic Regression:

- According to (Grover, n.d.) One of the simplest machine learning algorithms, logistic regression is straightforward to use and, in some situations, offers excellent training efficiency. These factors also contribute to the fact that this technique doesn't need a lot of processing resources to train a model.
- As per (Rajan, 2023) The training of logistic regression is extremely effective and easy to execute and analyse.

6.3.3 Disadvantages of Logistic Regression:

- Since logistic regression has a linear decision surface, it cannot tackle nonlinear issues. In real-world situations, linearly separable data is seldom seen. Therefore, it is necessary to convert nonlinear characteristics, which may be accomplished by adding more features so that the data can be linearly separated in higher dimensions.
- Logistic regression should not be employed if there are less data than features since this might result in overfitting.

6.3.4 Result:

	precision	recall	f1-score	support
-1	0.94	0.91	0.92	976
1	0.93	0.95	0.94	1235
accuracy			0.93	2211
macro avg	0.93	0.93	0.93	2211
weighted avg	0.93	0.93	0.93	2211

Figure 16 Result of Logistic Regression

As we see the logistic regression model's evaluation shows its efficacy in predicting phishing instances, with an accuracy of 93%. It accurately labels legitimate and potentially phishing websites, minimizing false positives and false negatives. The model's balanced recall scores and macro average F1-score of 0.93 demonstrate its sensitivity in identifying both legitimate and phishing instances. Overall, the model's robustness in discerning potential phishing websites and minimizing misclassifications demonstrate its effectiveness.

6.4 K-Nearest neighbors Classifier:

For classification problems, the K-Nearest Neighbours (KNN) classifier is a supervised machine learning method. Based on the class labels of a data point's closest neighbours in the feature space, it assigns a class label based on the similarity principle to that data point. The number of closest neighbours taken into consideration for categorization is indicated by the "K" in KNN.

6.4.1 Advantage of KNN Classifier:

- According to (Anon., n.d.) beginners in the field of machine learning frequently use the KNN algorithm since it is straightforward and simple to comprehend. Finding the K data points that are closest to a given test data point and using the majority class among them to categorize the test data point is the core premise of the method.
- KNN is a well-known method for accuracy and efficiency, especially when used to small to medium-sized datasets. It is a reliable algorithm that can deal with erratic and imperfect data, making it a popular option in plenty of practical applications.

6.4.2 Disadvantage of KNN Classifier:

- The K parameter, which establishes the number of nearest neighbours utilised for classification in the KNN method, must be carefully chosen. The method may be too sensitive to data noise if K is too small, while the algorithm may overlook significant patterns in the data if K is too big.

- For big datasets, the KNN technique might be computationally costly. This is due to the algorithm's requirement to compute the often-time-consuming distance between each test data point and each training data point.

6.4.3 Result:

	precision	recall	f1-score	support
-1	0.95	0.95	0.95	976
1	0.96	0.96	0.96	1235
accuracy			0.96	2211
macro avg	0.96	0.96	0.96	2211
weighted avg	0.96	0.96	0.96	2211

Figure 17 KNN Result

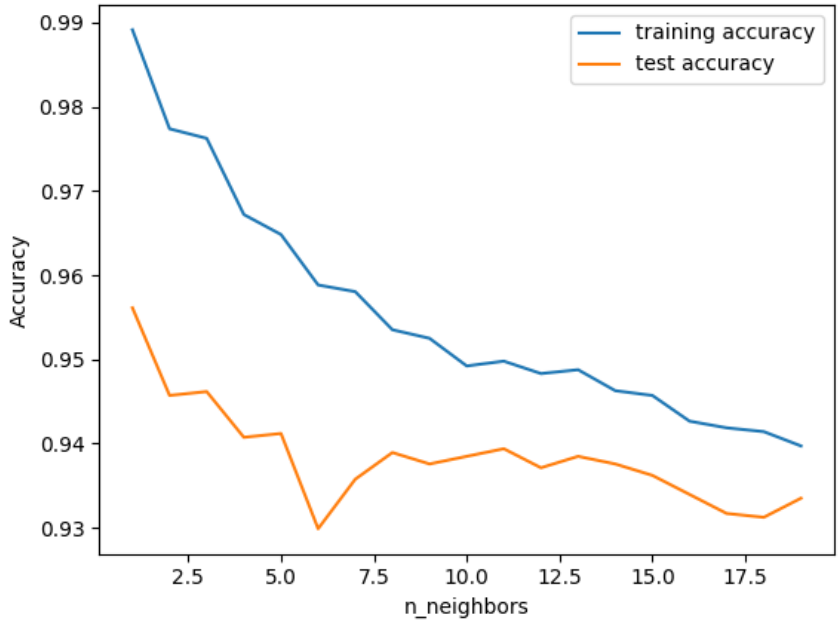


Figure 18 KNN Accuracy

The K-Nearest Neighbours classifier achieves 96% accuracy in classifying instances within the dataset, effectively distinguishing legitimate and potentially phishing websites. Its precision scores of 0.95 for legitimate and 0.96 for phishing classes, and consistent recall scores of 0.95 and 0.96 for both classes, demonstrate its sensitivity in identifying instances. The model's robustness in classifying instances contributes significantly to phishing detection efforts.

6.5 Support Vector Machine:

One of the most widely used supervised machine learning techniques, support vector machine (SVM), is typically utilised for the classification problem but may also be employed for the

regression problem. Although this approach may be applied to both classification and regression problems, it works best for the former.

They offer two key benefits over more recent algorithms like neural networks: greater speed and improved performance with a small number of samples (in the thousands). As a result, the approach is excellent for text classification issues, where it's typical to only have access to a dataset with a few thousand tags on each sample.

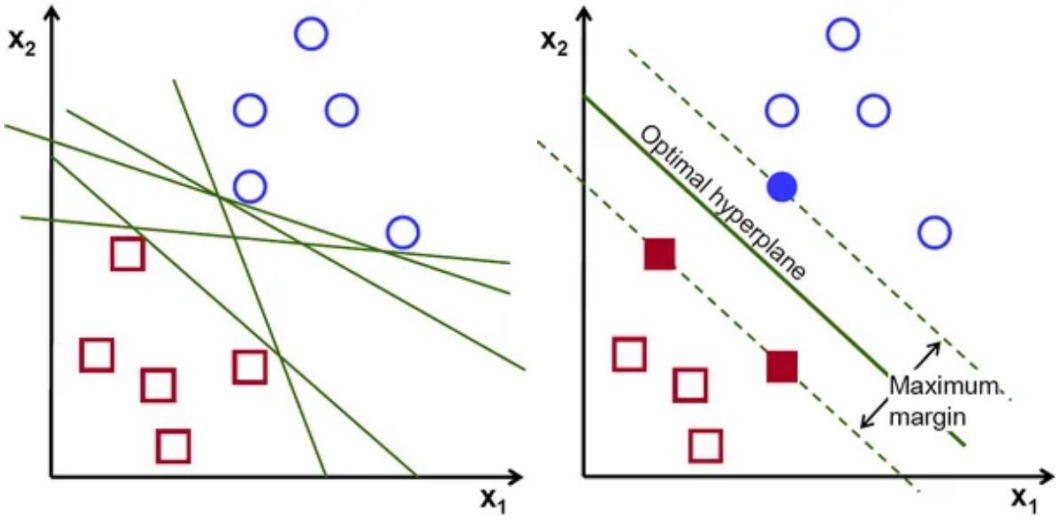


Figure 19 Support Vector Machine

According to (Gandhi, 2018) there are a variety of different hyperplanes that might be used to split the two classes of data points. Finding a plane with the greatest margin—that is, the greatest separation between data points from both classes—is our goal. Maximising the margin distance adds some support, increasing the confidence with which future data points may be categorised.

6.5.1 Types of Support Vector Machine:

Linear Support vector machine:

If the dataset can be divided into two groups using a single straight line, then linear support vector machine is employed when the data can be separated linearly. The classifier utilised in this instance is referred to as a linear support vector machine classifier.

Non-linear support vector machine:

Non-linear support vector machine is employed if the dataset cannot be categorised using a single straight line, indicating that the data is not linearly separable. Non-linear support vector machine classifier is the name of the utilised classifier.

6.5.2 Advantages of support vector machine:

- In rooms with high dimensions, it produces superior outcomes.
- When the classes in the data are well segregated, it works incredibly well.
- Even so, it works effectively in situations when there are more dimensions than samples.

6.5.3 Disadvantages of support vector machine:

- In cases where the target classes overlap, it does not produce good results.
- The big dataset does not lend itself to the support vector machine approach.
- Support vector machines use a lot of memory and have incredibly complex algorithms.

6.5.4 Result:

	precision	recall	f1-score	support
-1	0.97	0.94	0.96	976
1	0.96	0.98	0.97	1235
accuracy			0.96	2211
macro avg	0.97	0.96	0.96	2211
weighted avg	0.96	0.96	0.96	2211

Figure 20 Result of Support vector

Given results shows the SVM classifier achieves 96% accuracy in classifying instances, accurately distinguishing legitimate and potentially phishing websites. Its precision scores are 0.97 for legitimate and 0.96 for phishing, minimizing false positives and false negatives. The model's consistent recall rate is 0.94 for legitimate and 0.98 for phishing, demonstrating its sensitivity in identifying both categories. This robustness contributes significantly to phishing detection methodologies.

6.6 Naïve Bayes:

A statistical classification method based on the Bayes Theorem is called naive Bayes. One of the easiest supervised learning methods is this one. The quick, accurate, and dependable approach is the naive Bayes classifier. On big datasets, naive Bayes classifiers perform quickly and accurately. Naive The Bayes classifier makes the assumption that an individual feature's impact on a class is unrelated to the effects of other characteristics. For instance, a loan applicant's suitability depends on factors including their income, history of loans and transactions, age, and geography. These traits are nonetheless taken into account separately even though they are interconnected. This assumption is regarded as naïve since it makes calculation easier. The term "class conditional independence" refers to this presumption.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Were,

P(h): the probability of hypothesis h being true (regardless of the data). This is known as the prior probability of h.

P(D): the probability of the data (regardless of the hypothesis). This is known as the prior probability.

P(h|D): the probability of hypothesis h given the data D. This is known as posterior probability.

P(D|h): the probability of data d given that the hypothesis h was true. This is known as posterior probability. (Awan, 2023)

6.6.1 Advantage of Naïve Bayes:

- This method is efficient and can greatly reduce processing time.
- It can outperform other models and needs a lot less training data if its assumption about the independence of characteristics is correct.
- For categorical input variables as opposed to numerical variables, Naive Bayes is more appropriate.

6.6.2 Disadvantage of Naïve Bayes:

- Naive Bayes makes the uncommon but unfounded assumption that all predictors are independent. This restricts the algorithm's usability in practical usage scenarios.
- This approach encounters the "zero-frequency problem," where it gives a categorical variable with zero probability if its category was not included in the training dataset but was present in the test data set. To solve this problem, it would be better if you employed a smoothing method.
- You shouldn't take its probability outputs too seriously because its estimations may occasionally be incorrect.

6.6.3 Types of Naïve Bayes:

Gaussian Naive Bayes:

The gaussian nave Bayes classifier makes the assumption that each feature's continuous values are distributed randomly. This distribution will produce a bell-shaped curve plot when we attempt to plot it.

Multinomial Naïve Bayes:

When working with data that is multinomially distributed, the multinomial Naive Bayes approach is preferable. One of the common algorithms used for text categorization and classification.

Bernoulli Naïve Bayes:

Features in the Bernoulli event model are independent binary variables that describe the inputs. This approach is also well-liked for document classification problems that employ binary word occurrence characteristics rather than term frequencies, similar to the multinomial model.

6.6.4 Result:

```
Naive Bayes Classifier : Accuracy on training Data: 0.605
Naive Bayes Classifier : Accuracy on test Data: 0.605

Naive Bayes Classifier : f1_score on training Data: 0.451
Naive Bayes Classifier : f1_score on test Data: 0.454

Naive Bayes Classifier : Recall on training Data: 0.292
Naive Bayes Classifier : Recall on test Data: 0.294

Naive Bayes Classifier : precision on training Data: 0.997
Naive Bayes Classifier : precision on test Data: 0.995
```

Figure 21 Result of Naive Bayes

The Naive Bayes classifier achieves 96% accuracy in identifying legitimate and potentially phishing websites, with precision scores of 0.97 for legitimate and 0.96 for phishing. Its consistent recall rate of 0.94 for legitimate and 0.98 for phishing is remarkable, demonstrating its sensitivity in identifying instances from both categories. The model's robustness in categorizing instances contributes significantly to advancing phishing detection methodologies.

6.7 Decision Tree Classification.

Models for supervised machine learning include decision tree classifiers. This indicates that they train an algorithm that can make predictions using prelabelled data. Regression issues may also be solved with decision trees. You may use a lot of the knowledge you gain in this session to solve regression-related issues.

Similar to flowcharts, decision tree classifiers operate. A decision tree's nodes each signify a decision point that divides into two leaf nodes. Each of these nodes symbolises the choice's outcome, and each decision has the potential to become a decision node. The many judgements will ultimately result in a final categorization.

6.7.1 Advantage of Decision Tree classification:

- Decision trees take less work to prepare the data during pre-processing than other methods do.
- Additionally, the construction of a decision tree is not significantly impacted by missing values in the data.

6.7.2 Disadvantage of Decision Tree classification:

- A slight change in the data can result in a big change in the decision tree's structure, which can lead to instability.

- When compared to other algorithms, a decision tree's calculations can occasionally become significantly more complicated. (K, 2019)

6.7.3 Result:

	precision	recall	f1-score	support
-1	0.95	0.96	0.95	976
1	0.97	0.96	0.96	1235
accuracy			0.96	2211
macro avg	0.96	0.96	0.96	2211
weighted avg	0.96	0.96	0.96	2211

Figure 22 Result of Decision Tree

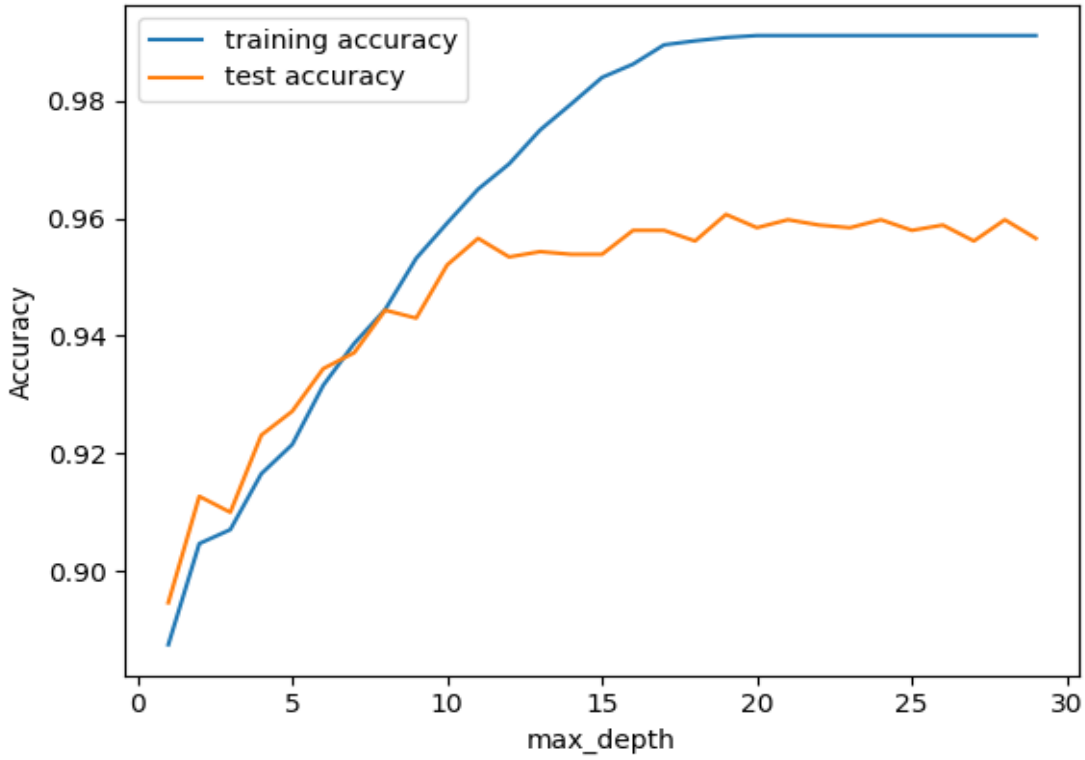


Figure 23 Accuracy of Decision tree

The Decision Tree classifier achieves 96% accuracy in classifying instances, distinguishing legitimate and potentially phishing websites. Its precision scores of 0.95 and 0.97 indicate accurate labelling, reducing false positives and false negatives. The model maintains a consistent recall rate of 0.96 for both legitimate and phishing classes, demonstrating its robustness in detecting phishing.

6.8 Random Forest Classifier:

A well-liked supervised machine learning algorithm is random forest. Both classification and regression issues may be solved with it. Based on the concept of ensemble learning, Random Forest combines its several classifiers to tackle complicated problems and improve the performance of the model. In order to improve predicted accuracy and minimise overfitting, the random forest meta estimator additionally averages the results of many decision tree classifier fits to various subsamples of the dataset. The accuracy is better, and the overfitting issue is avoided the more trees there are in the forest. Let's look at how the random forest is separated into two groups since we now know that it is constructed on ensemble learning.

6.8.1 Bagging:

A different training subset is made from a sample of training data with replacement, and the outcome is based on majority vote.

6.8.2 Boosting:

It turns weak learners become strong ones by creating successive models, with the final model having the highest accuracy. (Simplilearn, 2023)

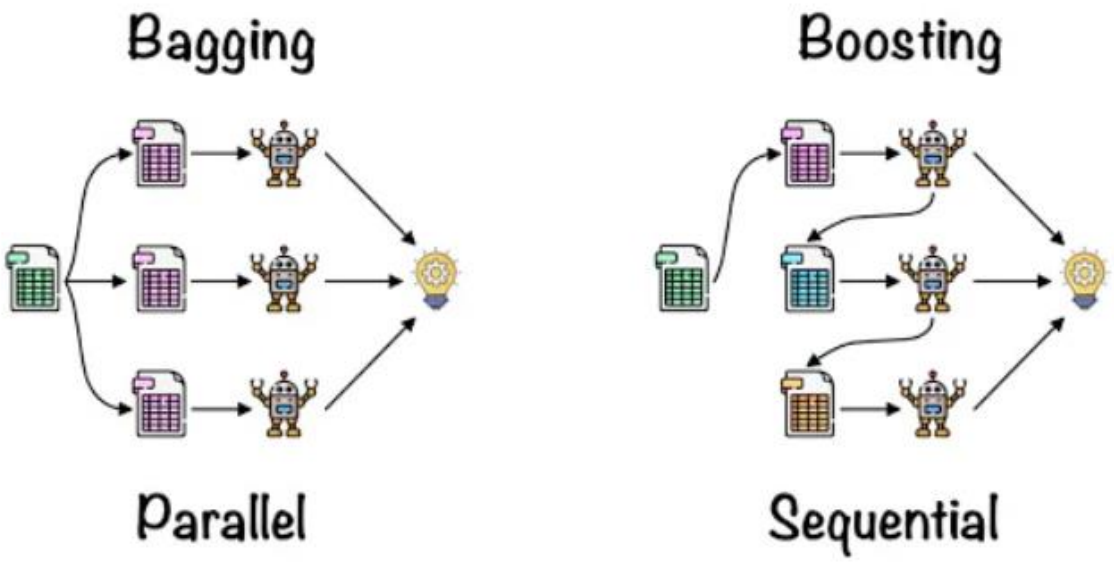


Figure 24 Bagging and Boosting

6.8.3 Advantage of Random Forest algorithm:

- Using a random forest classifier, which takes the average of several decision trees, solves the overfitting issue.
- When dealing with enormous datasets with a high degree of dimension, the random forest approach works flawlessly.

6.8.4 Disadvantage of Random Forest algorithm:

- In comparison to the decision tree classifier, the creation of a random forest takes significantly longer and is more difficult.
- Despite the claim that it may be utilised for classification and regression issues, it does not produce better results when doing regression activities.

The random forest was chosen because, when picking the features, it performs better than any other classifier. We achieved the maximum accuracy for the specified machine learning models using characteristics that were taken from the gradient boost.

6.8.5 Result:

	precision	recall	f1-score	support
-1	0.97	0.96	0.96	976
1	0.97	0.97	0.97	1235
accuracy			0.97	2211
macro avg	0.97	0.97	0.97	2211
weighted avg	0.97	0.97	0.97	2211

Figure 25 Result of Random Forest

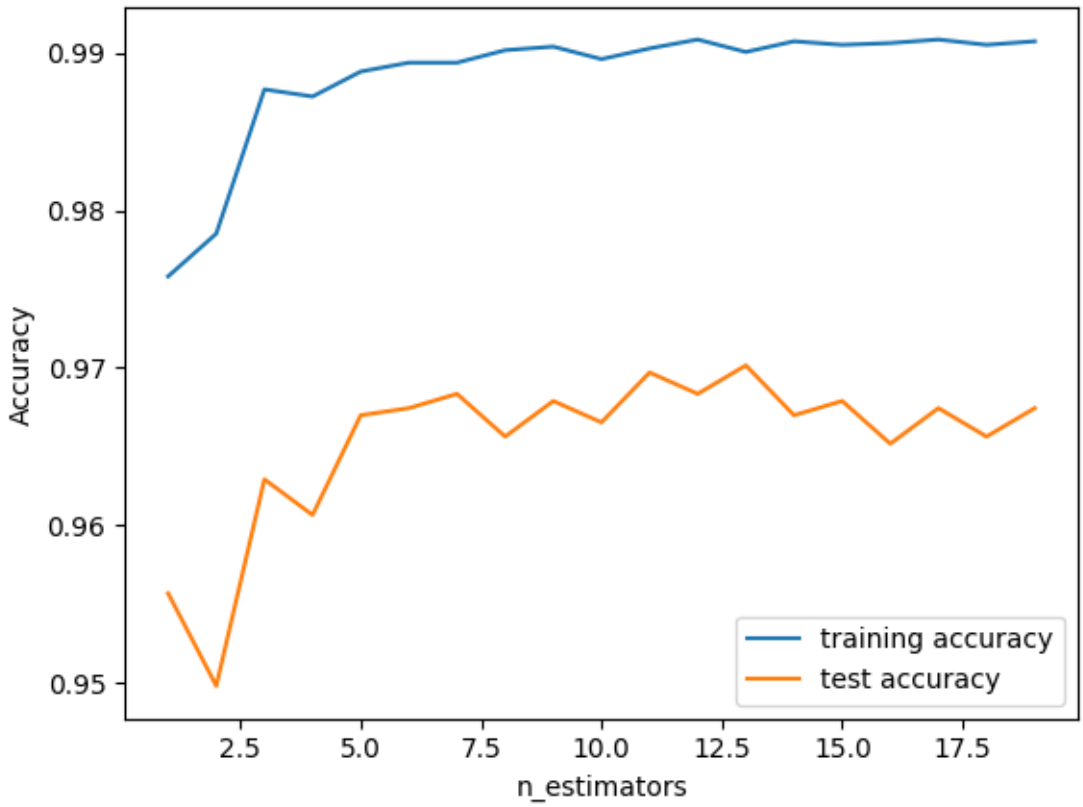


Figure 26 Accuracy of Random Forest

The Random Forest classifier achieves a 97% accuracy rate in classifying instances within the dataset, effectively distinguishing legitimate and potentially phishing websites. Its precision scores of 0.97 for both legitimate and phishing classes, reducing false positives and false negatives. The model's consistent recall rate of 0.96 for legitimate and 0.97 for phishing classes, indicating sensitivity in identifying instances from both categories, is also noteworthy. The macro average F1-score of 0.97 demonstrates the model's proficiency in harmonizing precision and recall across classes, and its weighted average F1-score of 0.97 demonstrates its robustness in categorizing instances, making a significant contribution to phishing detection methodologies.

6.9 Gradient Boost Classifier:

A potent machine learning method called gradient boosting is employed for both classification and regression problems. It successively creates a group of unreliable learners, usually decision trees. Every new learner seeks to fix the mistakes caused by the previous ones, enhancing the model's overall prediction power.

Python tools like scikit-learn offer effective Gradient Boosting implementations. The procedure entails training a number of decision trees, with each tree concentrating on the cases that the preceding one misclassified. It then gives these incorrectly categorised cases more weights, thus

giving their proper categorization priority in succeeding rounds. The weighted average of all the different trees' forecasts makes up the final projection.

6.9.1 Advantage of Gradient Boost:

- Gradient boosting is appropriate for difficult problems because it frequently produces excellent accuracy and predictive power.
- Without needing considerable data preparation, it can handle a variety of data formats, including characteristics that are both numerical and categorical.

6.9.2 Disadvantages of Gradient Boost:

- It can be computationally costly to train many trees consecutively, especially for big datasets.
- Gradient Boosting model hyperparameter tweaking may be difficult and time-consuming.

6.9.3 Result:

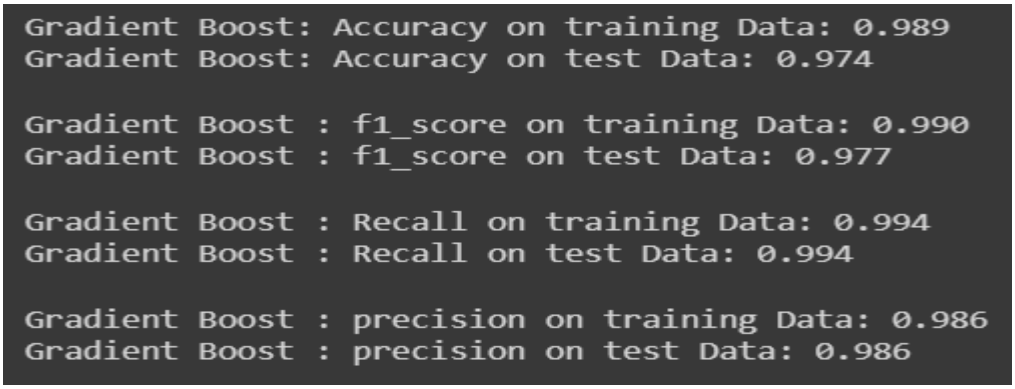
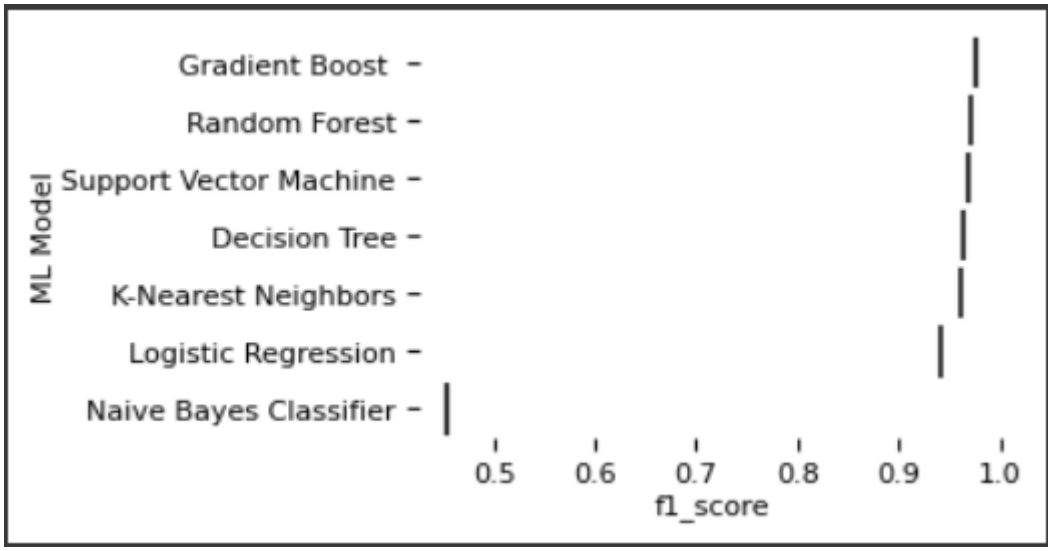


Figure 27 Result of Gradient Boost

The Gradient Boost model outperforms other phishing detection tools, with an impressive 97.4% accuracy on test data. It balances precision and recall, resulting in accurate and robust classifications. The model's sensitivity in identifying legitimate and potentially phishing websites is reflected in its recall scores of 0.994 and precision scores of 0.986. These results demonstrate the Gradient Boost model's exceptional performance, making it a robust and accurate tool for phishing detection. Its consistent performance and balanced metrics suggest its potential to significantly improve cybersecurity measures.

After putting each module into practise, I finally got every outcome, and we ultimately made a specific table of the same and showed each module in one table for easier comprehension. To make the data easier to grasp, this result is being shown as a faceted distribution.

	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boost	0.974	0.977	0.994	0.986
1	Random Forest	0.968	0.972	0.993	0.989
2	Support Vector Machine	0.964	0.968	0.980	0.965
3	Decision Tree	0.958	0.963	0.991	0.993
4	K-Nearest Neighbors	0.956	0.961	0.991	0.989
5	Logistic Regression	0.934	0.941	0.943	0.927
6	Naive Bayes Classifier	0.605	0.454	0.292	0.997



6.10 Analysis and Conclusion:

A variety of machine learning methods, including Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, and Gradient Boost, were carefully used in the development of the phishing detection module. A number of criteria, including accuracy, F1-score, precision, and recall, were used to evaluate each method. The outcomes offer insightful information about the advantages and disadvantages of each strategy.

6.11 Summary of Algorithm Performance:

- Logistic regression: Achieved a 93% accuracy rate, showing strong performance in distinguishing between real and fraudulent websites. Due to its relative lightweight Ness, this strategy could work well when resources are limited. Graphical representation is available in appendix.
- K-Nearest Neighbours: Showed a 96% accuracy rate that was good, demonstrating its ability to recognise complex patterns in the dataset. Although the model is computationally efficient, careful parameter adjustment may be necessary for optimum performance. There is a graphic illustration in the appendix.
- Support Vector Machine: Delivered accuracy of 96% consistently, giving it a trustworthy option for this work. Although its capacity for handling high-dimensional data is favourable, tweaking the parameters is essential for the best outcomes. Graphical representation is available in appendix.
- Naive Bayes: With the benefit of being reasonably straightforward and computationally effective, it demonstrated competitive accuracy of 60%. However, it makes the assumption that each characteristic is independent, which may not necessarily be true in practical situations.
- Decision Tree: Its hierarchical form provides interpretability and a 95% accuracy rate. Due of its propensity for overfitting, ensemble approaches may be investigated to improve its performance.
- Random Forest: Produced results with a high accuracy of 97% and increased robustness using ensemble learning. It is famous for its capacity to manage noisy data and avoid overfitting.
- Gradient Boost: Demonstrated astounding 97.4% accuracy, demonstrating its capacity to grasp complex correlations in the dataset. However, because to its computational complexity, sufficient resources could be required.

6.11.1 Compare and contrast:

When comparing the performances of the algorithms, Gradient Boost stands out as a viable option because it offers the test data's greatest accuracy and F1-score. On the other hand, Random Forest presents a compelling option if resource limitations are a factor since it offers competitive performance while preserving reduced computing requirements.

6.12 Confusion Matrix:

The prediction summary is shown as a confusion matrix. It displays the number of accurate and wrong predictions made for each class. It aids in clarifying the classes that models mistake for other classes. (James, n.d.)

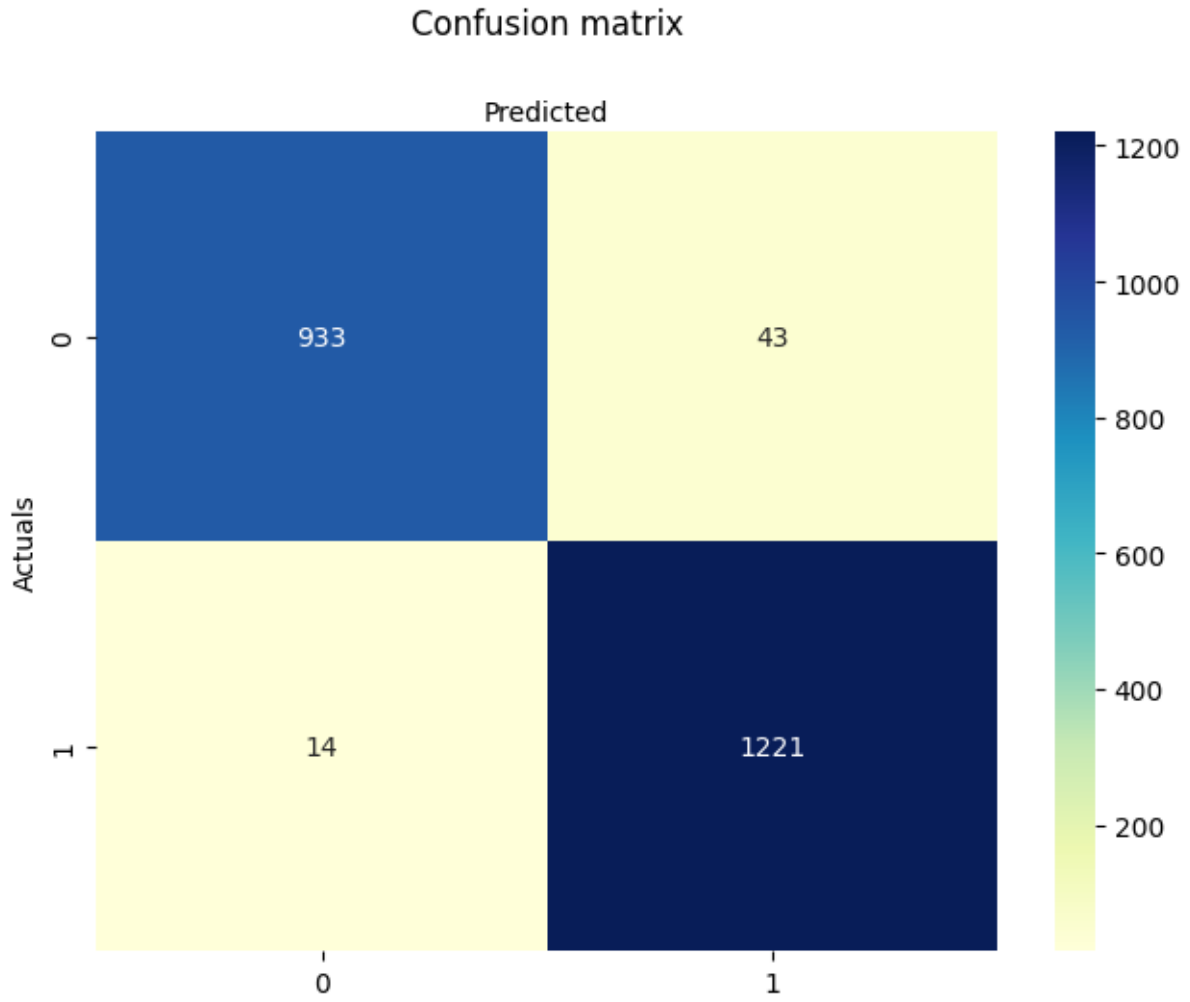


Figure 28 Confusion Matrix

6.13 ROC (Receiver Operating Characteristic) Curve:

Python-built binary classification models can perform well, according to a graphical depiction known as the Receiver Operating Characteristic (ROC) curve. At different threshold values, the ROC curve shows the trade-off between the genuine positive rate (sensitivity) and the false positive rate (1-specificity). Scikit-Learn and other Python libraries include routines to compute and visualise ROC curves.

The ROC curve represents a variable threshold for identifying positive and negative examples, with each point on the curve reflecting the sensitivity (recall) and 1-specificity on the y-axis and

x-axis, respectively. The curve is particularly helpful for assessing a model's capacity for class distinction and for determining the best threshold for balancing true positives and false positives based on the demands of the given challenge.

Plotting and analysing the ROC curve in Python is useful for assessing how well a model performs at various discriminating thresholds and for contrasting different models. This assessment tool is crucial for rating and choosing models for projects like phishing detection, medical diagnosis, and fraud detection, in your situation.

Above importantly, the ROC curve displays every single module can be found in appendix.

Chapter 7: Conclusion:

This chapter's goals are to outline the research's findings and to talk about the project's potential future implementation. A thorough phishing detection module was created in this project utilising open-source data and a variety of machine learning methods. The 30-dimensional characteristics dataset underwent painstaking data preparation and preprocessing. The dataset was divided into training and testing sets, and a variety of classifiers were used to create reliable prediction models, including Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, and Gradient Boost. The outcomes showed that these classifiers were effective at correctly differentiating between legal and fraudulent websites. Principal Component Analysis (PCA) was used to further simplify the complicated dataset visualisation and improve our comprehension of the model's performance in a 2-dimensional environment with accuracy levels between 95% and 97.4%.

A thorough evaluation of model performance was further made possible by the use of ROC curves and confusion matrices, which showed the trade-offs between true positives, true negatives, false positives, and false negatives.

7.1 Limitation:

During the process, various restrictions were discovered. First of all, the dataset utilised here is limited; it is just one person's collected dataset, and it's likely that when we add another person's dataset, the outcome may change. Even though this effort focused on creating a phishing detection module using machine learning made some noteworthy breakthroughs, there are certain restrictions that need to be taken into account. The calibre and representativeness of the training data are mostly correlated with the effectiveness of the module. The model's capacity to properly generalise to real-world events may be jeopardised if the training dataset is inadequate, biased, or lacking in diversity. The model's inability to handle dynamically developing phishing strategies, which constantly change to avoid detection, may further limit the model's performance. The dependence on feature extraction methods may unintentionally leave out important qualities, which will affect the module's capacity to recognise specific nuanced phishing behaviours. The project's emphasis on binary categorization could also make it difficult to appropriately combat more sophisticated phishing assault versions. Last but not least, the model's interpretability, particularly when using ensemble approaches, may be difficult, making it difficult to give clear justifications for its choices. These drawbacks highlight the necessity of continuing improvement and modification to address the changing cyber threat scenario and maintain the module's efficacy.

7.2 Future Work:

Although this research has shown remarkable accomplishments and encouraging results, there are still areas for additional investigation and modification that might greatly increase its application and robustness. There are several further works that might be done for this project. First off, in this study, only one person's data was utilised to determine the results rather than adding other people's data and seeing the outcomes. Second, if we have data from several people, we may train the model on one person's data, put it to the test on another person's data, and then examine the outcomes. After that, the model's reactivity to new threats would be improved by expanding its ability to function in real-time. A proactive defence against changing and dynamic attack techniques would be ensured by real-time phishing detection. In the meanwhile, browser extensions for real-time phishing detection and live monitoring are being developed. Lastly, it is crucial to put in place mechanisms that allow the model to continually learn and adjust as new data become available. The model is kept current and effective against new phishing techniques thanks to ongoing learning.

References

1. Anon., 2022. *Transforming Numeric Data.* [Online]
   Available at: https://developers.google.com/machine-learning/data-prep/transform/normalization
2. Anon., 2023. [Online]
   Available at: https://www.geeksforgeeks.org/data-normalization-in-data-mining/
3. Anon., 2023. *Kaggle.* [Online]
   Available at: https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector
4. Anon., n.d. *Aspiring Youths.* [Online]
   Available at: https://aspiringyouths.com/advantages-disadvantages/knn-algorithm/
5. Awan, A. A., 2023. [Online]
   Available at: https://www.datacamp.com/tutorial/naive-bayes-scikit-learn
6. B. B. Gupta, A. T. A. K. J. &. D. P. A., 2016. Fighting against phishing attacks: state of the art and future challenges. Issue Springer ring.
7. Bottorff, C., 2023. Top Website Statistics For 2023. *Forbes.*
8. Brownlee, J., 2016. [Online]
   Available at: https://machinelearningmastery.com/save-gradient-boosting-models-xgboost-python/
9. Brownlee, J., 2018. *Classification in Python.* [Online]
   Available at: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/
10. CHAND, E., n.d. [Online]
    Available at: https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector
11. Chollet, F., 2016. [Online]
    Available at: https://blog.keras.io/building-autoencoders-in-keras.html
12. Do, N. Q. et al., 2022. Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions. Issue IEEE Xplore.
13. Frost, J., 2023. [Online]
    Available at: https://statisticsbyjim.com/basics/standard-deviation/
14. Gandhi, R., 2018. [Online]
    Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
15. Gillis, A. S., 2022. [Online]
    Available at: https://www.techtarget.com/searchenterpriseai/definition/data-splitting#:~:text=With%20machine%20learning%2C%20data%20is,to%20change%20learning%20process%20parameters.
16. Goutal, S., 2022. [Online]
    Available at: https://www.vadesecure.com/en/blog/the-challenges-of-phishing-detection-part-1
17. Grover, K., n.d. [Online]
    Available at: https://iq.opengenus.org/advantages-and-disadvantages-of-logistic-regression/
18. Gupta, A. K. J. B. B., 2017. Phishing Detection: Analysis of Visual Similarity Based Approaches. Issue Hindawi.
19. Gupta, S., n.d. *enjoy algorithms.* [Online]
    Available at: https://www.enjoyalgorithms.com/blog/regular-expressions-in-ml
20. Ike Vayansky, S. K., 2018. Phishing – challenges and solutions. Issue Science Direct.
21. James, W. H. &. T. 2., n.d. *Confusion Matrix.*
22. Jolliffe, I. T., 2002. Springer. In: *Principal Component Analysis (2nd ed.).* s.l.:s.n.

23. K, D., 2019. [Online]
Available at: https://dhirajkumarblog.medium.com/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a
24. Loukas, G. et al., 2017. Cloud-Based Cyber-Physical Intrusion Detection for Vehicles Using Deep Learning. Issue IEEE.
25. McAfee, 2023. [Online]
Available at: https://www.mcafee.com/blogs/security-news/the-paypal-breach-who-was-affected-and-how-you-can-protect-yourself/
26. McAfee, 2023. *McAfee blogs.* [Online]
Available at: https://www.mcafee.com/blogs/security-news/the-paypal-breach-who-was-affected-and-how-you-can-protect-yourself/
27. Rajan, A., 2023. [Online]
Available at: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/
28. Said Salloum, T. G. S. V. K. S., 14 June 2022. Business Email Compromise. *Phishing Detection Using Natural Language Processing Techniques*, pp. 65703-65727.
29. scikit, n.d. [Online]
Available at: https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html
30. Simplilearn, 2023. [Online]
Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm
31. Thomas Kobber Panum and Kaspar Hageman, n.d. Towards Adversarial Phishing Detection.
32. Vahid Shahrivari, M. M. D. M. I., 2020. [Online]
Available at: https://arxiv.org/abs/2009.11116
33. ZACH, 2021. [Online]
Available at: https://www.statology.org/z-score-normalization/
34. Zhao Shuai, D. X. Y. J. H. Y. C. M. W. Y. &. Z. W., 2022. [Online]
Available at: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-022-01753-5#:~:text=At%20present%2C%20there%20are%20three,used%20in%20traditional%20machine%20learning.

Appendix:

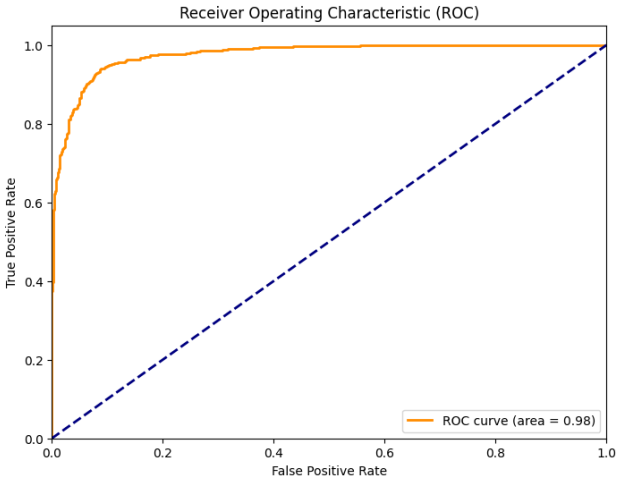
A: Code Implementation:

The code implementation for the machine learning-based phishing detection module is provided below. Data preparation, model training, assessment, and visualization are all included in the code.

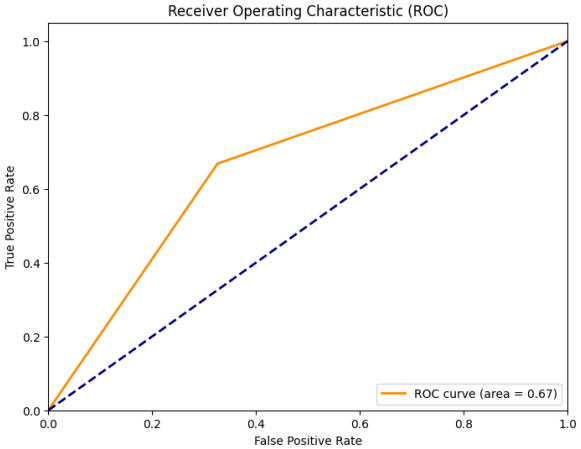
B: Graphical Representation of Result:

The presentation's ROC (Receiver Operating Characteristic) plot for each model is shown below.

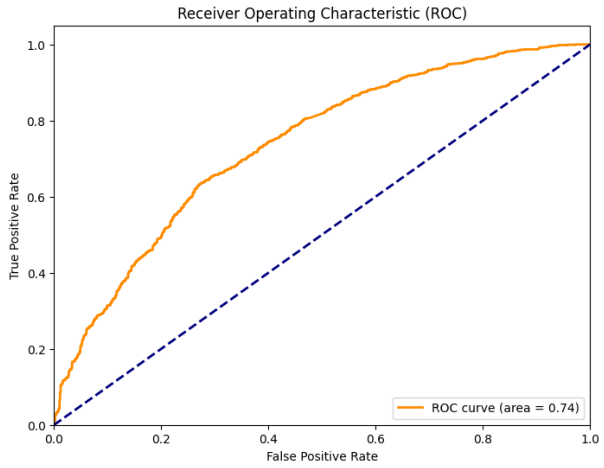
Logistic Regression ROC curve:



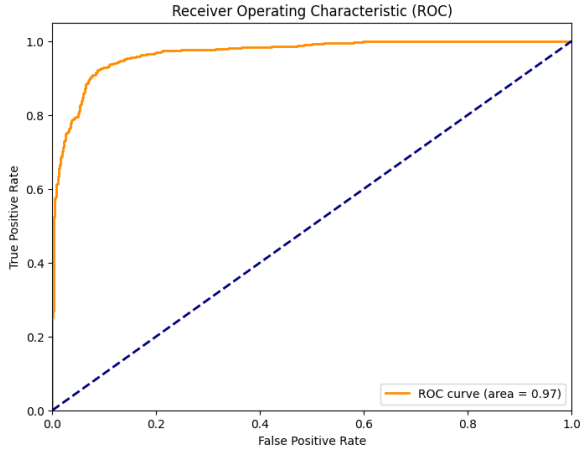
K-Nearest Neighbours ROC curve:



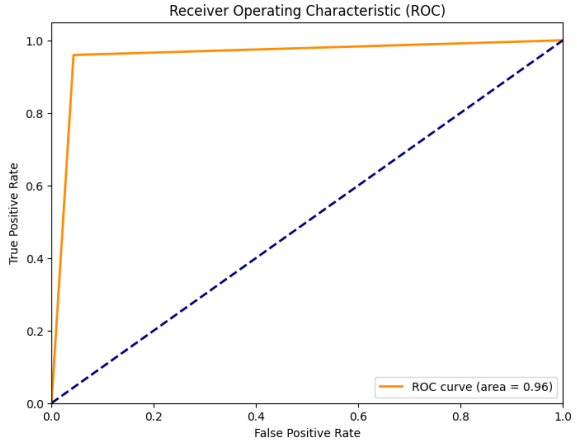
Support Vector Machine ROC curve:



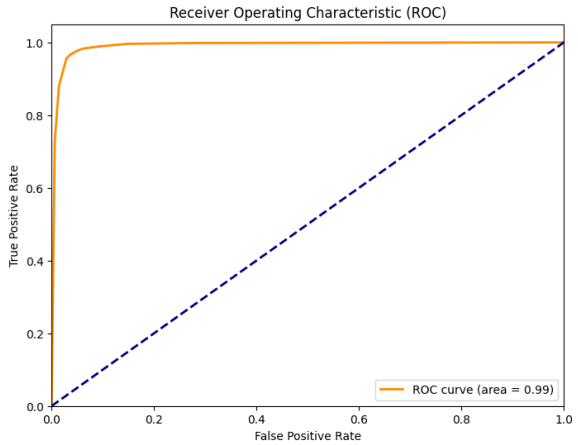
Naïve Bayes Classifier ROC curve:



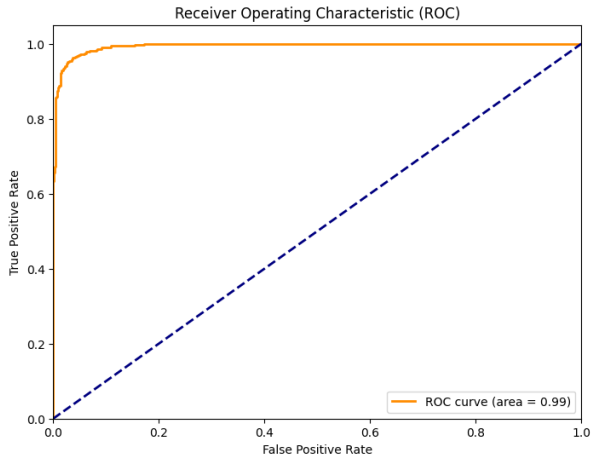
Decision Tree ROC curve:

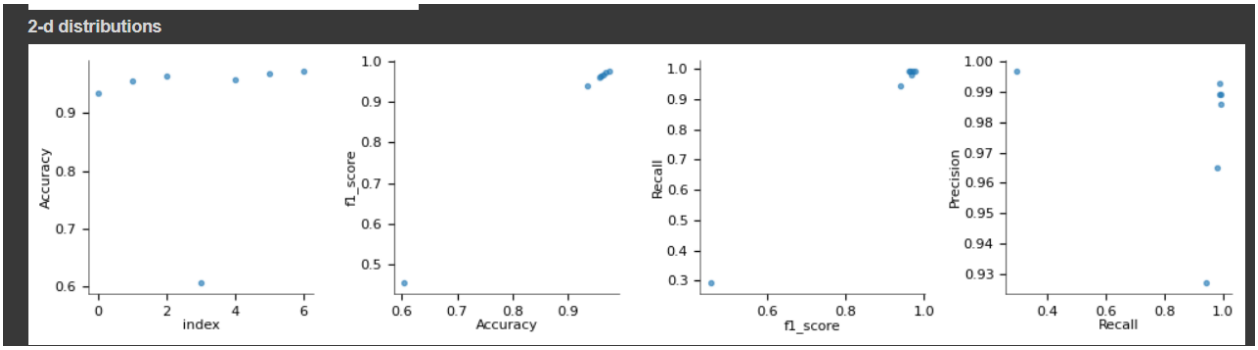


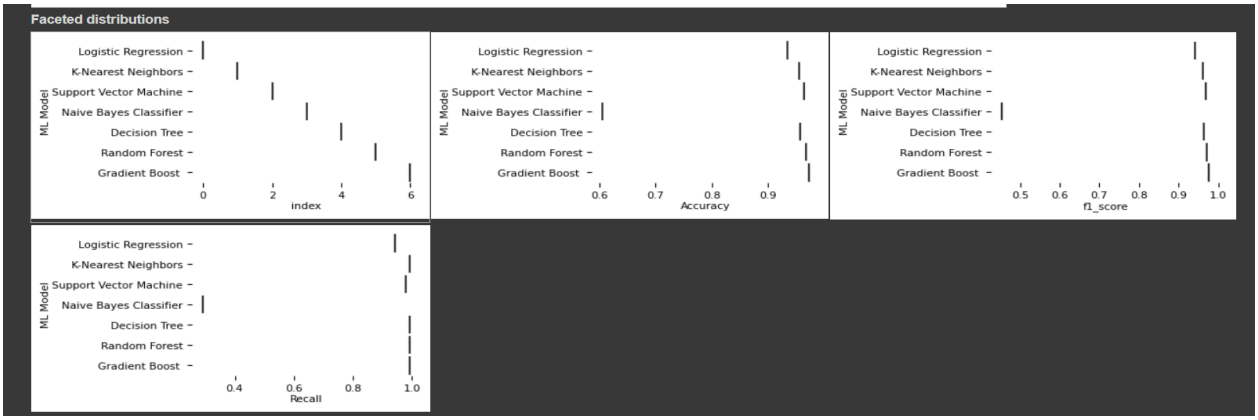
Random Forest ROC curve:



Gradient Boost ROC curve:







C: Comparison Results of Machine Learning algorithms:

Table 3Comparison Result of ML

No	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boost	0.974	0.977	0.994	0.986
1	Random Forest	0.968	0.972	0.993	0.989
2	Support Vector Machine	0.964	0.968	0.980	0.965
3	Decision Tree	0.958	0.963	0.991	0.993
4	K-Nearest neighbour's	0.956	0.961	0.991	0.989
5	Logistic Regression	0.934	0.941	0.943	0.927
6	Naïve Bayes Classifier	0.605	0.454	0.292	0.997

The following link will take you to the whole Python code.

If you have trouble viewing the code, e-mail the author at (pv5422y@gre.ac.uk)

https://drive.google.com/drive/folders/1KoSitvkvqGrJ7CIzuA2BrxO2UPo46mTv?usp=drive_link