A Project Report on

Phishing Website Detection Using Machine Learning

Submitted in partial fulfillment of award of

BACHELOR OF TECHNOLOGY

Degree
in

COMPUTER SCIENCE AND ENGINEERING

By

AYUSH SHARMA -(1808210039)
AKSHITA VERMA -(1808210017)
ANUBHAV MISHRA -(1808210033)
AVNISH KUMAR -(1808210037)

Session: (2018-2022)

Mr. PRAVEEN SAINI
(Assistant Professor)

SUPERVISOR

IN PURSUIT OF EXCELLENCE

Department of Computer Science & Engineering
Moradabad Institute of Technology
Moradabad (U.P.)
2021-2022

CERTIFICATE

Certified that the Project Report entitled **“Phishing Websites Detection Using Machine Learning”** submitted by **Ayush Sharma (1808210039), Akshita Verma (1808210017), Anubhav Mishra (1808210033), Avnish Kumar (1808210037)** is their own work and has been carried out under my supervision. It is recommended that the candidates may now be evaluated for their project work by the University.

Date: **(Mr. Praveen Saini)**

Assistant Professor

ABSTRACT

The phishing attack is the simplest way to obtain sensitive information from innocent users. The aim of the phishers is to acquire critical information like username, password, and bank account details. Cyber security persons are now looking for trustworthy and steady detection techniques for phishing websites detection. This paper deals with machine learning technology for the detection of phishing URLs by extracting and analysing various features of legitimate and phishing URLs. Decision Trees, random forest and support vector machine algorithms are used to detect phishing websites. The aim of the paper is to detect phishing URLs as well as narrow them down to the best machine learning algorithm by comparing the accuracy rate, false positive, and false-negative rate of each algorithm. Phishing Detection is the prevention of cybercrime in which phishing aims to collect sensitive and personal information such as usernames, passwords, credit card numbers, and even money by impersonating a legitimate website. A measurement for phishing detection is the number of suspicious e-mails reported to the security team. This measurement is designed to evaluate the number of employees who followed the proper procedure for reporting suspicious messages. Simple spelling mistakes, broken English, grammatical errors, or low-resolution images should act as a red flag that you

are on a phishing site and should leave immediately. Another area of the website that may indicate a phishing site is the lack of a “contact us” section. We present a phishing detection system using machine learning in which we are collecting legitimate website and phishing website and check how many sites can be fake website by some common features of website URLs.

ACKNOWLEDGEMENT

It is our privilege to express our sincerest regards to our project coordinator, **Mr. Praveen Saini (Assistant Professor)**, for his valuable input, able guidance, encouragement, whole-hearted cooperation, and constructive criticism throughout the duration of our project. We take this opportunity to thank **Mr. Zubair Iqbal(Assistant Professor), Dr. Manish Gupta (Associate Professor), Mr. Himanshu Agarwal (Assistant Professor)** and all our teachers who have directly or indirectly helped our project. We would like to express our gratitude towards our parents for their constant support and encouragement which help us in completion of this project. Our thanks and appreciation also go to our colleagues in developing the project and people who are willing to help with their abilities.

AYUSH SHARMA

AKSHITA VERMA

ANUBHAV MISHRA

AVNISH KUMAR

TABLE OF CONTENTS

Content **Page No**

LIST OF FIGURES

Figure No. **Page No**

CHAPTER 1

INTRODUCTION

With the widespread usage of the internet for online banking and trade, phishing attacks and forms of identity theft-based frauds are becoming extremely popular among hacker communities. In 2004 alone, more than fifty million phishing emails were sent. Their result was ten billion dollars of damage to banks and financial institutions. Most of the recent phishing attacks are carried out as a three-step process. In the first step, the phishers send emails to their victims from social engineering attacks, webpages, and forums. Large volumes of phishing emails with legal banking domains are sent out using anonymous servers or compromised machines.[1] These emails contain hyperlinks with an appearance like a legitimate website. The fake webpage contains input forms requesting personal critical information such as credit card, social security numbers, mother's maiden name, etc. Although existing spam filtering techniques can be employed to combat phishing emails, these measures are not entirely scalable. Several readily available tools can bypass both the statistical and rule-based spam filters. As these mechanisms are not uniquely tuned for the detection of phishing emails despite their existence, the threats caused by phishing emails are prevalent. Furthermore, unlike spamming, which impacts bandwidth, phishing attacks directly affect their victims by inflicting a thefty loss due to monetary damage. Moreover, attackers can use technical vulnerabilities to construct socially engineered messages (i.e., use of legitimate, but spoofed, domain names can be far more persuasive than using different domain names), which makes phishing attacks a severe problem. Effective mitigation would require addressing issues at the technical and human layers. Since phishing attacks aim at exploiting weaknesses found in humans (i.e., system end-users), it is difficult to mitigate them. For example, as evaluated, end-users failed to detect 29% of phishing attacks, even when trained with the best performing user awareness program. On the other hand, software phishing detection techniques are evaluated against phishing attacks, which makes their five performances practically unknown with targeted forms of phishing attacks. These limitations in phishing mitigation techniques have almost resulted in security breaches against several organizations, including leading information security providers. Now days, there are so many people are being aware of using internet to perform various activities like online shopping, online bill payment, online mobile recharge, banking transaction. Due to wide use of this customer face various security threats like cybercrime. There are many cybercrimes that are widely performed for example spam, fraud, cyber terrorisms, and phishing. Among this phishing is new cybercrime and very popular nowadays. Phishing is fraud attempt, which performed to obtain sensitive information of user. Phisher design website which looks same as any legitimate site and spoof user for obtaining private information of user such as username, password, banking details for miscellaneous reasons. According to APWG 2Q report, the total number of phish detected in 2Q 2018 was 233,040, compared to 263,538 in 1Q 2018. These totals exceed the 180,577 observed in 4Q 2017 and the 190,942 seen in 3Q 2017. There were increases in SAAS/webmail targeted sector with 21% of overall phishing attack. Payment sector is continuing as most attractive target for phishing. According to APWG 1Q report, the total number of phish detected in 1Q 2018 was 263,538. This was up 46 percent from the 180,577 observed in 4Q 2017. It was also significantly more than the 190,942 seen in 3Q 2017. The number of unique phishing reports submitted to APWG during 1Q 2018 was 262,704, compared to 233,613 in 4Q 2017 and 296,208 in 3Q 2017.

1.1 Problem Statement

Nowadays Phishing becomes a main area of concern for security researchers because it is not difficult to create the fake website which looks so close to legitimate website. Experts can identify fake websites but not all the users can identify the fake website and such users become the victim of phishing attack. Main aim of the attacker is to steal banks account credentials. Phishing attacks are becoming successful because lack of user awareness. Since phishing attack exploits the weaknesses found in users, it is very difficult to mitigate them, but it is very important to enhance phishing detection techniques. There are too many approaches to detect a phishing website we will use Machine Learning to do. We use few machine learning algorithms like Decision Tree Classifier, Random Forest and XGBoost.

1.2 Industry / Society benefited

a) The Project can be implemented by many E-Commerce or other websites in order to have good customer relationship.
b) Users can make online payment securely.
c) Eliminate the cyber threat risk level.
d) Increase user alertness to phishing risks.
e) In still a cyber security culture and create cyber security heroes.
f) Change behaviours to eliminate the automatic trust response.

1.3 Proposed Solution

Machine learning technique detects phishing sites based on markup visualization. Machine learning models trained on the visual representation of website code can help improve the accuracy and speed of detecting phishing websites. Use anti-phishing protection and anti-spam software to protect yourself when malicious messages slip through to your computer. Anti-malware is included to prevent other types of threats. Similar to anti-spam software, anti-malware software is programmed by security researchers to spot even the stealthiest malware.

1.4 Feasibility

The Phishing Detection System is the best technique to distinguish between the website is legitimate or a phishing website. Machine learning technique detects phishing sites based on markup visualization. Machine learning models trained on the visual representation of website code can help improve the accuracy and speed of detecting phishing websites.

1.5 Literature Survey

Rao et al. [2] proposed a novel classification approach that use heuristic based feature extraction approach. In this, they have classified extracted features into three categories such as URL Obfuscation features, Third-Party-based features, Hyperlink-based features. Moreover, proposed technique gives 99.55% accuracy. Drawback of this is that as this model uses third-party features, classification of website dependent on speed of third-party services. And also this model is purely depends on the quality and quantity of the training set and Broken links feature extraction has a limitation of more execution time for the websites with more number of links.

Chunlin et al. [3] proposed approach that primarily focus on character frequency features. In this paper, they have combined statistical analysis of URL with machine learning technique to get result that is more accurate for classification of malicious URLs. Also they have compared six machine-learning algorithms to verify the effectiveness of proposed algorithm which gives 99.7% precision with false positive rate less than 0.4%.

Sudhanshu et al. [4] used association data mining approach. They have proposed rule based classification technique for phishing website detection. They have concluded that association classification algorithm is better than any other algorithms because of their simple rule transformation. They achieved 92.67% accuracy by extracting 16 features but this is not up to mark so proposed algorithm can be enhanced for efficient detection rate.

M. Amaad et al. [5] presented a hybrid model for classification of phishing website. In this paper, proposed model carried out in two phase. In phase 1, they individually perform classification techniques, and select the best three models based on high accuracy and other performance criteria. While in phase 2, they further combined each individual model with best three model and makes hybrid model that gives better accuracy than individual model. They achieved 97.75% accuracy on testing dataset. There is limitation of this model that it requires more time to build hybrid model.

Hossein et al. [6] developed an open-source framework known as “Fresh-Phish”. For phishing websites, machine-learning data can be created using this framework. In this, they have used reduced features set and using python for building query. They build a large labelled dataset and analyse several machine-learning classifiers against this dataset. Analysis of this gives very good accuracy using machine learning classifiers. These analyses how long time it takes to train the model.

Gupta et al. [7] proposed a novel anti phishing approach that extracts features from client-side only. Proposed approach is fast and reliable as it is not dependent on third party but it extracts features only from URL and source code. In this paper, they have achieved 99.09% of overall detection accuracy for phishing website. This paper have concluded that this approach has limitation as it can detect webpage written in HTML. Non-HTML webpage cannot detect by this approach.

Bhagyashree et al. [8] proposed a feature based approach to classify URL’s as phishing and non-phishing. Various features this approach uses are lexical features, WHOIS features, Page Rank and Alexa rank and Phish Tank-based features for disguising phishing and non-phishing website.

Mustafa et al. [9] developed safer framework for detecting phishing website. They have extracted URL features of website and using subset based selection technique to obtain better accuracy. In this paper, author evaluated CFS subset based and content based subset selection methods and Machine learning algorithms are used for classification purpose.

Priyanka et al. [10] proposed novel approach by combining two or more algorithms. In this paper, author has implemented two algorithm Adaline and Backpropotion along with SVM for getting good detection rate and classification purpose.

Pradeepthi et al. [11] In this paper, author studied different classification algorithm and concluded that tree-based classifiers are best and gives better accuracy for phishing URL detection. The author also uses various features such as lexical features, URL based feature, network based feature and domain based feature.

Luong et al. [12] proposed new technique to detect phishing website. In proposed method, Author used six heuristics that are primary domain, sub domain, path domain, page rank, and alexa rank, alexa reputation whose weight and values are evaluated. This approach gives 97% accuracy but still improvement can be done by enhancing more heuristics.

Ahmad et al. [13] proposed three new features to improve accuracy rate for phishing website detection. In this paper, Author used both type of features as commonly known and new features for classification of phishing and non-phishing site. At the end author has concluded this work can be enhanced by using this novel feature with decision tree machine learning classifiers.

Mohammad et al. [14] proposed model that automatically extracts important features for phishing website detection without requiring any human intervention. Author has concluded in this paper that the process of extracting feature by their tool is much faster and reliable than any manual extraction.

CHAPTER 2

MACHINE LEARNING

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.

Although machine learning is a field within computer science, it differs from traditional computational approaches. [15] In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

2.1 What is Machine Learning?

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning is an important component of the growing field of data science. Using statistical methods, algorithms are trained to make classifications or predictions, uncovering key insights within data mining projects. These insights subsequently drive decision-making within applications and businesses, ideally impacting key growth metrics as shown in fig 2.1. As big data continues to expand and grow, the market demand for data scientists will increase, requiring them to assist in the identification of the most relevant business questions and subsequently the data to answer them. A neat diagram of machine learning is shown below in fig. 2.1.

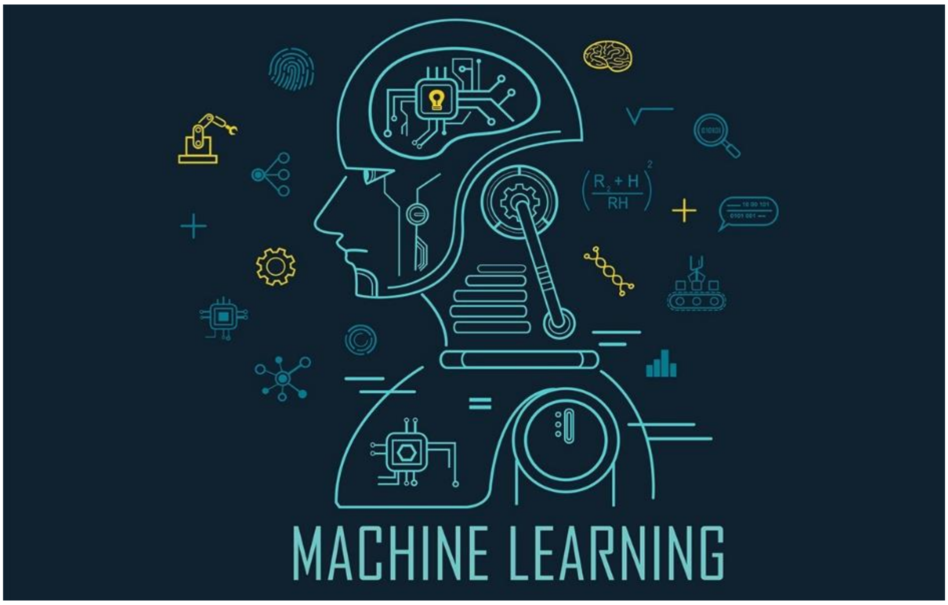

Fig 2.1 Machine Learning

2.2 History of Machine Learning

Before some years (about 40-50 years), machine learning was science fiction, but today it is the part of our daily life. Machine learning is making our day to day life easy from **self-driving cars** to **Amazon virtual assistant "Alexa"**. However, the idea behind machine learning is so old and has a long history. Below some milestones are given which have occurred in the history of machine learning:

a) The early history of Machine Learning (Pre-1940):

   i. 1834: In 1834, Charles Babbage, the father of the computer, conceived a device that could be programmed with punch cards. However, the machine was never built, but all modern computers rely on its logical structure.

   ii. 1936: In 1936, Alan Turing gave a theory that how a machine can determine and execute a set of instructions.

b) The era of stored program computers:

   i. 1940: In 1940, the first manually operated computer, "ENIAC" was invented, which was the first electronic general-purpose computer. After that stored program computer such as EDSAC in 1949 and EDVAC in 1951 were invented.

   ii. 1943: In 1943, a human neural network was modelled with an electrical circuit. In 1950, the scientists started applying their idea to work and analysed how human neurons might work.

c) Computer machinery and intelligence:

   i. 1950: In 1950, Alan Turing published a seminal paper, "Computer Machinery and Intelligence," on the topic of artificial intelligence. In his paper, he asked, "Can machines think?"

d) Machine intelligence in Games:

   i. 1952: Arthur Samuel, who was the pioneer of machine learning, created a program that helped an IBM computer to play a checkers game. It performed better more it played.

   ii. 1959: In 1959, the term "Machine Learning" was first coined by Arthur Samuel.

e) The first "AI" winter:

   i. The duration of 1974 to 1980 was the tough time for AI and ML researchers, and this duration was called as AI winter.In this duration, failure of machine translation occurred, and people had reduced their interest from AI, which led to reduced funding by the government to the researches.

f) Machine Learning from theory to reality

   i. 1959: In 1959, the first neural network was applied to a real-world problem to remove echoes over phone lines using an adaptive filter.

   ii. 1985: In 1985, Terry Sejnowski and Charles Rosenberg invented a neural network NETtalk, which was able to teach itself how to correctly pronounce 20,000 words in one week.

   iii. 1997: The IBM's Deep blue intelligent computer won the chess game against the chess expert Garry Kasparov, and it became the first computer which had beaten a human chess expert.

g) Machine Learning at 21st century

   i. 2006: In the year 2006, computer scientist Geoffrey Hinton has given a new name to neural net research as "deep learning," and nowadays, it has become one of the most trending technologies.

   ii. 2012: In 2012, Google created a deep neural network which learned to recognize the image of humans and cats in YouTube videos.

iii. 2014: In 2014, the Chabot "Eugen Goostman" cleared the Turing Test. It was the first Chabot who convinced the 33% of human judges that it was not a machine.

iv. 2014: DeepFace was a deep neural network created by Facebook, and they claimed that it could recognize a person with the same precision as a human can do.

v. 2016: AlphaGo beat the world's number second player Lee sedol at Go game. In 2017 it beat the number one player of this game Ke Jie.

vi. 2017: In 2017, the Alphabet's Jigsaw team built an intelligent system that was able to learn the online trolling. It used to read millions of comments of different websites to learn to stop online trolling.

h) Machine Learning at present:

i. Now machine learning has got a great advancement in its research, and it is present everywhere around us, such as self-driving cars, Amazon Alexa, Catboats, recommender system, and many more. It includes Supervised, unsupervised, and reinforcement learning with clustering, classification, decision tree, SVM algorithms, etc.

ii. Modern machine learning models can be used for making various predictions, including weather prediction, disease prediction, stock market analysis, etc.

2.3 How does Machine Learning Work?

UC Berkeley (link resides outside IBM) breaks out the learning system of a machine learning algorithm into three main parts.

A Decision Process: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabelled, your algorithm will produce an estimate of a pattern in the data.

An Error Function: An error function serves to evaluate the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model. A Model Optimization Process: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example. and the model estimate. The algorithm will repeat this evaluate and optimize process, updating weights autonomously until a threshold of accuracy has been met.

2.4 Literature Survey

A core objective of a learner is to generalize from its experience. The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory. Because training sets are finite and the future is uncertain, learning theory usually does not yield guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common. The bias–variance decomposition is one way to quantify generalization error.

For the best performance in the context of generalization, the complexity of the hypothesis should match the complexity of the function underlying the data. If the hypothesis is less complex than the function, then the model has underfit the data. If the complexity of the model is increased in response, then the training error decreases. But if the hypothesis is too complex, then the model is subject to overfitting and generalization will be poorer.

In addition to performance bounds, learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results. Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

2.5 The Challenges Facing Machine Learning

While there has been much progress in machine learning, there are also challenges. For example, the mainstream machine learning technologies are black-box approaches, making us concerned about their potential risks. To tackle this challenge, we may want to make machine learning more explainable and controllable. As another example, the computational complexity of machine learning algorithms is usually very high and we may want to invent lightweight algorithms or implementations. Furthermore, in many domains such as physics, chemistry, biology, and social sciences, people usually seek elegantly simple equations (e.g., the Schrödinger equation) to uncover the underlying laws behind various phenomena. Machine learning takes much more time. You have to gather and prepare data, then train the algorithm. There are much more uncertainties. That is why, while in traditional website or application development an experienced team can estimate the time quite precisely, a machine learning project used for example to provide product recommendations can take much less or much more time than expected. Why? Because even the best machine learning engineers don't know how the deep learning networks will behave when analysing different sets of data. It also means that the machine learning engineers and data scientists cannot guarantee that the training process of a model can be replicated.

2.6 Features of Machine Learning

a) Machine learning uses data to detect various patterns in a given dataset.
b) It can learn from past data and improve automatically.
c) It is a data-driven technology.
d) Machine learning is much similar to data mining as it also deals with a huge amount of data.

2.7 Types of Machine Learning

There are three types of machine learning as shown in fig 2.7:

a) Supervised
b) Unsupervised
c) Reinforcement

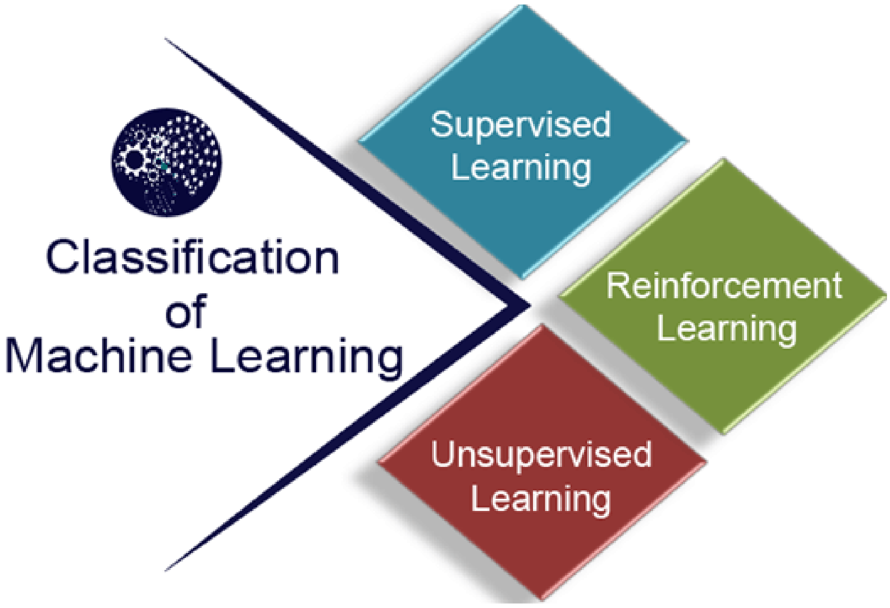

Fig 2.7 Types of Machine Learning

2.7.1 Supervised Learning

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labelled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross-validation process. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox.

2.7.2 Unsupervised Learning

Unsupervised learning refers to the use of artificial intelligence (AI) algorithms to identify patterns in data sets containing data points that are neither classified nor labelled. Unsupervised learning is commonly used for finding meaningful patterns and groupings inherent in data, extracting generative features, and exploratory purposes.

2.7.3 Reinforcement Learning

Reinforcement learning is a machine learning training method based on rewarding desired behaviours and/or punishing undesired ones. In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions, and learn through trial and error.

2.8 Machine Learning Lifecycle

Machine learning has given the computer systems the abilities to automatically learn without being explicitly programmed. But how does a machine learning system work? So, it can be described using the life cycle of machine learning. Machine learning life cycle is a cyclic process to build an efficient machine learning project. The main purpose of the life cycle is to find a solution to the problem or project. A neat diagram of machine learning lifecycle is shown below in fig. 2.8.

Machine learning life cycle involves seven major steps, which are given below:

a) Gathering Data
b) Data preparation
c) Data Wrangling
d) Analyse Data
e) Train the model
f) Test the model
g) Deployment

The most important thing in the complete process is to understand the problem and to know the purpose of the problem. Therefore, before starting the life cycle, we need to understand the problem because a good result depends on a better understanding of the problem. In the complete life cycle process, to solve a problem, we create a machine learning system called a "model", and this model is created by providing "training". But to train a model, we need data, hence, the life cycle starts by collecting data.

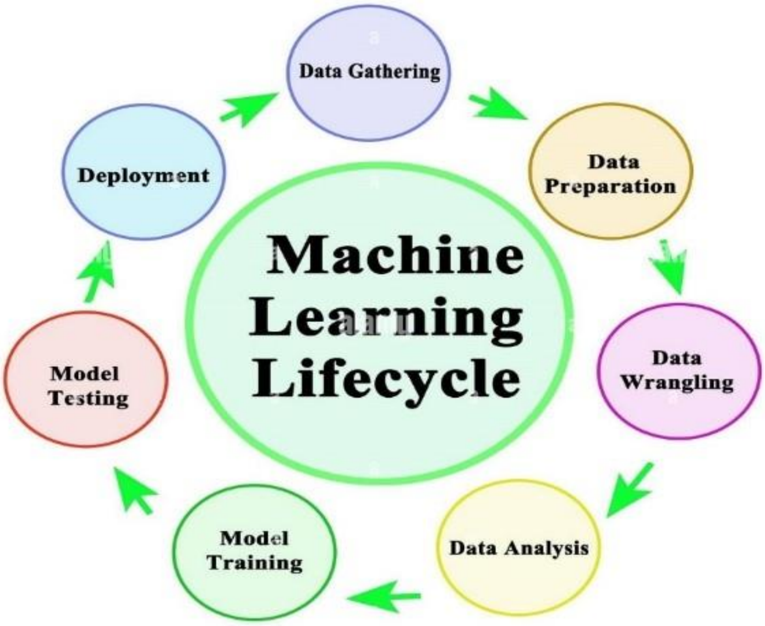

Fig. 2.8 Machine Learning Lifecycle

The most important thing in the complete process is to understand the problem and to know the purpose of the problem. Therefore, before starting the life cycle, we need to understand the problem because a good result depends on a better understanding of the problem. In the complete life cycle process, to solve a problem, we create a machine learning system called a "model", and this model is created by providing "training". But to train a model, we need data, hence, the life cycle starts by collecting data:

a) Data Gathering

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.

In this step, we need to identify the different data sources, as data can be collected from various sources such as **files**, **databases**, the **internet**, or **mobile devices**. It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction.

b) Data preparation

After collecting the data, we need to prepare it for further steps. Data preparation is a step where we put our data into a suitable place and prepare it to use in our machine learning training.

c) Data Wrangling

Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues.

d) Analyse Data

The aim of this step is to build a machine learning model to analyse the data using various analytical techniques and review the outcome. It starts with the determination of the type of the problems, where we select the machine learning techniques such as Classification, Regression, Cluster analysis, Association, etc. then build the model using prepared data, and evaluate the model.

e) Train the model

Now the next step is to train the model, in this step we train our model to improve its performance for a better outcome of the problem. We use datasets to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules, and features.

f) Test the model

Once our machine learning model has been trained on a given dataset, then we test the model. In this step, we check for the accuracy of our model by providing a test dataset to it. Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

g) Deployment

The last step of machine learning life cycle is deployment, where we deploy the model in the real-world system. If the above-prepared model is producing an accurate result as per our requirement with acceptable speed, then we deploy the model in the real system. But before deploying the project, we will check whether it is improving its performance using available data or not. The deployment phase is similar to making the final report for a project.

2.9 Applications of Machine Learning

a) **Web Search Engine:** One of the reasons why search engines like google, bing etc work so well is because the system has learnt how to rank pages through a complex learning algorithm.

b) **Photo tagging Applications:** Be it Facebook or any other photo tagging application, the ability to tag friends makes it even more happening. It is all possible because of a face recognition algorithm that runs behind the application.

c) **Spam Detector:** Our mail agent like Gmail or Hotmail does a lot of hard work for us in classifying the mails and moving the spam mails to spam folder. This is again achieved by a spam classifier running in the back end of mail application.

d) **Database Mining for growth of automation:** Typical applications include Web-click data for better UX, Medical records for better automation in healthcare, biological data and many more.

e) **Applications that cannot be programmed:** There are some tasks that cannot be programmed as the computers we use are not modelled that way. Examples include Autonomous Driving, Recognition tasks from unordered data (Face Recognition/ Handwriting Recognition), Natural language Processing, computer Vision etc.

f) **Understanding Human Learning:** This is the closest we have understood and mimicked the human brain. It is the start of a new revolution, The real AI. Now, after a brief insight lets come to a more formal definition of Machine Learning

2.10 Future Scope

` Future of Machine Learning is as vast as the limits of human mind. We can always keep learning, and teaching the computers how to learn. And at the same time, wondering how some of the most complex machine learning algorithms have been running in the back of our own mind so effortlessly all the time. There is a bright future for machine learning. Companies like Google, Quora, and Facebook hire people with machine learning. There is intense research in machine learning at the top universities in the world. The global machine learning as a service market is rising expeditiously mainly due to the Internet revolution. The process of connecting the world virtually has generated vast amount of data which is boosting the adoption of machine learning solutions. Considering all these applications and dramatic improvements that ML has brought us, it doesn't take a genius to realize that in coming future we will definitely see more advanced applications of ML, applications that will stretch the capabilities of machine learning to an unimaginable level.

CHAPTER 3

MACHINE LEARNING LANGUAGES

Machine learning is a growing area of computer science and several programming languages support ML framework and libraries. Among all of the programming languages, Python is the most popular choice followed by C++, Java, JavaScript, and C#.

3.1 Python – The New Generation Language

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for an emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

3.1.1 Features

a) **Interpreted:** In Python there is no separate compilation and execution steps like C/C++. It directly run the program from the source code. Internally, Python converts the source code into an intermediate form called bytecodes which is then translated into native language of specific computer to run it.

b) **Platform Independent:** Python programs can be developed and executed on the multiple operating system platform. Python can be used on Linux, Windows, Macintosh, Solaris and many more.

c) **Multi- Paradigm:** Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect oriented programming.

d) **Simple:** Python is a very simple language. It is a very easy to learn as it is closer to English language. In python more emphasis is on the solution to the problem rather than the syntax.

e) **Rich Library Support:** Python standard library is very vast. It can help to do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.

f) **Free and Open Source:** Firstly, Python is freely available. Secondly, it is open-source. This means that its source code is available to the public. We can download it, change it, use it, and distribute it. This is called FLOSS (Free/Libre and Open-Source Software). As the Python community, we're all headed toward one goal- an ever-bettering Python.

3.1.2 Why Python Is a Suitable Language for Machine Learning?

a) **A great library ecosystem:** A great choice of libraries is one of the main reasons Python is the most popular programming language used for AI. A library is a module or a group of modules published by different sources which include a pre-written piece of code that allows users to reach some functionality or perform different actions. Python libraries provide base level items so developers don't have to code them from the very beginning every time. ML requires continuous data processing, and Python's libraries let us access, handle and transform data. These are some of the most widespread libraries you can use for ML and AI:

i. **Scikit-learn** for handling basic ML algorithms like clustering, linear and logistic regressions, regression, classification, and others. o Pandas for high-level data structures and analysis. It allows merging and filtering of data, as well as gathering it from other external sources like Excel, for instance.

ii. **Keras** for deep learning. It allows fast calculations and prototyping, as it uses the GPU in addition to the CPU of the computer. o TensorFlow for working with deep learning by setting up, training, and utilizing artificial neural networks with massive datasets.

iii. **Matplotlib** for creating 2D plots, histograms, charts, and other forms of visualization. o NLTK for working with computational linguistics, natural language recognition, and processing.

iv. **Scikit** image for image processing.

v. **PyBrain** for neural networks, unsupervised and reinforcement learning.

vi. **Caffe** for deep learning that allows switching between the CPU and the GPU and processing 60+ mln images a day using a single NVIDIA K40 GPU.

vii. **StatsModels** for statistical algorithms and data exploration.

In the PyPI repository, we can discover and compare more python libraries.

b) **A low entry barrier:** Working in the ML and AI industry means dealing with a bunch of data that we need to process in the most convenient and effective way. The low entry barrier allows more data scientists to quickly pick up Python and start using it for AI development without wasting too much effort into learning the language. In addition to this, there's a lot of documentation available, and Python's community is always there to help out and give advice.

c) **Flexibility:** Python for machine learning is a great choice, as this language is very flexible:

i. It offers an option to choose either to use OOPs or scripting.

ii. There's also no need to recompile the source code, developers can implement any changes and quickly see the results.

iii. Programmers can combine Python and other languages to reach their goals.

d) **Good Visualization Options:** For AI developers, it's important to highlight that in artificial intelligence, deep learning, and machine learning, it's vital to be able to represent data in a human-readable format. Libraries like Matplotlib allow data scientists to build charts, histograms, and plots for better data comprehension, effective presentation, and visualization. Different application programming interfaces also simplify the visualization process and make it easier to create clear reports.

e) **Community Support:** It's always very helpful when there's strong community support built around the programming language. Python is an open-source language which means that there's a bunch of resources open for programmers starting from beginners and ending with pros. A lot of Python documentation is available online as well as in Python communities and forums, where programmers and machine learning developers discuss errors, solve problems, and help each other out. Python programming language is absolutely free as is the variety of useful libraries and tools.

f) **Growing Popularity:** As a result of the advantages discussed above, Python is becoming more and more popular among data scientists. According to Stack Overflow, the popularity of Python is predicted to grow until 2020, at least. This means it's easier to search for developers and replace team players if required. Also, the cost of their work maybe not as high as when using a less popular programming language.

3.2 Introduction to R

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, …) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open-Source route to participation in that activity.

One of R’s strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation’s GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

3.2.1 Features

a) **Open-source**

R is an open-source software environment. It is free of cost and can be adjusted and adapted according to the user’s and the project’s requirements. You can make improvements and add packages for additional functionalities. R is freely available. You can learn how to install R, Download and start practicing it.

b) **Strong Graphical Capabilities**

R can produce static graphics with production quality visualizations and has extended libraries providing interactive graphic capabilities.

c) **Highly Active Community**

R has an open-source library which is supported by its growing number of users. The R environment is continuously growing. This growth is due to its large user-base.

d) **A Wide Selection of Packages**

CRAN or Comprehensive R Archive Network houses more than 10,000 different packages and extensions that help solve all sorts of problems in data science. High-quality interactive graphics, web application development, quantitative analysis or machine learning procedures, there is a package for every scenario available. R contains a sea of packages for all the forms of disciplines like astronomy, biology, etc. While R was originally used for academic purposes, it is now being used in industries as well.

e) Comprehensive Environment

R has a very comprehensive development environment meaning it helps in statistical computing as well as software development. R is an object-oriented programming language. It also has a robust package called Rshiny which can be used to produce full-fledged web apps. Combined with data analysis and data visualization, R can be used for highly interactive online data-driven storytelling.

f) Can Perform Complex Statistical Calculations

R can be used to perform simple and complex mathematical and statistical calculations on data objects of a wide variety. It can also perform such operations on large data sets.

g) Running Code Without a Compiler

R is an interpreted language which means that it does not need a compiler to make a program from the code. R directly interprets provided code into lower-level calls and pre-compiled code.

h) Data Variety

R can handle a variety of structured and unstructured data. It also provides various data modelling and data operation facilities due to its interaction with databases.

i) Cross-platform Support

Cross Platform compatible with R. R is machine-independent. It supports the cross-platform operation. Therefore, it can be used on many different operating systems.

3.2.2 Why R Is a Suitable Language for Machine Learning?

a) It provides good explanatory code. For example, if you are at the early stage of working with a machine learning project and you need to explain the work you do, it becomes easy to work with R language comparison to python language as it provides the proper statistical method to work with data with fewer lines of code.

b) R language is perfect for data visualization. R language provides the best prototype to work with machine learning models.

c) R language has the best tools and library packages to work with machine learning projects. Developers can use these packages to create the best pre-model, model, and post-model of the machine learning projects. Also, the packages for R are more advanced and extensive than python language which makes it the first choice to work with machine learning projects.

d) Suitable for Analysis — if the data analysis or visualization is at the core of your project then R can be considered as the best choice as it allows rapid prototyping and works with the datasets to design machine learning models.

e) The bulk of useful libraries and tools — Similar to Python, R comprises of multiple packages which help to improve the performance of the machine learning projects. For instance — Caret boosts the machine learning capabilities of the R with its special set of functions which helps to create predictive models efficiently. R developers gain advantage from the advanced data analysis packages which cover the pre- and post-modelling stages which are directed at specific tasks like model validation or data visualization.

f) Suitable for exploratory work — If you require any exploratory work in statistical models at the beginning stages of your project then R makes it easier to write them as the developers just need to add a few lines of code.

CHAPTER 4

SRS DOCUMENT

A software requirements specification (SRS) is a document explaining how and what the software/system will do. It defines the features and functionality that the product requires to satisfy all stakeholders' (business, users) needs.

4.1 Introduction

4.1.1 PURPOSE

The main purpose of preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system. This Document plays a vital role in the software development life cycle (SDLC), and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during the testing phase. Any changes made to the requirements in the future will have to go through a formal change approval process.

4.1.2 SCOPE

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

4.1.3 OBJECTIVE

The main objective of this project is to make the awareness among people that how the legitimate website and phishing website are different which are looking same in the page layout. Generally, the home page of the website will be same and when we click on the links of the page then they redirect to us on phishing webpage. Also, those links are not stopped they repeatedly open another webpage which is a symbol of that webpage might be a phishing webpage.

4.2 Specific Requirements

4.2.1 Hardware Requirements

a) Processor – Intel Xeon E2630 v4 – 10 core processor, 2.2 GHz with Turboboost up to 3.1 GHz.
b) Motherboard – ASRock EPC612D8A.
c) RAM – 128 GB DDR4 2133 MHz
d) 2 TB Hard Disk (7200 RPM) + 512 GB SSD.
e) GPU – NVidia TitanX Pascal (12 GB VRAM)
f) Intel Heatsink to keep the temperature under control.

4.2.2 Software Requirements

a) Anaconda Application

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free. 50 Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

b) Visual studio Application

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

c) Python Libraries (NumPy, Pandas, TensorFlow, Keras, Seaborn, Matplotlib)

i. NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. A neat diagram of numpy is shown below in fig. 4.2.2.1.

Fig 4.2.2.1 Numpy

ii. Pandas

Pandas is a software library written for the Python programming language for data fifty-one manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at QAR Capital while he was a researcher there from 2007 to 2010. A neat diagram of machine learning is shown below in fig. 4.2.2.2.

Fig 4.2.2.2 Pandas

iii. TensorFlow

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on the training and inference of deep neural networks. A neat diagram of tensorflow is shown below in fig. 4.2.2.3.

Fig 4.2.2.3 Tensorflow

iv. Keras

Keras is an open-source software library that provides a python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. A neat diagram of keras is shown below in fig. 4.2.2.4.

Fig 4.2.2.4 Keras

v. Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. A neat diagram of seaborn is shown below in fig. 4.2.2.5.

Fig 4.2.2.5 Seaborn

vi. Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of matplotlib. A neat diagram of matplotlibis shown below in fig. 4.2.2.6.

Fig 4.2.2.6 Matplotlib

vii. SKlearn

Scikit-learn is a key library for the Python programming language that is typically used in machine learning projects. Scikit-learn is focused on machine learning tools including mathematical, statistical, and general-purpose algorithms that form the basis for many machine learning technologies. Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering, and dimensionality reduction via a consistence interface in Python. A neat diagram of scikit learn is shown below in fig. 4.2.2.7.

Fig 4.2.2.7 Scikit-Learn

CHAPTER 5

MACHINE LEARNING ALGORITHMS

Machine learning algorithms are mathematical model mapping methods used to learn or uncover underlying patterns embedded in the data. Machine learning comprises a group of computational algorithms that can perform pattern recognition, classification, and prediction on data by learning from existing data (training set).

5.1 Decision Tree

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. A neat diagram of decision tree is shown in fig 5.1.

5.1.1 Importance of Decision Tree

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). In Decision Trees, for predicting a class label for a record we start from the root of the tree.

5.1.2 How Decision Tree works?

The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

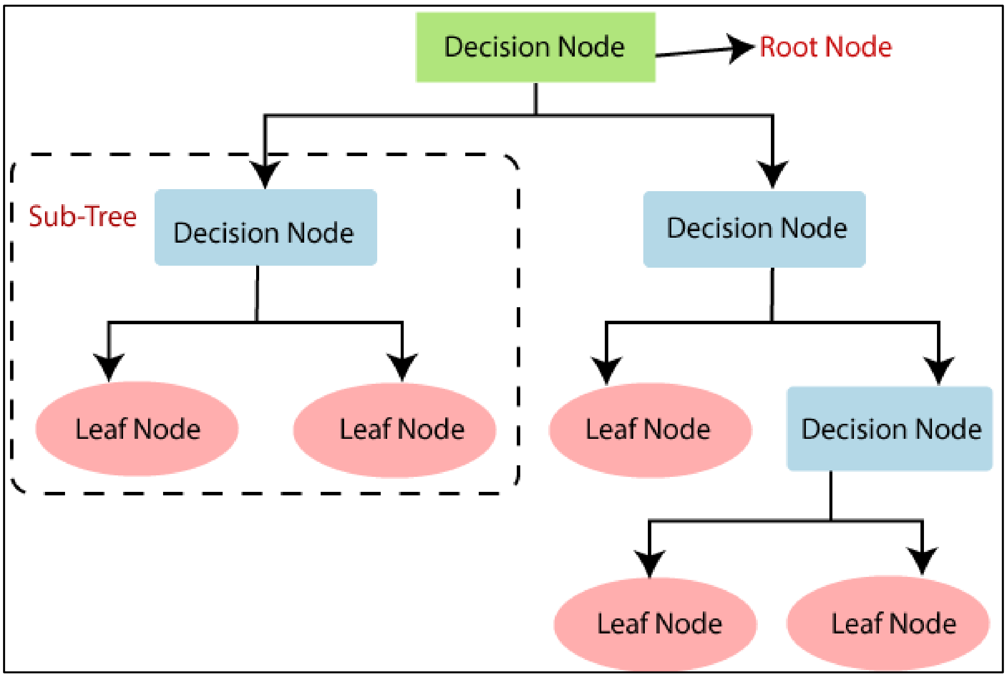

Fig. 5.1 Decision Tree

5.1.3 Advantage and Disadvantage of Decision Tree

Advantage of Decision Tree

a) It can be used for both classification and regression problems: Decision trees can be used to predict both continuous and discrete values i.e. they work well in both regression and classification tasks.

b) As decision trees are simple hence they require less effort for understanding an algorithm.

c) It can capture nonlinear relationships: They can be used to classify non-linearly separable data.

d) An advantage of the decision tree algorithm is that it does not require any transformation of the features if we are dealing with non-linear data because decision trees do not take multiple weighted combinations into account simultaneously.

e) They are very fast and efficient compared to KNN and other classification algorithms.

f) Easy to understand, interpret, visualize.

g) The data type of decision tree can handle any type of data whether it is numerical or categorical, or boolean.

h) Normalization is not required in the Decision Tree.

i) The decision tree is one of the machine learning algorithms where we don't worry about its feature scaling. Another one is random forests. Those algorithms are scale-invariant.

j) It gives us and a good idea about the relative importance of attributes.

k) Useful in data exploration: A decision tree is one of the fastest way to identify the most significant variables and relations between two or more variables. Decision trees have better power by which we can create new variables/features for the result variable.

l) Less data preparation needed: In the decision tree, there is no effect by the outsider or missing data in the node of the tree, that's why the decision tree requires fewer data.

m) Decision tree is non-parametric: Non-Parametric method is defined as the method in which there are no assumptions about the spatial distribution and the classifier structure.

Disadvantage of Decision Tree

a) Concerning the decision tree split for numerical variables millions of records: The time complexity right for operating this operation is very huge keep on increasing as the number of records gets increased decision tree with to numerical variables takes a lot of time for training.

b) Similarly, this happens in techniques like random forests, XGBoost.

c) Decision tree for many features: Take more time for training-time complexity to increase as the input increases.

d) Growing with the tree from the training set: Overfit pruning (pre, post), ensemble method random forest.

e) Method of overfitting: If we discuss overfitting, it is one of the most difficult methods for decision tree models. The overfitting problem can be solved by setting constraints on the parameters model and pruning method.

f) As you know, a decision tree generally needs overfitting of data. In the overfitting problem, there is a very high variance in output which leads to many errors in the final estimation and can show highly inaccuracy in the output. Achieve zero bias (overfitting), which leads to high variance.

g) Reusability in decision trees: In a decision tree there are small variations in the data that might output in a complex different tree is generated. This is known as variance in the decision tree, which can be decreased by some methods like bagging and boosting.

h) It can’t be used in big data: If the size of data is too big, then one single tree may grow a lot of nodes which might result in complexity and leads to overfitting.

i) There is no guarantee to return the 100% efficient decision tree.

5.2 Random Forest

Random forests are a supervised Machine learning algorithm that is widely used in regression and classification problems and produces, even without hyperparameter tuning a great result most of the time. It is perhaps the most used algorithm because of its simplicity. It builds a number of decision trees on different samples and then takes the majority vote if it's a classification problem. A neat diagram of random forest is shown in fig 5.2.

5.2.1 Random Forest Importance

Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a great tool for this that measures a feature's importance by looking at how much the tree nodes that use that feature reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results, so the sum of all importance is equal to one.

If you don't know how a decision tree works or what a leaf or node is, here is a good description from Wikipedia: "In a decision tree, each internal node represents a 'test' on an attribute (e.g., whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). A node that has no children is a leaf."

By looking at the feature importance you can decide which features to possibly drop because they don't contribute enough (or sometimes nothing at all) to the prediction process. This is important because a general rule in machine learning is that the more features you have the more likely your model will suffer from overfitting and vice versa.

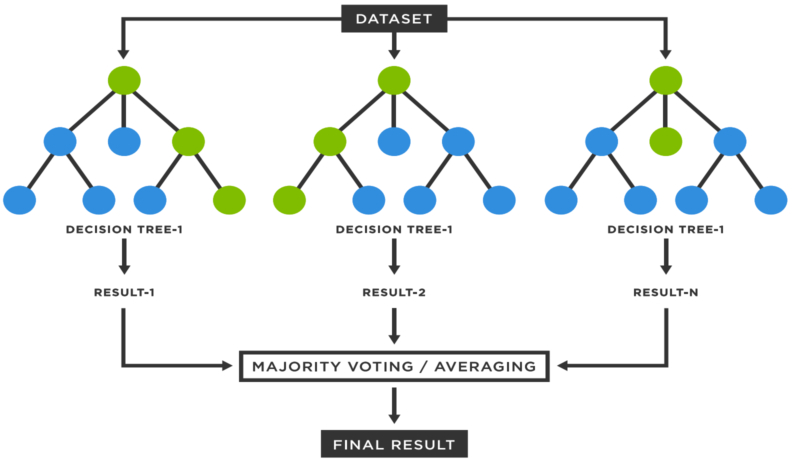

Fig 5.2 Random Forest

5.2.2 How Random Forest Works

Random forest is a supervised learning algorithm. The “forest” it builds is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Put simply: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

Let’s look at random forest in classification, since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like with two trees:

Random forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, there’s no need to combine a decision tree with a bagging classifier because you can easily use the classifier-class of random forest. With random forest, you can also deal with regression tasks by using the algorithm’s regressor.

Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

5.2.3 Advantages and Disadvantages Random Forest Algorithm

Advantages

a) One of the biggest advantages of random forest is its versatility. It can be used for both regression and classification tasks, and it’s also easy to view the relative importance it assigns to the input features.

b) Random forest is also a very handy algorithm because the default hyperparameters it uses often produce a good prediction result. Understanding the hyperparameters is pretty straightforward, and there’s also not that many of them.

c) One of the biggest problems in machine learning is overfitting, but most of the time this won’t happen thanks to the random forest classifier. If there are enough trees in the forest, the classifier won’t overfit the model.

Disadvantages

a) The main limitation of random forest is that a large number of trees can make the algorithm too slow and ineffective for real-time predictions. In general, these algorithms are fast to train, but quite slow to create predictions once they are trained. A more accurate prediction requires more trees, which results in a slower model.

b) In most real-world applications, the random forest algorithm is fast enough but there can certainly be situations where run-time performance is important and other approaches would be preferred. And, of course, random forest is a predictive modelling tool and not a descriptive tool, meaning if you're looking for a description of the relationships in your data, other approaches would be better.

5.3 Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Reading algorithms, used for Scheduling and retrieval problems. Mainly, however, it is used for Distribution Problems in Machine Learning.

The goal of the SVM algorithm is to create a better line or decision line that can divide n-dimensional space into classes so that we can easily place a data point in the appropriate category in the future. This best decision-making limit is called the hyperplane.

SVM selects the extra points / vectors that help create the hyperplane. These extreme cases are called supporting vectors, which is why the algorithm is called Vector Support Machine. A neat diagram of Support Vector Machine is shown in fig 5.3

5.5.1 Importance of Support Vector Machine

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs.

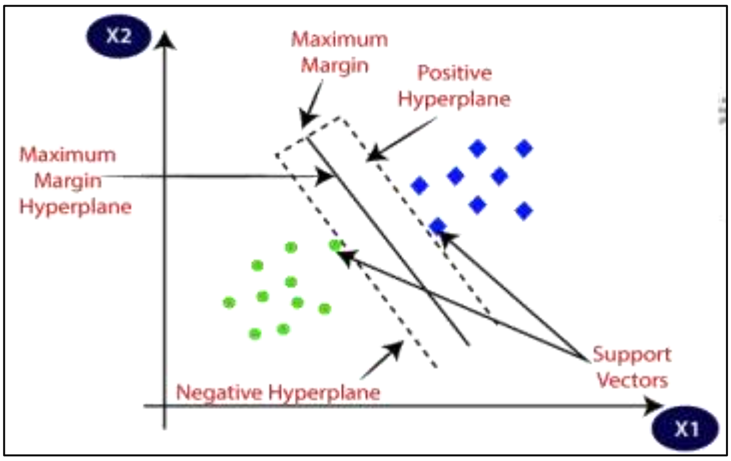

Fig 5.3 Support Vector Machine

5.3.2 How SVM works

SVM works by mapping the location of a high-resolution feature so that the data points are separated, even though the data can be categorized differently. A separator is found between sections, and the data is converted in such a way that the separator can be drawn as a hyperplane. After this, new data features can be used to predict which group the new record should belong to.

5.3.3 Advantage and Disadvantage of SVM

Advantages of Support Vector Machine (SVM)

a) Regularization capabilities: SVM has L2 Regularization feature. So, it has good generalization capabilities which prevent it from over-fitting.

b) Handles non-linear data efficiently: SVM can efficiently handle non-linear data using Kernel trick.

c) Solves both Classification and Regression problems: SVM can be used to solve both classification and regression problems. SVM is used for classification problems while SVR (Support Vector Regression) is used for regression problems.

d) Stability: A small change to the data does not greatly affect the hyperplane and hence the SVM. So, the SVM model is stable.

Disadvantages of Support Vector Machine (SVM)

a) Choosing an appropriate Kernel function is difficult: Choosing an appropriate Kernel function (to handle the non-linear data) is not an easy task. It could be tricky and complex.
b) In case of using a high dimension Kernel, you might generate too many support vectors which reduce the training speed drastically.
c) Extensive memory requirement: Algorithmic complexity and memory requirements of SVM are very high. You need a lot of memory since you have to store all the support vectors in the memory and this number grows abruptly with the training dataset size.
d) Requires Feature Scaling: One must do feature scaling of variables before applying SVM.
e) Long training time: SVM takes a long training time on large datasets.
f) Difficult to interpret: SVM model is difficult to understand and interpret by human beings unlike Decision Trees.

5.4 XGBoost

XGBoost is termed as Extreme Gradient Boosting Algorithm which is again an ensemble method that works by boosting trees. Boost makes use of a gradient descent algorithm which is the reason that it is called Gradient Boosting. The whole idea is to correct the previous mistake done by the model, learn from it and its next step improves the performance. The previous results are rectified, and performance is enhanced.

This gets continued until there is no scope of further improvements. Regularization is the feature that is dominant for this type of predictive algorithm. It is fast to execute and gives good accuracy. This algorithm is commonly used in Kaggle Competitions due to the ability to handle missing values and prevent overfitting. There are again a lot of hyperparameters that are used in this type of algorithm like a booster, learning rate, objective, etc.

The performance of XGBoost is no joke — it's become the go-to library for winning many Kaggle competitions. Its gradient boosting implementation is second to none and there's only more to come as the library continues to garner praise.

5.4.1 Importance of XGBoost

XGboost is a gradient boosting library. It provides parallel boosting trees algorithm that can solve Machine Learning tasks. It is available in many languages, like C++, Java, Python, R, Julia, Scala. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way.

5.4.2 How XGBoost Works?

XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leaf's that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

5.4.3 Advantage and Disadvantage of XGBoost

Advantage

a) Regularization: XGBoost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting. That is why, XGBoost is also called regularized form of GBM (Gradient Boosting Machine).

b) Parallel Processing: XGBoost utilizes the power of parallel processing and that is why it is much faster than GBM. It uses multiple CPU cores to execute the model.

c) Handling Missing Values: XGBoost has an in-built capability to handle missing values. When XGBoost encounters a missing value at a node, it tries both the left and right hand split and learns the way leading to higher loss for each node. It then does the same when working on the testing data.

d) Cross Validation: XGBoost allows user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run. This is unlike GBM where we have to run a grid-search and only a limited values can be tested.

e) Effective Tree Pruning: A GBM would stop splitting a node when it encounters a negative loss in the split. Thus, it is more of a greedy algorithm. XGBoost on the other hand make splits up to the max_depth specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain.

Disadvantage

a) XGBoost does not perform so well on sparse and unstructured data.

b) A common thing often forgotten is that Gradient Boosting is very sensitive to outliers since every classifier is forced to fix the errors in the predecessor learners.

c) The overall method is hardly scalable. This is because the estimators base their correctness on previous predictors, hence the procedure involves a lot of struggle to streamline.

CHAPTER 6

GATHERING DATA

Data collection is the process of gathering and measuring information from countless different sources. In order to use the data we collect to develop practical artificial intelligence (AI) and machine learning solutions, it must be collected and stored in a way that makes sense for the business problem at hand.

6.1 Data Collection

The set of phishing URLs are collected from opensource service called PhishTank. This service provide a set of phishing URLs in multiple formats like csv, json etc. that gets updated hourly. To download the data: https://www.phishtank.com/developer_info.php. From this dataset, 5000 random phishing URLs are collected to train the ML models.

The legitimate URLs are obtained from the open datasets of the University of New Brunswick, https://www.unb.ca/cic/datasets/url-2016.html. This dataset has a collection of benign, spam, phishing, malware & defacement URLs. Out of all these types, the benign url dataset is considered for this project. From this dataset, 5000 random legitimate URLs are collected to train the ML models.

6.1.1 Column Definitions

a) **phish_id:** The ID number by which Phishtank refers to a phish submission. All data in PhishTank is tied to this ID. This will always be a positive integer.
b) **phish_detail_url:** PhishTank detail url for the phish, where you can view data about the phish, including a screenshot and the community votes.

c) **url:** The phish URL. This is always a string, and in the XML feeds may be a CDATA block.

d) **submission_time:** The date and time at which this phish was reported to Phishtank. This is an ISO 8601 formatted date.

e) **verified:** Whether or not this phish has been verified by our community. In these data files, this will always be the string 'yes' since we only supply verified phishes in these files.

f) **verification_time:** The date and time at which the phish was verified as valid by our community. This is an ISO 8601 formatted date.

g) **online:** Whether or not the phish is online and operational. In these data files, this will always be the string 'yes' since we only supply online phishes in these files.

h) **target:** The name of the company or brand the phish is impersonating, if it's known.

6.2 Full Dataset Content

There are 5 zip files in total and range from ~2 gb to 3 gb in size. Additionally, we randomly sampled 5% of these datasets and created a smaller dataset for use in Kernels. The random sample contains 5000 sites.

CHAPTER 7

LIBRARIES USED AND THEIR INSTALLATION

7.1 NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, [16] various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

7.1.1 Description

NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project, and you can use it freely. NumPy stands for Numerical Python.

7.1.2 Function

NumPy fully supports an object-oriented approach, starting, once again, with ndarray. For example, the array is a class, possessing numerous methods and attributes. Many of its methods are mirrored by functions in the outermost NumPy namespace, allowing the programmer to code in whichever paradigm they prefer. This flexibility has allowed the NumPy array dialect and NumPy array class to become the de-facto language of seventy multi-dimensional data interchange used in Python.

7.1.3 Installation

a) If you use conda, you can install NumPy from the defaults or conda-forge channels:

```
conda create -n my-env
conda activate my-env
conda config –env –add channels conda-forge
conda install numpy
```

b) If you use pip, you can install NumPy with:

```
pip install numpy
```

7.1.4 Use in the Project:

At the core of the NumPy package, is the ndarray object.

a) Vectorization describes the absence of any explicit looping, indexing, etc., in the code- NumPy uses Vectorization hence is very fast.
b) NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python’s built-in sequences.
c) A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays.

In the project, the images are stored in the NumPy array.

7.2 Pandas

Pandas is mainly used for data analysis. Pandas allow importing data from various file formats such as comma-separated-values, JSON, SQL, and Microsoft Excel. Pandas allow various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.

7.2.1 Description

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three clause BSD license.

Pandas stand for “Python Data Analysis Library.” According to the Wikipedia page on Pandas, “the name is derived from the term “panel data,” an econometrics term for multidimensional structured data sets.”

7.2.2 Function

a) Fast and efficient Data Frame object with default and customized indexing.
b) Tools for loading data into in-memory data objects from different file formats.
c) Data alignment and integrated handling of missing data.
d) Reshaping and pivoting of date sets.
e) Label-based slicing, indexing and subsetting of large data sets.
f) Columns from a data structure can be deleted or inserted.
g) Group by data for aggregation and transformations.
h) High performance merging and joining of data.
i) Time Series functionality.

7.2.3 Installation

a) If you use conda, you can install Pandas from the defaults or conda-forge channels:

```
conda create -n my-env
conda activate my-env
conda config –env –add channels conda-forge
conda install pandas
```

b) If you use pip, you can install Pandas with:

```
pip install pandas
```

7.2.4 Use in the project

Data I/O:

We used a .csv file, to load the data of the file we have to use the Dataframe object of pandas. Data preview:

7.3 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt.

7.3.1 Description

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python

7.3.2 Function

a) Develop publication quality plots with just a few lines of code.
b) Use interactive figures that can zoom, pan, update etc.

c) Take full control of line styles, font properties, axes properties.

d) Export and embed to a number of file formats and interactive environments.

7.3.3 Installation

a) If you use conda, you can install Matplotlib from the defaults or conda-forge channels:

```
conda create -n my-env
conda activate my-env
conda config –env –add channels conda-forge
conda install matplotlib
```

b) If you use pip, you can install Matplotlib with:

```
pip install matplotlib
```

7.3.4 Use in the Project

In the project Matplotlib is used for drawing insights of the given features and the diseases. For e.g.: Effect of the different thorax diseases on Group of people (by age).

7.4 Util

Python Utils is a collection of small Python functions and classes which make common patterns shorter and easier.

7.4.1 Description

Utility Class, also known as Helper class, is a class, which contains just static methods, it is stateless and cannot be instantiated. It contains a bunch of related methods, so they can be reused across the application.

7.4.2 Function

This module makes it easy to execute common tasks in Python scripts such as converting text to numbers and making sure a string is in unicode or bytes format.

7.4.3 Installation

The package can be installed through pip (this is the recommended method):

```
pip install python-utils
```

7.4.4 Use in the Project

We have used utils to extract numbers from strings from the age field.

7.5 Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

7.5.1 Description

.

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas' data structures. Behind the scenes, seaborn uses matplotlib to draw its plots.

7.5.2 Function

Here is some of the functionality that seaborn offers:

a) A dataset-oriented API for examining relationships between multiple variables

b) Specialized support for using categorical variables to show observations or aggregate statistics.

c) Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data.

d) Automatic estimation and plotting of linear regression models for different kinds

e) dependent variables

f) Convenient views onto the overall structure of complex datasets

g) High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations

h) Concise control over matplotlib figure styling with several built-in themes

i) Tools for choosing color palettes that faithfully reveal patterns in your data

7.5.3 Installation

a) Official releases of seaborn can be installed from PyPI:

```
pip install seaborn
```

b) The library is also included as part of the Anaconda distribution:

```
conda install seaborn
```

7.5.4 Use in the Project

We have used the seaborn library to plot a boxplot. This boxplot checks for outliers in our dataset.

7.7 Keras

Keras is an open-source deep learning library written in Python. The project was started in 2015 by Francois Chollet. It quickly became a popular framework for developers, becoming one of, if not the most, popular deep learning libraries.

7.7.1 Description

Keras is popular because the API was clean and simple, allowing standard deep learning models to be defined, fit, and evaluated in just a few lines of code. A secondary reason Keras took-off was because it allowed you to use any one among the range of popular deep learning mathematical libraries as the backend (e.g., used to perform the computation), such as TensorFlow, Theano, and later, CNTK. This allowed the power of these libraries to be harnessed (e.g., GPUs) with a very clean and simple interface.

7.7.2 Function

The Keras functional API provides a more flexible way for defining models.

a) It specifically allows you to define multiple input or output models as well as models that share layers. More than that, it allows you to define ad hoc acyclic network graphs.

b) Models are defined by creating instances of layers and connecting them directly to each other in pairs, then defining a Model that specifies the layers to act as the input and output to the model.

7.7.3 Installation

Virtualenv is used to manage Python packages for different projects. This will be helpful to avoid breaking the packages installed in the other environments. So, it is always recommended to use a virtual environment while developing Python applications.

- **Linux/Mac OS**

Linux or mac OS users, go to your project root directory and type the below command to create virtual environment,

```
python3 -m venv kerasenv
```

After executing the above command, “kerasenv” directory is created with bin, lib and include folders in your installation location.

- **Windows**

```
py -m venv keras
```

Now, activate the environment..

- **Linux/Mac OS**

Now we have created a virtual environment named “kerasvenv”. Move to the folder and type the below command,

```
$ cd kerasvenv kerasvenv
```

- **Windows**

Windows users move inside the “kerasenv” folder and type the below command,

```
$ source bin/activate.\env\Scripts\activate
```

Keras depends on the following python libraries:

i. NumPy
ii. Pandas
iii. Scikit-learn
iv. Matplotlib
v. SciPy
vi. Seaborn

7.7.4 Use in the Project

To train our model we need Dense, Flatten and conv2d layers which all fall under the library keras. We have also used keras to save the deep learning model. We have also used keras to load the saved model which is later used during deployment of our model in tkinter.

7.7 Warnings in Python

Warning messages are typically issued in situations where it is useful to alert the user of some condition in a program, where that condition (normally) doesn’t warrant raising an exception and terminating the program. For example, one might want to issue a warning when a program uses an obsolete module.

Python programmers issue warnings by calling the warn() function defined in this module. (C programmers use PyErr_WarnEx(); see Exception Handling for details).

Warning messages are normally written to sys.stderr, but their disposition can be changed flexibly, from ignoring all warnings to turning them into exceptions. The disposition of warnings can vary based on the warning category, the text of the warning message, and the source location where it is issued. Repetitions of a particular warning for the same source location are typically suppressed.

There are two stages in warning control: first, each time a warning is issued, a determination is made whether a message should be issued or not; next, if a message is to be issued, it is formatted and printed using a user-settable hook.

The determination whether to issue a warning message is controlled by the warning filter, which is a sequence of matching rules and actions. Rules can be added to the filter by calling filterwarnings() and reset to its default state by calling resetwarnings().

The printing of warning messages is done by calling showwarning(), which may be overridden; the default implementation of this function formats the message by calling formatwarning(), which is also available for use by custom implementations

7.7.1 Warning Categories

In Python there are a variety of built-in exceptions which reflect categories of warning, some of them are:

a) **Warning Class:** It is the super class of all warning category classes and a subclass of the Exception class.
b) **UserWarning Class:** warn() function default category.\
c) **DeprecationWarning Class:** Base category for alerts regarding obsolete features when those warnings are for other developers (triggered by code in __main__ unless ignored).
d) **SyntaxWarning Class:** Base class for warnings of suspicious syntactic attributes.
e) **RuntimeWarning Class:** Base class for warnings of suspicious run time attributes.
f) **FutureWarning Class:** Base class for warnings on obsolete features when certain warnings are meant for end-users of Python-written programs.
g) **PendingDeprecationWarning Class:** Base class for warnings of an outdated attribute.
h) **ImportWarning Class:** Base class for warnings caused during a module importation process.
i) **UnicodeWarning Class:** Base class for Unicode based warnings.
j) **BytesWarning Class:** Base class for bytes and bytearray based warnings.
k) **ResourceWarning Class:** Base class for resource-related warnings.

7.8 Statsmodels

Statsmodel is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct. The package is released under the open source Modified BSD (3-clause) license. The online documentation is hosted at statsmodels.org.

The models module of scipy.stats was originally written by Jonathan Taylor. For some time it was part of scipy but was later removed. During the Google Summer of Code 2009, statsmodels was corrected, tested, improved and released as a new package. Since then, the statsmodels development team has continued to add new models, plotting tools, and statistical methods.

Most results have been verified with at least one other statistical package: R, Stata or SAS. The guiding principle for the initial rewrite and for continued development is that all numbers have to be verified. Some statistical methods are tested with Monte Carlo studies. While we strive to follow this test-driven approach, there is no guarantee that the code is bug-free and always works. Some auxiliary function are still insufficiently tested, some edge cases might not be correctly taken into account, and the possibility of numerical problems is inherent to many of the statistical models. We especially appreciate any help and reports for these kind of problems so we can keep improving the existing models.

The existing models are mostly settled in their user interface and we do not expect many large changes going forward. For the existing code, although there is no guarantee yet on API stability, we have long deprecation periods in all but very special cases, and we try to keep changes that require adjustments by existing users to a minimal level. For newer models we might adjust the user interface as we gain more experience and obtain feedback. These changes will always be noted in our release notes available in the documentation.

If you encounter a bug or an unexpected behaviour, please report it on the issue tracker. Use the show versions command to list the installed versions of statsmodels and its dependencies.

7.8.1 Features

Statsmodels is a popular library in Python that enables us to estimate and analyze various statistical models. It is built on numeric and scientific libraries like NumPy and SciPy. Some of the essential features of this package are-

a) It includes various models of linear regression like ordinary least squares, generalized least squares, weighted least squares, etc.
b) It provides some efficient functions for time series analysis.
c) It also has some datasets for examples and testing.
d) Models based on survival analysis are also available.
e) All the statistical tests that we can imagine for data on a large scale are present.

7.9 Scikit-Learn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Scikit-learn, first developed as a Google Summer of Code project in 2007, is the now widely considered to be the most popular Python library for machine learning.

There are a number of reasons why this library is seen as one of the best choices for machine learning projects, especially in production systems. These include, but aren't limited to the following:

a) It has a high level of support and strict governance for the development of the library which means that it is an incredibly robust tool.
b) There is a clear, consistent code style which ensures that your machine learning code is easy to understand and reproducible, and also vastly lowers the barrier to entry for coding machine learning models.
c) It is widely supported by third-party tools so it is possible to enrich the functionality to suit a range of use cases.

If you are learning machine learning then Scikit-learn is probably the best library to start with. Its simplicity means that it is fairly easy to pick up and by learning how to use it you will also gain a good grasp of the key steps in a typical machine learning workflow.

7.9.1 Features

Rather than focusing on loading, manipulating and summarising data, Scikit-learn library is focused on modeling the data. Some of the most popular groups of models provided by Sklearn are as follows:

a) **Supervised Learning algorithms:** Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.
b) **Unsupervised Learning algorithms:** On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.
c) **Clustering:** This model is used for grouping unlabeled data.
d) **Cross Validation:** It is used to check the accuracy of supervised models on unseen data.

e) **Dimensionality Reduction:** It is used for reducing the number of attributes in data which can be further used for summarisation, visualisation and feature selection.

f) **Ensemble methods:** As name suggest, it is used for combining the predictions of multiple supervised models.

g) **Feature extraction:** It is used to extract the features from data to define the attributes in image and text data.

h) **Feature selection:** It is used to identify useful attributes to create supervised models.

i) **Open Source:** It is open source library and also commercially usable under BSD license.

CHAPTER 8

FEATURE EXTRACTION

One of the challenges faced by our research was the unavailability of reliable training datasets. In fact, this challenge faces any researcher in the field. However, although plenty of articles about predicting phishing websites using data mining techniques have been disseminated these days, no reliable training dataset has been published publicly, maybe because there is no agreement in literature on the definitive features that characterize phishing websites, hence it is difficult to shape a dataset that covers all possible features.

8.1 Address Bar based Features

In a web browser, the address bar (also location bar or URL bar) is a GUI widget that shows the current URL. The user can type a URL into the bar to navigate to a chosen website in most modern browsers, non-URLs are automatically sent to a search engine. The address bar is the familiar text field at the top of a web browser's graphical user interface (GUI) that displays the name or the URL (uniform resource locator) of the current web page. Users request websites and pages by typing either the name or the URL into the address bar.

8.1.1 Using the IP Address

If an IP address is used as an alternative of the domain name in the URL, such as "http://125.98.3.123/fake.html", users can be sure that someone is trying to steal their personal information. Sometimes, the IP address is even transformed into hexadecimal code as shown in the following link "http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html".

If the domain part of website has an ip address. So, the website is under phishing website. Otherwise, the website is safe and called as legitimate website.

8.1.2 Long URL to Hide the Suspicious Part

Phishers can use long URL to hide the doubtful part in the address bar. For example: http://federmacedoadv.com.br/3f/aze/ab51e2e319e51502f416dbe46b773a5e/?cmd=_hom&amp;dispatch=11004d58f5b74f8dc1e7c2e8dd4105e811004d58f5b74f8dc1e7c2e8dd4105e8@phishing.website.html. To ensure accuracy of our study, we calculated the length of URLs in the dataset and produced an average URL length. The results showed that if the length of the URL is greater than or equal 54 characters then the URL classified as phishing. By reviewing our dataset, we were able to find 1220 URLs lengths equals to 54 or more which constitute 48.8% of the total dataset size.

If the URL length is less than 54 then the website is safe and called as legitimate website or URL length is between 54 and 75 then the website is suspicious. Otherwise, the website is phishing website.

We have been able to update this feature rule by using a method based on frequency and thus improving upon its accuracy

8.1.3 Using URL Shortening Services

URL shortening is a method on the “World Wide Web” in which a URL may be made considerably smaller in length and still lead to the required webpage. This is accomplished by means of an “HTTP Redirect” on a domain name that is short, which links to the webpage that has a long URL. For example, the URL “http://portal.hud.ac.uk/” can be shortened to “bit.ly/19DXSk4”.

If the URL is very short or tiny then the website is phishing website. Otherwise, the website is legitimate website.

8.1.4 URL's having "@" Symbol

Some websites having special character which causes of phishing. Using "@" symbol in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol.

If the website has @ symbol then website may be phishing website. Otherwise, the website is safe and called as legitimate website.

8.1.5 Redirecting using "//"

The existence of "//" within the URL path means that the user will be redirected to another website. An example of such URL's is: "http://www.legitimate.com//http://www.phishing.com". We examine the location where the "//" appears. We find that if the URL starts with "HTTP", that means the "//" should appear in the sixth position. However, if the URL employs "HTTPS" then the "//" should appear in seventh position.

The Position of the Last Occurrence of "//" in the URL is greater than 7 then the website is phishing website. Otherwise, the website is legitimate website.

8.1.6 Adding Prefix or Suffix Separated by (-) to the Domain

Another special symbol is hyphen (-) or dash symbol which sometimes treat as phishing website symbol. The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example: http://www.Confirme-paypal.com/.

If the domain name part includes – symbol then the website is phishing website. Otherwise, the website is legitimate website.

8.1.7 Sub Domain and Multi Sub Domains

Let us assume we have the following link: http://www.hud.ac.uk/students/. A domain name might include the country-code top-level domains (ccTLD), which in our example is “uk”. The “ac” part is shorthand for “academic”, the combined “ac.uk” is called a second-level domain (SLD) and “hud” is the actual name of the domain. To produce a rule for extracting this feature, we firstly have to omit the (www.) from the URL which is in fact a sub domain in itself. Then, we have to remove the (ccTLD) if it exists. Finally, we count the remaining dots. If the number of dots is greater than one, then the URL is classified as “Suspicious” since it has one sub domain. However, if the dots are greater than two, it is classified as “Phishing” since it will have multiple sub domains. Otherwise, if the URL has no sub domains, we will assign “Legitimate” to the feature.

If number of dots present on the website is one then the website is legitimate or if number of dots present in website are two then the website is suspicious. Otherwise, the website is under the category of phishing.

8.1.8 HTTPS (Hyper Text Transfer Protocol with Secure Sockets Layer)

The existence of HTTPS is very important in giving the impression of website legitimacy, but this is clearly not enough. The authors in (Mohammad, Thabtah and McCluskey 2012) (Mohammad, Thabtah and McCluskey 2013) suggest checking the certificate assigned with HTTPS including the extent of the trust certificate issuer, and the certificate age. Certificate Authorities that are consistently listed among the top trustworthy names include: “GeoTrust, GoDaddy, Network Solutions, Thawte, Comodo, Doster and

VeriSign”. Furthermore, by testing out our datasets, we find that the minimum age of a reputable certificate is two years.

If the website age of certificate is less than one year then the website may be the phishing website. If the website age of certificate is greater than or equal to one year then the website is come under legitimate.

8.1.9 Domain Registration Length

Based on the fact that a phishing website lives for a short period of time, we believe that trustworthy domains are regularly paid for several years in advance. In our dataset, we find that the longest fraudulent domains have been used for one year only.

If domain of website expires in less than or equal to one year then the website is phishing website. Otherwise, the website is legitimate website.

8.1.10 Favicon

A favicon is a graphic image (icon) associated with a particular Web page and/or Web site. Many recent user agents (such as graphical browsers and newsreaders) display them as a visual reminder of the Web site identity in the address bar or in tabs. If the favicon is loaded from a domain other than that shown in the address bar, then the webpage is likely to be considered a Phishing attempt.

If favicon loaded from external domain then the website is phishing website. Otherwise, the website is legitimate website.

8.1.11 Using Non-Standard Port

This feature is useful in validating if a particular service (e.g., HTTP) is up or down on a specific server. In the aim of controlling intrusions, it is much better to merely open ports that you need. Several firewalls, Proxy and Network Address Translation (NAT) servers will, by default, block all or most of the ports and only open the ones selected. If all ports are open, phishers can run almost any service they want and as a result, user information is threatened.

If Port # is of the preferred Status then the website is phishing website. Otherwise, the website is legitimate website.

8.1.12 The Existence of “HTTPS” Token in the Domain Part of the URL

The phishers may add the “HTTPS” token to the domain part of a URL in order to trick users. For example: http://https-www-paypal-it-webapps-mpp-home.soft-hair.com/.

If using http token in domain part of the url then the website is phishing website. Otherwise, the website is legitimate website.

8.2 Abnormal Based Features

8.2.1. Request URL:

Request URL examines whether the external objects contained within a webpage such as images, videos and sounds are loaded from another domain. In legitimate webpages, the webpage address and most of objects embedded within the webpage are sharing the same domain.

Rule: if (i) % of Request URL < 22% → Legitimate

(ii) %of Request URL ≥ 22% and 61% → Suspicious

(iii) Otherwise → feature = Phishing

8.2.2. URL of Anchor

An anchor is an element defined by the tag. This feature is treated exactly as "Request URL". However, for this feature we examine:

a) If the tags and the website have different domain names. This is similar to request URL feature.

b) If the anchor does not link to any webpage.

i. <a href = "#">

ii. <a href = "#content">

iii. <a href = "#skip">

iv. <a href = "Javascript::void(0)">

8.2.3. Links in <Meta>, <Script> and <Link> tags

Given that our investigation covers all angles likely to be used in the webpage source code, we find that it is common for legitimate websites to use <Meta> tags to offer metadata about the HTML document; <Script> tags to create a client side script: and <Link> tags to retrieve other web resources. It is expected that these tags are linked to the same domain of the webpage.

8.2.4. Server Form Handler (SFH)

SFHs that contain an empty string or "about:blank" are considered doubtful because an action should be taken upon the submitted information. In addition, if the domain name in SFHs is different from the domain name of the webpage, this reveals that the webpage is suspicious because the submitted information is rarely handled by external domains.

8.2.5. Submitting Information to Email

Web form allows a user to submit his personal information that is directed to a server for processing. A phisher might redirect the user's information to his personal email. To that end, a server-side script language might be used such as "mail()" function in PHP. One more client-side function that might be used for this purpose is the "mailto:" function.

8.2.6. Abnormal URL

This feature can be extracted from WHOIS database. For a legitimate website, identity is typically part of its URL.

8.3. HTML and JavaScript Based Features

8.3.1. Website Forwarding

The fine line that distinguishes phishing websites from legitimate ones is how many times a website has been redirected. In our dataset, we find that legitimate websites have been redirected one time max. On the other hand, phishing websites containing this feature have been redirected at least 4 times.

8.3.2. Status Bar Customization

Phishers may use JavaScript to show a fake URL in the status bar to users. To extract this feature, we must dig-out the webpage source code, particularly the "onMouseOver" event, and check if it makes any changes on the status bar.

8.3.3. Disabling Right Click

Phishers use JavaScript to disable the right-click function, so that users cannot view and save the webpage source code. This feature is treated exactly as "Using onMouseOver to hide the Link”. Nonetheless, for this feature, we will search for event “event.button==2” in the webpage source code and check if the right click is disabled.

8.3.4. Using Pop-up Window

It is unusual to find a legitimate website asking users to submit their personal information through a pop-up window. On the other hand, this feature has been used in some legitimate websites and its main goal is to warn users about fraudulent activities or broadcast a welcome announcement, though no personal information was asked to be filled in through these pop-up windows.

8.3.5. IFrame Redirection

IFrame is an HTML tag used to display an additional webpage into one that is currently shown. Phishers can make use of the “iframe” tag and make it invisible i.e. without frame borders. In this regard, phishers make use of the “frameBorder” attribute which causes the browser to render a visual delineation.

8.4. Domain based Features

8.4.1. Age of Domain

This feature can be extracted from WHOIS database (Whois 2005). Most phishing websites live for a short period of time. By reviewing our dataset, we find that the minimum age of the legitimate domain is 6 months.

8.4.2. DNS Record

For phishing websites, either the claimed identity is not recognized by the WHOIS database (Whois 2005) or no records founded for the hostname (Pan and Ding 2006). If the DNS record is empty or not found then the website is classified as “Phishing”, otherwise it is classified as “Legitimate”.

8.4.3. Website Traffic

This feature measures the popularity of the website by determining the number of visitors and the number of pages they visit. However, since phishing websites live for a short period of time, they may not be recognized by the Alexa database (Alexa the Web Information Company., 1996). By reviewing our dataset, we find that in worst scenarios, legitimate websites ranked among the top 100,000. Furthermore, if the domain has no traffic or is not recognized by the Alexa database, it is classified as “Phishing”. Otherwise, it is classified as “Suspicious”.

8.4.4. PageRank

PageRank is a value ranging from “0” to “1”. PageRank aims to measure how important a webpage is on the Internet. The greater the PageRank value the more important the webpage. In our datasets, we find that about 95% of phishing webpages have no PageRank. Moreover, we find that the remaining 5% of phishing webpages may reach a PageRank value up to “0.2”

CHAPTER 9

DATASET PREPARATION

When we talk about data, we usually think of some large datasets with huge numbers of rows and columns. While that is a likely scenario, it is not always the case data could be in so many different forms: Structured Tables, Images, Audio files, Videos etc.

Machines don’t understand free text, image or video data as it is, they understand 1s and 0s. So it probably won’t be good enough if we put on a slideshow of all our images and expect our machine learning model to get trained just by that. In any Machine Learning process, Data Preprocessing is that step in which the data gets transformed, or Encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm. A dataset can be viewed as a collection of data objects, which are often also called records, points, vectors, patterns, events, cases, samples, observations, or entities. Data objects are described by a number of features that capture the basic characteristics of an object, such as the mass of a physical object or the time at which an event occurred, etc. Features are often called variables, characteristics, fields, attributes, or dimensions.

7.1 Loading Libraries

We'll make use of the following packages:

a) numpy and pandas is what we'll use to manipulate our data
b) matplotlib.pyplot and seaborn will be used to produce plots for visualization 85
c) util will provide the locally defined utility functions that have been provided for this assignment.

A. Exploratory Data Analysis

Exploratory data analysis (EDA) is performed in order to gain a preliminary understanding and allow us to get acquainted with the dataset. In a typical data science project, one of the first things that I would do is "eyeballing the data" by performing EDA so as to gain a better understanding of the data.

Three major EDA approaches that I normally use includes:

a) **Descriptive statistics** — Mean, median, mode, standard deviation
b) **Data visualisations** — Heat maps (discerning feature intra-correlation), box plot (visualize group differences), scatter plots (visualize correlations between features), principal component analysis (visualize distribution of clusters presented in the dataset), etc.
c) **Data shaping/ cleaning** — Pivoting data, grouping data, filtering data, etc.

7.2 Descriptive Statistics

In Descriptive Statistics you are describing, presenting, summarizing and organizing your data (population), either through numerical calculations or graphs or tables. There are four major types of descriptive statistics:

a) Measures of Frequency: * Count, Percent, Frequency.
b) Measures of Central Tendency: * Mean, Median, and Mode.
c) Measures of Dispersion or Variation: * Range, Variance, Standard Deviation.
d) Measures of Position: * Percentile Ranks, Quartile Ranks.

7.3 Dataset Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. In the world of Big Data, data visualization tools and technologies are essential to analyse massive amounts of information and make data-driven decisions. Our eyes are drawn to colours and patterns. We can quickly identify red from blue, square from circle. Our culture is visual, including everything from art and advertisements to TV and movies. Data visualization is another form of visual art that grabs our interest and keeps our eyes on the message. When we see a chart, we quickly see trends and outliers. If we can see something, we internalize it quickly. It's storytelling with a purpose. If you've ever stared at a massive spreadsheet of data and couldn't see a trend, you know how much more effective a visualization can be. As the "age 87 of Big Data" kicks into high-gear, visualization is an increasingly key tool to make sense of the trillions of rows of data generated every day. Data visualization helps to tell stories by curating data into a form easier to understand, highlighting the trends and outliers. A good visualization tells a story, removing the noise from data and highlighting the useful information.

However, it's not simply as easy as just dressing up a graph to make it look better or slapping on the "info" part of an infographic. Effective data visualization is a delicate balancing act between form and function. The plainest graph could be too boring to catch any notice or it makes a powerful point; the most stunning visualization could utterly fail at conveying the right message or it could speak volumes. The data and the visuals need to work together, and there's an art to combining great analysis with great storytelling. It's hard to think of a professional industry that doesn't benefit from making data more understandable. Every STEM field benefits from understanding data—and so do fields in government, finance, marketing, history, consumer goods, service industries, education, sports, and so on.

While we'll always wax poetically about data visualization (you're on the Tableau website, after all) there are practical, real-life applications that are undeniable. And, since visualization is so prolific, it's also one of the most useful professional skills to develop. The better you can convey your points visually, whether in a dashboard or a slide deck, the better you can leverage that information. The concept of the citizen data scientist is on the rise. Skill sets are changing to accommodate a data-driven world. It is increasingly valuable for professionals to be able to use data to make decisions and use visuals to tell stories of when data informs the who, what, when, where, and how. While traditional education typically draws a distinct line between creative storytelling and technical analysis, the modern professional world also values those who can cross between the two: data visualization sits right in the middle of analysis and visual storytelling. Data visualization is the representation of data or information in a graph, chart, or other visual format. It communicates relationships of the data with images. This is important because it allows trends and patterns to be more easily seen. With the rise of big data upon us, we need to be able to interpret increasingly larger batches of data. Machine learning makes it easier to conduct analyses such as predictive analysis, which can then serve as helpful visualizations to present. But data visualization is not only important for data scientists and data analysts, it is necessary to understand data visualization in any career. Whether you work in finance, 88 marketing, tech, design, or anything else, you need to visualize data.

7.4 Dataset Cleaning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

When combining multiple data sources, there are many opportunities for data to be duplicated or mislabelled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset.

7.4.1 How do you clean data?

While the techniques used for data cleaning may vary according to the types of data your company stores, you can follow these basic steps to map out a framework for your organization.

Step 1: Remove duplicate or irrelevant observations

Remove unwanted observations from your dataset, including duplicate observations or irrelevant observations. Duplicate observations will happen most often during data collection. When you combine data sets from multiple places, scrape data, or receive data from clients or multiple departments, there are opportunities to create duplicate data. Deduplication is one of the largest areas to be considered in this process.

Irrelevant observations are when you notice observations that do not fit into the specific problem you are trying to analyze. For example, if you want to analyze data regarding millennial customers, but your dataset includes older generations, you might remove those irrelevant observations. This can make analysis more efficient and minimize distraction from your primary target—as well as creating a more manageable and more performant dataset.

Step 2: Fix structural errors

Structural errors are when you measure or transfer data and notice strange naming conventions, typos, or incorrect capitalization. These inconsistencies can cause mislabelled categories or classes. For example, you may find “N/A” and “Not Applicable” both appear, but they should be analysed as the same category.

Step 3: Filter unwanted outliers

Often, there will be one-off observations where, at a glance, they do not appear to fit within the data you are analysing. If you have a legitimate reason to remove an outlier, like improper data-entry, doing so will help the performance of the data you are working with. However, sometimes it is the appearance of an outlier that will prove a theory you are working on.

Remember: just because an outlier exists, doesn't mean it is incorrect. This step is needed to determine the validity of that number. If an outlier proves to be irrelevant for analysis or is a mistake, consider removing it.

Step 4: Handle missing data

You can't ignore missing data because many algorithms will not accept missing values. There are a couple of ways to deal with missing data. Neither is optimal, but both can be considered.

a) As a first option, you can drop observations that have missing values, but doing this will drop or lose information, so be mindful of this before you remove it.
b) As a second option, you can input missing values based on other observations; again, there is an opportunity to lose integrity of the data because you may be operating from assumptions and not actual observations.
c) As a third option, you might alter the way the data is used to effectively navigate null values.

Step 5: Inconsistent values

We know that data can contain inconsistent values. Most probably we have already faced this issue at some point. For instance, the 'Address' field contains the 'Phone number'. It may be due to human error or maybe the information was misread while being scanned from a handwritten form. It is therefore always advised to perform data assessment like knowing what the data type of the features should be and whether it is the same for all the data objects. We had age as a string type in the form “66Y” where “Y” needs to be eliminated and converted to integer type for processing.

Step 6: Duplicate values

A dataset may include data objects which are duplicates of one another. It may happen when the same person submits a form more than once. The term deduplication is often used to refer to the process of dealing with duplicates. In most cases, the duplicates are removed so as to not give that particular data object an advantage or bias, when running machine learning algorithms.

Step 7: Validate and QA

At the end of the data cleaning process, you should be able to answer these questions as a part of basic validation:

a) Does the data make sense?
b) Does the data follow the appropriate rules for its field?
c) Does it prove or disprove your working theory, or bring any insight to light?
d) Can you find trends in the data to help you form your next theory?
e) If not, is that because of a data quality issue?

False conclusions because of incorrect or “dirty” data can inform poor business strategy and decision-making. False conclusions can lead to an embarrassing moment in a reporting meeting when you realize your data doesn’t stand up to scrutiny.

Before you get there, it is important to create a culture of quality data in your organization. To do this, you should document the tools you might use to create this culture and what data quality means to you.

7.4.2 Components of quality data

Determining the quality of data requires an examination of its characteristics, then weighing those characteristics according to what is most important to your organization and the application(s) for which they will be used.

A. Characteristics of quality data

a) **Validity**: The degree to which your data conforms to defined business rules or constraints.
b) **Accuracy:** Ensure your data is close to the true values.
c) **Completeness:** The degree to which all required data is known.
d) **Consistency:** Ensure your data is consistent within the same dataset and/or across multiple data sets.
e) **Uniformity:** The degree to which the data is specified using the same unit of measure.

Having clean data will ultimately increase overall productivity and allow for the highest quality information in your decision-making. Benefits include:

a) Removal of errors when multiple sources of data are at play.
b) Fewer errors make for happier clients and less-frustrated employees.
c) Ability to map the different functions and what your data is intended to do.
d) Monitoring errors and better reporting to see where errors are coming from, making it easier to fix incorrect or corrupt data for future applications.
e) Using tools for data cleaning will make for more efficient business practices and quicker decision-making.

7.5 Feature Sampling

Sampling is a very common method for selecting a subset of the dataset that we are analyzing. In most cases, working with the complete dataset can turn out to be too expensive considering the memory and time constraints. Using a sampling algorithm can help us reduce the size of the dataset to a point where we can use a better, but more expensive, machine learning algorithm. The key principle here is that the sampling should be done in such a manner that the sample generated should have approximately the same properties as the original dataset, meaning that the sample is representative. This involves choosing the correct sample size and sampling strategy. Simple Random Sampling dictates that there is an equal probability of selecting any particular entity. It has two main variations as well:

a) Sampling without Replacement: As each item is selected, it is removed from the set of all the objects that form the total dataset.
b) Sampling with Replacement: Items are not removed from the total dataset after 93 getting selected. This means they can get selected more than once. Although simple random sampling provides two great sampling techniques, it can fail to output a representative sample when the dataset includes object types which vary drastically in ratio. This can cause problems when the sample needs to have a proper representation of all object types, for example, when we have an imbalanced dataset. An imbalanced dataset is one where the number of instances of a classes are significantly higher than another classes, thus leading to an imbalance and creating rarer classes.

7.6 Dimensionality Reduction

Most real world datasets have a large number of features. For example, consider an image processing problem, we might have to deal with thousands of features, also called dimensions. As the name suggests, dimensionality reduction aims to reduce the number of features - but not simply by selecting a sample of features from the feature-set, which is something else — Feature Subset Selection or simply Feature Selection. Conceptually, dimension refers to the number of geometric planes the dataset lies in, which could be so high that it cannot be visualized with pen and paper. More the number of such planes, more is the complexity of the dataset.

7.6.1 The Curse of Dimensionality

This refers to the phenomena that generally data analysis tasks become significantly harder as the dimensionality of the data increases. As the dimensionality increases, the number planes occupied by the data increases thus adding more and more sparsity to the data which is difficult to model and visualize. A representation of how principal components can be visualized. Representation of components in different spaces. What dimension reduction essentially does is that it maps the dataset to a lower-dimensional space, which may very well be to a number of planes which can now be visualized, say 2D. The basic objective of techniques which are used for this purpose is to reduce the dimensionality of a dataset by creating new features which are a combination of the old features. In other words, the higher-dimensional feature-space is mapped to a lower-dimensional feature-space. Principal Component Analysis and Singular Value Decomposition are two widely accepted techniques. Data Analysis algorithms work better if the dimensionality of the dataset is lower. This is mainly because irrelevant features and noise have now been eliminated. The models which are built on top of lower dimensional data are more understandable and explainable.

7.6.2 Feature Encoding

Machine learning models can only work with numerical values. For this reason, it is necessary to transform the categorical values of the relevant features into numerical ones. This process is called feature encoding. As mentioned before, the whole purpose of data preprocessing is to encode the data in order to bring it to such a state that the machine now understands it. Feature encoding is basically performing transformations on the data such that it can be easily accepted as input for machine learning algorithms while still retaining its original meaning. Data frame analytics automatically performs feature encoding. The input data is pre-processed with the following encoding techniques:

a) one-hot encoding: Assigns vectors to each category. The vector represents whether the corresponding feature is present (1) or not (0).

b) target-mean encoding: Replaces categorical values with the mean value of the target variable.

c) frequency encoding: Takes into account how many times a given categorical value is present in relation with a feature.

When the model makes predictions on new data, the data needs to be processed in the same way it was trained. Machine learning model inference in the Elastic Stack does this automatically, so the automatically applied encodings are used in each call for inference. Feature importance is calculated for the original categorical fields, not the automatically encoded features.

RESULT

Scikit-learn tool has been used to import Machine learning algorithms. Dataset is divided into training set and testing set in 80:20 ratios respectively. Each classifier is trained using training set and testing set is used to evaluate performance of classifiers. Performance of classifiers has been evaluated by calculating classifier's accuracy score, false negative rate and false positive rate.

We get following detection accuracies using different algorithms and we also found algorithms perform better on high training set:

a) XGBoost algorithm gives detection accuracy of 85.91% which is higher than other algorithms.

b) Decision tree algorithm gives detection accuracy of 81.7%.

c) Support Vector Machine algorithm gives detection accuracy of 81.53%.

d) Random Forest algorithm gives detection accuracy of 83.2%.

FUTURE WORK

The project here has some limitations and it can be extended further. The first limitation is that we considered a small data set that contains 5000 URLs, and there are 17 features for each URL. The second limitation is that all features are discrete. Often, classifiers such as decision trees, Random Forest, and rule-based systems are more suitable when features are discrete. Furthermore, we used features that were already extracted from URLs.

The present work can be extended as be we can evaluate classifiers using a large data set that contains a few thousands of URLs and extract more number of features that may be significant in decision making. Larger data sets are available in public domain.

We can generate associative rules using the frequent item data sets with the minimum support and confidence values and build a rule-based system using associative rules to classify URLs. The rule-based classifier then can be compared with other classification methods. another approach for generating classification rules from data samples is to divide the feature space using fuzzy membership functions and extract and optimize classification rules. The extracted rules can be used to build a fuzzy inference system that can classify URLs.

In order to avoid the problem of overfitting a classifier, we need to include a pre-process stage. In processing, we can use clustering to find out outliers or noisy data samples. Such samples should not be used in the training set data.

CONCLUSION

Thus, to summarize, we have seen how phishing is a huge threat to the security and safety of the web and how phishing detection is an important problem domain. We have reviewed some of the traditional approaches to phishing detection; namely blacklist and heuristic evaluation methods, and their drawbacks. We have tested four machine learning algorithms on the Phishing Websites Dataset and reviewed their results. We then selected the best algorithm based on its performance and built a Chrome extension for detecting phishing web pages.

The extension allows easy deployment of our phishing detection model to end users. We have detected phishing websites using XGBoost algorithm with accuracy of 85.9%. For future enhancements, we intend to build the phishing detection system as a scalable web service which will incorporate online learning so that new phishing attack patterns can easily be learned.

REFERENCES

[1] Gunter Ollmann, “The Phishing Guide Understanding & Preventing Phishing Attacks”, IBM Internet Security Systems, 2007.

[2] Routhu Srinivasa Rao1, Alwyn Roshan Pais: Detection of phishing websites using an efficient feature-based machine learning framework In Springer 2018.

[3] Chunlin Liu, Bo Lang: Finding effective classifier for malicious URL detection: In ACM, 2018

[4] Sudhanshu Gautam, Kritika Rani and Bansidhar Joshi: Detecting Phishing Websites Using Rule-Based Classification Algorithm: A Comparison: In Springer, 2018.

[5] M. Amaad Ul Haq Tahir, Sohail Asghar, Ayesha Zafar, Saira Gillani: A Hybrid Model to Detect Phishing-Sites using Supervised Learning Algorithms: In International Conference on Computational Science and Computational Intelligence IEEE, 2016.

[6] Hossein Shirazi, Kyle Haefner, Indrakshi Ray: Fresh-Phish: A Framework for Auto-Detection of Phishing Websites: In (International Conference on Information Reuse and Integration (IRI)) IEEE, 2017

[7] Ankit Kumar Jain, B. B. Gupta: Towards detection of phishing websites on client-side using machine learning based approach: In Springer Science+Business Media, LLC, part of Springer Nature, 2017

[8] Bhagyashree E. Sananse, Tanuja K. Sarode: Phishing URL Detection: A Machine Learning and Web Mining-based Approach: In International Journal of Computer Applications, 2015

[9] Mustafa AYDIN, Nazife BAYKAL: Feature Extraction and Classification Phishing Websites Based on URL: IEEE, 2015

[10] Priyanka Singh, Yogendra P.S. Maravi, Sanjeev Sharma: Phishing Websites Detection through Supervised Learning Networks: In IEEE, 2015

[11] Pradeepthi. K V and Kannan. A: Performance Study of Classification Techniques for Phishing URL Detection: In 2014 Sixth International Conference on Advanced Computing (ICoAC) IEEE, 2014.

[12] Luong Anh Tuan Nguyen, Ba Lam To, Huu Khuong Nguyen† and Minh Hoang Nguyen: Detecting Phishing Web sites: A Heuristic URL-Based Approach: In The 2013 International Conference on Advanced Technologies for Communications (ATC'13).

[13] Ahmad Abunadi, Anazida Zainal, Oluwatobi Akanb: Feature Extraction Process: A Phishing Detection Approach: In IEEE, 2013.

[14] Rami M. Mohammad, Fadi Thabtah, Lee McCluskey: An Assessment of Features Related to Phishing Websites using an Automated Technique: In The 7th International Conference for Internet Technology and Secured Transactions, IEEE, 2012.

[15] https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML#:~:text=Machine%20learning%20(ML)%20is%20a,to%20predict%20new%20output%20values.

[16] https://www.zenesys.com/blog/top-10-python-libraries-for-machine-learning