

VISVESVARAYA TECHNOLOGY UNIVERSITY
JANA SANGAMA, Belagavi – 590018



Internship report on
“PYTHON WITH MACHINE LEARNING”

Submitted in partial fulfilment of the requirements of the 3rd Semester in
MASTER OF COMPUTER APPLICATIONS

Submitted By
ADNAN AHAD
1AT21MC001
Under the guidance of

Dr. Dattatreya P Mankame
Head of Department,
Department of MCA



ATRIA INSTITUTE OF TECHNOLOGY

Ananda Nagar, Bangalore-560024

DEPARTMENT OF MCA

2021-2023

ATRIA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

Ananda Nagar, Bengaluru – 560024



INTERNSHIP CERTIFICATE

This is to certify that **ADNAN AHAD** bearing **1AT21MC001** has satisfactorily completed the internship – 20MCA43 entitled **PYTHON WITH MACHINE LEARNING** in the academic year 2022-2023 as prescribed by VTU for the third semester of Master of Computer Applications.

Signature of the guide

Signature of HOD

INTERNSHIP CERTIFICATE FROM THE COMPANY

Serial No : PI/22/3081



CERTIFICATE OF COMPLETION

This internship certificate for brilliant and consistent high standards of workmanship signed this day by the duly authorized officers is proudly presented to

Mr/Ms. Adnan Ahad

For the successful completion of internship in

Python with Machine Learning

under Pie Infocomm Private Limited from **7th November, 2022** to **28th December, 2022**. It led to the

Completion of Project titled **"Catching Illegal Fishing"**



Vijay Kumar Jaiswal
(Director)

DECLARATION

I, **ADNAN AHAD (1AT21MC001)** hereby declare that this internship work titled **“PYTHON WITH MACHINE LEARNING”** is carried out under the guidance of **Dr DATTATREYA P MANKAME**, H.O.D, Department of MCA. This internship work is submitted to Visvesvaraya Technological University in partial fulfilment of the requirement for the award of the degree of **Master of Computer Applications** for the academic year 2022-2023.

Place: Bangalore

Date: / /2023

Signature of Student

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere gratitude to my college and my internship coordinator for their immense support and guidance throughout my internship journey.

Firstly, I want to thank my college for providing me with an excellent learning environment and equipping me with the skills and knowledge necessary to excel in my chosen field. My college has been instrumental in shaping my academic and professional journey, and I am grateful for the opportunities it has provided me to grow both personally and professionally.

Furthermore, I would like to express my heartfelt appreciation to Dr. Dattatreya P Mankame, my internship guide, for their invaluable guidance and support throughout my internship. From the outset, they provided me with the necessary resources, advice, and motivation to succeed in my role. Their expertise, knowledge, and willingness to guide me have been instrumental in shaping my internship experience and helping me to achieve my goals.

My internship guide Dr. Dattatreya P Mankame went above and beyond to ensure that my internship experience was both productive and fulfilling. They provided me with regular feedback, helped me navigate any challenges that arose and offered valuable insights into the industry. Their mentorship has been an invaluable asset to me, and I am grateful for the opportunity to learn from such a skilled and experienced professional.

In conclusion, I want to express my sincere appreciation to my college and coordinator for their unwavering support and guidance throughout my internship journey. They have played an integral role in shaping my professional development and helping me to achieve my goals. I look forward to applying the knowledge and skills I have gained during my internship to my future career endeavors.

DEPARTMENT VISION

To be a leading knowledge centre for imparting quality education in computer applications, interdisciplinary research and developing competent professionals to serve society.

DEPARTMENT MISSION

Department of MCA committed to :

M1: Empower graduates with knowledge, skills and attitude to develop highly competent professionals.

M2: Interdisciplinary projects, internships and value-added training programs in cutting-edge technologies.

M3: Collaborate with academia and industry for research, consultancy and encouraging entrepreneurship.

M4: Develop competent professionals with social and ethical values to serve society.

PROGRAMME EDUCATIONAL OBJECTIVES

Professionals from the MCA should:

PEO1: Develop the ability to plan, analyze, design, implement and maintain software products for real life applications.

PEO2: Practice effectively as individuals and as team members in multidisciplinary projects by inculcating professional skills, ethical behavior, and leadership qualities.

PEO3: Develop lifelong learning capabilities to set up their own enterprise in various sectors of computer applications and be able to pursue higher studies.

PROGRAM OUTCOMES

1. **Computational Knowledge:** Apply Knowledge of Computing Fundamentals, Computing Specialization, Mathematics, and Domain Knowledge appropriate for the Computing Specialization to the Abstraction and Conceptualization of Computing models from defined problems and requirements.

2. **Problem analysis:** Identify, Formulate, Research Literature, and solve Complex Computing problems reaching substantiated conclusions using fundamental Principles of Mathematics, Computing Sciences, and relevant Domain Disciplines.

3. **Design/development of solutions:** Design and Evaluate solutions for Complex Computing problems, and Design and Evaluate systems, Components, or Processes that meet specified needs with appropriate consideration for Public Health and Safety, Cultural, Societal, and Environmental considerations.

4. **Conduct investigations of complex problems:** Use Research-Based Knowledge and Research methods including design of Experiments, Analysis and Interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, Select, Adapt and Apply Appropriate techniques, resources, and Modern Computing tools to Complex Computing activities, with an understanding of the limitations.

6. **Professional Ethics:** Understand and Commit to Professional Ethics and Cyber regulations, Responsibilities, and norms of Professional Computing Practices.

7. **Life-long Learning:** Recognize the need, and have the ability, to engage in independent Learning for Continual Development as a Computing Professional.

8. **Project management and finance:** Demonstrate Knowledge and understanding of the Computing and Management Principles and Apply these to one's own work, as a member and leader in a team, to manage projects and in Multidisciplinary Environments.

9. **Communication Efficacy:** Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

10. **Societal and Environmental Concern:** Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practices.

11. **Individual and Team Work:** Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

12. **Innovation and Entrepreneurship:** Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

Program Specific Outcomes (PSOs)

Upon graduation students will be able to :

PSO1: efficiently design and develop computer-based systems in the areas related to algorithms, web design, big data analytics, database management, IOT and networking.

PSO2: apply standards practices and strategies in software project development using open-ended programming environments to provide quality solutions.

PSO3: inculcate modern technologies and platforms in building innovative career paths as an entrepreneur and pursuing higher studies.

TABLE OF CONTENTS

Contents	Page no.
I. Abstract	7
II. About the company	8
III. Internship Overview	10
IV. Internship Timeline	11
V. Training program	17
VI. Learning experiences	36
VII. Appendices: References and resources used	38

Chapter 1:

ABSTRACT

During my one-month internship at a professional organization, I was able to gain practical work experience and develop important professional skills in my field of interest. The internship provided me with an opportunity to apply my classroom learning to real-world situations, work alongside experienced professionals, and build valuable connections in my industry.

Throughout my internship, I had the opportunity to work on a variety of projects and tasks that were aligned with my career goals. I was able to develop new skills and techniques that are valuable in my field, such as project management and data analysis. I also learned how to navigate complex work environments and adapt to changing work priorities.

One of the most valuable aspects of my internship was the opportunity to work alongside experienced professionals in my field. I was able to learn from their experiences, gain new insights into my industry, and observe their work styles and approaches to problem-solving. This allowed me to see how classroom learning can be applied to real-world situations and develop a deeper understanding of the industry's best practices.

Throughout the internship, I was able to develop important professional skills that will be essential for my future career success. I gained experience working in a team environment, collaborating with colleagues to complete tasks and achieve project goals. I also developed my communication and time management skills, which will be valuable in any future career I pursue.

The internship also provided me with valuable networking opportunities. I was able to build relationships with professionals in my field, which will be helpful when seeking future employment opportunities. I also received positive feedback from my internship supervisors, which will be useful when applying for jobs and seeking referrals.

Overall, my one-month internship provided me with a range of valuable learning experiences and professional connections. It allowed me to apply my classroom learning to real-world situations, develop important professional skills, and build valuable connections in my industry. I am grateful for the opportunity and believe that it has had a positive impact on my future career success.

Chapter 2:

ABOUT THE COMPANY

PIE INFOCOMM PVT, LTD, is an Indian software-based company that provides specialized IT services and business solutions to make business operations easier. This company was introduced in 2002 when they provided business ideas and developed software for clients. They provide AutoCAD design, blueprints of motors and spare parts, they also develop chip-level designing using MATLAB technology and they also provide high-level scientific calculation. Pie Infocomm claims to have a strong focus on quality and customer satisfaction, and its website highlights the company's commitment to providing innovative solutions to its client's business challenges.



Figure 2.1 Logo of PIE INFOCOMM

The company was established in 2002 and is in Lucknow, Uttar Pradesh, and the current director is Vijay Kumar Jaiswal. PIE INFOCOMM PVT, LTD is an ISO-certified software development company with IT services and business solutions for the last 19 years. The company has a share capital of INR 1.00 lac and a total paid-up capital of INR 1.00 lac. They have many branches all over India which provide internships online based. They also provide placement opportunities to their trainees where students are recruited at various prestigious organizations.

Pie Infocomm Private Limited (PVT LTD) offers a range of IT services and solutions to its clients. Some of the products and services offered by the company include:

1. Custom Software Development: Pie Infocomm PVT LTD provides custom software development services to clients from various industries. The company's software development team uses modern technologies and frameworks to deliver customized solutions that meet clients' specific requirements.
2. Mobile App Development: Pie Infocomm PVT LTD offers mobile application development services for iOS and Android platforms. The company's mobile app development team creates customized mobile apps that are scalable, user-friendly, and engaging.
3. Web Development: Pie Infocomm PVT LTD also provides web development services that include designing, developing, and maintaining websites. The company's web development team has expertise in using various platforms and technologies like WordPress, Shopify, Magento, and more.
4. Cloud Computing: Pie Infocomm PVT LTD offers cloud computing services to help clients manage and optimize their cloud infrastructure. The company's cloud computing team provides services like cloud migration, cloud security, and cloud monitoring.
5. Digital Marketing: Pie Infocomm PVT LTD also offers digital marketing services to help clients increase their online visibility and reach. The company's digital marketing team provides services like search engine optimization (SEO), pay-per-click (PPC) advertising, social media marketing, and more.

Chapter 3:

INTERNSHIP OVERVIEW

As a student or recent graduate, the purpose of an internship for me is to gain practical work experience in my chosen field and to build a professional network. Internships provide the opportunity to apply the knowledge and skills I have learned in the classroom to real-world situations and to learn from experienced professionals in the industry.

During an internship, I will work on projects and tasks that are relevant to my field of study, allowing me to gain hands-on experience and develop new skills. I will have the opportunity to work alongside experienced professionals who can offer guidance and mentorship, and who can help me to better understand the industry and potential career paths. By completing an internship, will be able to gain insights into the day-to-day workings of your industry, which can help to make informed decisions about my career path.

Another important purpose of an internship is to help develop a stronger understanding of your strengths, weaknesses, and interests. By working in a real-world setting, will be able to see how my skills and knowledge apply to the industry, and gain a better understanding of the type of work that I enjoy and excel at. This can help to make more informed decisions about my future career path and to identify areas where I may want to focus on professional development.

Overall, the purpose of an internship is to gain practical work experience, build a professional network, and develop a deeper understanding of skills, strengths, and interests. By completing an internship, I will be better prepared to enter the workforce and pursue a successful career in my chosen field.

Chapter 4:

INTERNSHIP TIMELINE

1. Week – 1:

❖ What are translators and their types

A tool known as a language translator is utilized in the computer science field to convert code written in one programming language into another. Its core purpose is to facilitate the writing of code by programmers, enhancing their efficiency and minimizing errors, while also preserving the integrity of the final code product.

❖ String Operations, ID function, Type function, underscore operation

- String Operations: In the realm of Python programming, an abundance of string manipulation methods are available for the purpose of modifying strings. Among these, concatenation, slicing, and formatting are some of the most commonly utilized techniques.
- ID function: In Python programming, the `id()` function is responsible for returning a distinct identifier assigned to each object. This identifier is an integer that remains constant and unique throughout the lifespan of the object.
- Underscore Operation: In Python, the underscore symbol (`_`) can be used in a variety of ways. One common use is as a placeholder for values that are not used. For example:

❖ LIST and TUPLE with functions

In the world of Python programming, lists and tuples are highly prevalent data structures that enable you to accumulate a collection of values. However, there are some notable distinctions between these two structures. The following are several noteworthy characteristics and functionalities associated with both lists and tuples:

- Lists in Python possess a mutable nature, implying that the elements within them can be altered even after their creation.
- Lists can contain elements of different types.
- Tuples are immutable, which means that you cannot change their contents after they are created.
- Tuples can contain elements of different types.

❖ Control Statements

In Python, there are three main types of control statements: conditional statements, loops, and function calls.

- **Conditional statements:** Conditional statements provide the ability to execute specific blocks of code based on whether a certain condition is true or false. In Python programming, the most frequently utilized conditional statement is the if statement. However, additional clauses such as elif and else are available to assess multiple conditions.
- **Loops:** Loops allow you to repeat a block of code multiple times. There are two main types of loops in Python: for loops and while loops.
- **for loops:** for loops are used when you know the number of times you want to repeat the code. They are often used to iterate over a sequence (such as a list or a string).
- **while loops:** while loops are used when you don't know how many times you want to repeat the code, but you know the condition that should cause the loop to stop.

2. Week – 2:

❖ Difference between scalar addition & vector addition.

- Scalar addition and vector addition are fundamental operations in the fields of mathematics and physics, but they function on distinct categories of mathematical entities.
- Scalar addition is the process of adding one numerical value (a scalar) to another numerical value. This operation is both commutative and associative, which means that the order in which the addition occurs does not impact the result, and you can regroup the terms in any manner you choose.
- Vector addition, on the other hand, involves adding two vectors together. A vector is a mathematical object that has both magnitude (length) and direction. When you add two vectors, you combine their magnitudes and directions to form a new vector that represents the sum of the two original vectors. Vector addition is not commutative (i.e., the order matters), but it is associative. For example, if vector A has magnitude 3 and points to the right, and vector B has magnitude 4 and points up, then $A + B$ is a new vector that has magnitude 5 and points diagonally up and to the right. Vector

addition is used to combine physical quantities that have both magnitude and direction, such as velocities, forces, and electric fields.

❖ **Difference between package and module.**

- A module in Python refers to a single file that consists of Python code that performs a specific functionality or set of functionalities. These modules can be imported and used in other Python scripts or modules by using the import statement. Modules can contain various objects, including variables, functions, classes, etc., which can help organize and reuse code across multiple Python scripts or projects.
- A package is a collection of modules that are arranged in a hierarchical directory structure. It can include other packages and modules and is utilized to organize larger code projects that require multiple files and functionalities. The package is identified by a special file called "**init.py**", which is executed when the package is imported. The "**init.py**" file is used to initialize the package's namespace and import other modules or packages.

❖ **Using of functions.**

- A function in Python is a block of code that performs a specific task or set of tasks. Functions are used to break down complex programs into smaller, more manageable pieces of code that can be reused throughout the program or even in other programs.
- In Python, you can define a function using the keyword "**def**" followed by the function name and a set of parentheses. The parameters that the function accepts can also be specified inside the parentheses. These parameters serve as inputs to the function that the function can use to perform its task.

❖ **Objects and functions.**

- In object-oriented programming, a class is a template or a blueprint that defines the attributes (properties or characteristics) and methods (functions or behaviors) of a particular type of object. Once a class is defined, you can create multiple instances or objects of that class, each with its own unique state and behavior, based on the blueprint provided by the class. In other words, a class provides a set of instructions for creating and managing objects, and serves as a fundamental building block for creating complex software systems.

- For example, a class called Person might define attributes such as name, age, and gender, as well as methods such as speak() or walk(). An object is an instance of a class that has its own unique values for the attributes defined in the class.
- To use a class in Python, you need to first define the class with the class keyword, specifying its attributes and methods. Once the class is defined, you can create an instance of the class by calling its constructor method, which is usually named **init**. The **init** method is used to initialize the object's attributes, which are defined within the class. Creating an instance of a class allows you to use the methods and access the attributes of that specific object.

3. Week – 3:

❖ Creating tkinter

To create a tkinter project in Python, you will need to follow a few basic steps:

- Import the tkinter module: The first step is to import the tkinter module by using the statement `import tkinter`.
- Create the main window: You can create the main window of your application by using the `Tk()` function from the tkinter module. This will create a blank window that you can add widgets to.
- Add widgets: You can add widgets such as buttons, labels, and text boxes to your application by using the various tkinter classes such as `Button`, `Label`, and `Entry`.
- Define event handlers: You can define functions that will be called when a widget is clicked or some other event occurs. These functions are referred to as event handlers.
- Start the main loop: The final step is to start the tkinter main loop using the `mainloop()` function. This will display the window and allow the user to interact with your application.
- **Example project:**
- Cryptography means changing the text of a message so that people who don't know
- your secret never understand your message. If the user chooses to encrypt the message,
- the user's message must be transformed into a secret code. But if the user chooses to
- decrypt the message, your program should be able to convert a secret code into a meaningful text.
- If the user chooses to encrypt the message, the user's message must be transformed into a secret code. But if the user chooses to decrypt the message, your program should be able to convert a secret code into a meaningful text.

- According to user input, we need to write an event program because the operation of the program depends on user input. Here we can use the dialogue box to get user input and the info box to show the encrypted and decrypted message to the user.
- As stated before, I will be using an infinite loop, so the program will keep running until the user wants to encrypt and decrypt using Python. The program will end at the point where the user gives input other than “encrypt” and “decrypt”.

4. Week – 4:

❖ **Creating a machine learning example project.**

Illegal fishing, which is also referred to as "IUU" fishing (Illegal, Unreported, and Unregulated), poses a significant worldwide issue that undermines efforts to sustainably manage fisheries and jeopardizes the livelihoods of millions of people who rely on fish as a source of income and nourishment. One solution to combat IUU fishing is to leverage technology to monitor fishing activities and detect illegal fishing practices

The project is composed of several components, including data collection, data processing, machine learning, and user interface.

- The initial step of the "Catinching" project is data collection. This involves gathering satellite images of fishing vessels, which can be obtained through a combination of publicly available satellite imagery and privately collected imagery. These images are then stored in a database for subsequent analysis
- The next step in the "Catinching" project is to process the collected satellite images. This involves preprocessing the images to make them suitable for analysis. The preprocessing step may include tasks such as adjusting the colour balance, removing artifacts, and cropping the images to focus on the fishing vessels. These tasks help to improve the quality of the images and make them easier to analyze using machine learning algorithms. The processed images are then ready for the next step in the project, which is to train the machine learning models.
- Machine learning: The main component of the Catinching project is a machine learning algorithm that uses satellite images to identify fishing vessels engaged in illegal fishing practices. The algorithm is designed to analyze the images and detect vessels that are using small nets to catch juvenile fish, known as catinching. To achieve this, the algorithm is trained on a dataset of labeled images that indicate whether or not a fishing vessel is engaged in catinching. The algorithm utilizes a convolutional neural network (CNN) architecture, which is a type of deep learning algorithm that can extract

features from images and make predictions about the presence of catinching. The algorithm is continually updated with new data to improve its accuracy and effectiveness in identifying illegal fishing practices.

- User interface: The last component of the "Catinching" project is a user interface that enables users to view the outcomes of the machine learning algorithm and take appropriate measures against the identified vessels. The interface presents a map of fishing vessels and their locations, and overlays details about which vessels have been recognized as engaging in illegal fishing practices. Users can choose a vessel and examine thorough information about its fishing operations, such as pictures of the vessel and its catch.

The Catinching project has the potential to bring numerous benefits. By leveraging technology to monitor fishing activities and identify illegal practices, it can contribute to the prevention of IUU fishing and support sustainable fishing practices. The project can also provide valuable data to fisheries managers and policymakers, enabling them to make informed decisions on managing fishery resources. Furthermore, the project can help safeguard the livelihoods of millions of people who rely on fish as a primary source of food and income.

The Catinching project has to overcome several challenges before it can be effectively implemented. One of the challenges is the high cost of collecting and processing satellite imagery, which can limit the project's accessibility to smaller organizations and governments. Another challenge is the need for accurate and reliable labeling of training data for the machine learning algorithm, which can be time-consuming and costly. Additionally, the project needs to address potential legal and ethical concerns related to privacy and surveillance. It is crucial to ensure that the project respects the rights of fishing vessel operators and complies with legal and ethical standards.

In summary, the Catinching project is a promising tool in the fight against IUU fishing. Its ability to use machine learning algorithms to analyze satellite images of fishing vessels can help deter illegal fishing practices and promote sustainability in the fishing industry. However, the project's high costs, the need for accurate labeling of training data, and legal and ethical concerns around privacy and surveillance are significant challenges that must be addressed for the project to be successful. Overall, the Catinching project demonstrates the potential of technology to promote sustainable fishing practices, but it also highlights the need for careful consideration of its limitations and challenges.

Chapter 5

TRAINING PROGRAM

1. Week - One:

❖ Translator and its types:

A translator refers to a software tool that serves as a medium to convert one programming language into another. Its primary objective is to aid programmers in writing code with greater efficiency while ensuring that the code is both accurate and free of errors.

There are three main types of translators in computer science: compilers, interpreters, and assemblers.

- **Compilers:** A software tool known as a compiler takes source code written in a high-level programming language and translates it into machine code that the computer's CPU can understand and execute. This procedure encompasses multiple stages, including lexical analysis, syntax analysis, code generation, and optimization. Compilers are well-regarded for producing faster and more effective code compared to other forms of translators.
- **Interpreters:** An interpretive program is a software application that sequentially reads and executes high-level programming language code. In contrast to compilers, which translate entire programs into machine code before execution, interpreters interpret code line-by-line, providing more flexibility and ease of use. Although they are generally slower than compilers, they are commonly employed in scripting languages like Python and JavaScript.
- **Assemblers:** An assembler is a software program that translates assembly language code, which is a type of low-level programming language that employs mnemonic codes to represent CPU instructions, into machine code that can be executed by a computer. The primary use of assemblers is to write system-level code, such as device drivers and operating systems.

❖ Python introduction:

Python is a programming language that was created by Guido van Rossum and released in 1991. It is an interpreted language that is considered to be high-level, and it is renowned for

its simplicity, readability, and user-friendliness. These qualities have contributed to its widespread popularity, and it is now one of the most widely-used programming languages in existence. Python is utilized across a variety of domains, including web development, scientific computing, artificial intelligence, machine learning, and data science.

❖ **IDLE introduction:**

- IDLE, or Integrated Development and Learning Environment, is an IDE (Integrated Development Environment) designed specifically for the Python programming language. It is included as part of the standard Python distribution and offers a user-friendly environment for both novice and experienced programmers alike. With its convenient features, IDLE provides an easy way for beginners to learn Python, while also offering advanced functionalities for debugging and developing Python code.
- IDLE is equipped with a range of features that are highly beneficial for Python developers, such as syntax highlighting, code autocompletion, debugging tools, and the ability to run Python scripts directly within the IDE. Additionally, it includes an interactive Python shell, which enables users to experiment with and test code snippets in real time. These functionalities make IDLE an ideal choice for both beginners and experienced Python developers who are seeking a comprehensive and user-friendly development environment.

❖ **String operations, ID function, Type function, underscore operation:**

- String Operations Python offers a multitude of string manipulation operations. Among the most frequently used operations are concatenation, slicing, and formatting. These operations allow for the seamless manipulation of strings and are widely utilized in various Python applications, ranging from basic data processing tasks to complex algorithms in machine learning and natural language processing.
- ID function: In Python, the `id()` function returns a distinct identifier for each object. This identifier is a unique integer value that remains constant throughout the object's lifetime. As a result, the `id()` function is often used in Python programs to identify and track individual objects.
- Underscore Operation: In Python, the underscore symbol (`_`) can be used in a variety of ways. One common use is as a placeholder for values that are not used.

❖ LIST and TUPLES with function:

Lists and tuples are popular data structures in Python, frequently used for storing multiple values. However, there are some important distinctions between these two data types. Here are some of the primary characteristics and operations associated with lists and tuples:

1. Lists:

- Lists are mutable, which means that you can change their contents after they are created.
- Lists can contain elements of different types.
- Some of the most commonly used functions for lists include:
- `len()`: returns the number of elements in a list.
- `append()`: adds an element to the end of a list.
- `insert()`: adds an element at a specific position in a list.
- `remove()`: removes the first occurrence of an element from a list.
- `sort()`: sorts the elements of a list in ascending order.
- `reverse()`: reverses the order of the elements in a list.

2. Tuples:

- Tuples are immutable, which means that you cannot change their contents after they are created.
- Tuples can contain elements of different types.
- Some of the most commonly used functions for tuples include:
- `len()`: returns the number of elements in a tuple.
- `index()`: returns the index of the first occurrence of a specified element in a tuple.
- `count()`: returns the number of times a specified element appears in a tuple.

❖ Control statements:

In Python, there are three main types of control statements: conditional statements, loops, and function calls.

- Conditional statements are a fundamental part of programming, and Python provides several options for executing different blocks of code based on the evaluation of certain conditions. The most commonly used conditional statement in Python is the "if" statement,

which allows you to execute a block of code only if a certain condition is true. In addition to the "if" statement, Python also provides "elif" and "else" clauses, which allow you to test multiple conditions and execute different blocks of code accordingly. These conditional statements provide a powerful mechanism for controlling the flow of your program based on various conditions.

- **Loops:** Loops and conditional statements are fundamental control structures in Python. Conditional statements are used to execute different blocks of code based on whether a certain condition is true or false. The most common conditional statement in Python is the if statement, but there are also elif and else clauses that allow you to test multiple conditions.
- **for loops:** loops are used when you know the number of times you want to repeat the code. They are often used to iterate over a sequence (such as a list or a string).
- **while loops:** while loops are used when you don't know how many times you want to repeat the code, but you know the condition that should cause the loop to stop.
- **Function calls:** Function calls allow you to execute a block of code that has been defined elsewhere in your program. Functions can take arguments (inputs) and return values (outputs).

2. Week - Two:

❖ Introduction to NumPy:

NumPy is a library for the Python programming language that is commonly used for scientific computing and data analysis. It is an acronym for "Numeric Python". NumPy offers a fast and efficient means to carry out numerical operations on arrays and matrices in Python.

Some of the key features of NumPy are:

- NumPy's primary object is the ndarray (n-dimensional array), which is a powerful multi-dimensional container for homogeneous data. It can represent arrays of various shapes and sizes, including vectors, matrices, and higher-dimensional arrays. The ndarray provides a fast and efficient way to store and manipulate large arrays of numerical data.
- **Broadcasting:** NumPy provides the ability to perform arithmetic operations on arrays of different shapes and sizes using broadcasting. Broadcasting allows NumPy to operate on arrays of different shapes and sizes as if they were the same size by automatically aligning their dimensions. This can greatly simplify the code needed to perform element-wise operations on arrays of different shapes, and can also improve performance by avoiding the need to create temporary arrays.

- **Mathematical functions:** NumPy library offers a variety of mathematical functions that can be applied to arrays to perform operations. These functions include trigonometric functions such as `sin`, `cos`, and `tan`, as well as exponential and logarithmic functions such as `exp` and `log`. NumPy also provides functions for statistical analysis, linear algebra, and Fourier transforms, among others.
- **Linear algebra:** NumPy provides a wide range of linear algebra functions, making it a popular choice for scientific computing and data analysis. These functions can be used to perform matrix and vector operations, such as matrix multiplication, matrix inversion, and calculating eigenvalues and eigenvectors. The linear algebra functions in NumPy are implemented using highly optimized and efficient algorithms, making them suitable for large-scale numerical computations.

❖ **Different ways to work with array functions:**

- `np.array()`: This function creates an array from a Python list or tuple.
- `np.linspace()`: This function creates an array with a specified number of evenly spaced values within a given interval.
- `np.arange()`: This function returns an array with evenly spaced values within a given interval.
- `np.logspace()`: This function creates an array with a specified number of logarithmically spaced values within a given interval.
- `np.zeros()`: This function creates an array filled with zeros.
- `np.ones()`: This function creates an array filled with ones.

Each of these functions in NumPy has specific parameters that can be used to customize the resulting array. For example, the `np.linspace()` function takes the start and end points of the interval, as well as the number of values to create, as input parameters. This allows the user to create arrays with specific properties and characteristics, tailored to their needs.

NumPy provides a variety of functions that can be used to create arrays with different values and shapes. These functions take different parameters to customize the resulting arrays. For example, `np.linspace()` takes the start and end points of the interval, as well as the number of values to create, as parameters. Similarly, `np.arange()` takes the start and end points of the interval, as well as the step size between values, as parameters. `np.logspace()` takes the start and end points of the interval, as well as the number of values

to create, and whether the values should be spaced logarithmically or not, as parameters. The resulting arrays can be used for a wide range of numerical computations and scientific applications..

❖ **Addition to NumPy array:**

- Scalar addition in NumPy refers to adding a single value to each element in a NumPy array. This mathematical operation can be performed using the `+` operator or the `np.add()` function, with the array and the scalar value as the operands. The resulting array will have the same shape as the original array, with each element increased by the scalar value. This operation can be useful for situations where you need to increase the values of an array by a constant amount.
- Vector addition in NumPy refers to the mathematical operation of adding two arrays element-wise. This means that each element in the first array is added to the corresponding element in the second array. This operation can be performed using the `+` operator or the `np.add()` function, where the two arrays are the operands. In order to add two arrays element-wise, the arrays must have the same shape. The resulting array will also have the same shape as the original arrays, with each element being the sum of the corresponding elements in the two arrays. Vector addition is a fundamental operation in linear algebra and is used in a wide range of scientific and engineering applications.

❖ **Mathematical and Trigonometrical functions using NumPy:**

NumPy provides a wide range of mathematical and trigonometric functions that can be applied to arrays. Here are some examples:

- `np.sin()`: This function computes the sine of each element in the input array.
- `np.cos()`: This function computes the cosine of each element in the input array.
- `np.tan()`: This function computes the tangent of each element in the input array.
- `np.exp()`: This function computes the exponential of each element in the input array.
- `np.log()`: This function computes the natural logarithm (base e) of each element in the input array.
- `np.log10()`: This function computes the base 10 logarithm of each element in the input array.
- `np.sqrt()`: This function computes the square root of each element in the input array.
- `np.abs()`: This function computes the absolute value of each element in the input array.

- `np.power()`: This function raises each element in the input array to a specified power.

❖ Copying NumPy array:

NumPy offers a convenient way to create a new array that is a duplicate of an existing one, using the `copy()` method. When you apply the `copy()` method to an array, it returns a brand new array that contains the same data as the original. Importantly, any changes made to the duplicate array will not affect the original array, and vice versa.

There are three different ways to create a new array that is based on an existing array: aliasing, shallow copying, and deep copying.

- Aliasing: When you create a new variable that refers to an existing array, you are creating an alias. Any changes made to the alias will affect the original array.
- Deep Copy: A deep copy creates a new array object with new data. The data of the new array is not pointing to the same memory location as the original array. Any modifications made to the new array will not affect the original array, and vice versa.
- Shallow Copy: A shallow copy creates a new array object that is a view of the original array object. The data of the new array is still pointing to the same memory location as the original array. Any modifications made to the new array will affect the original array, and vice versa.

3. Week - Three:

❖ Matrices using NumPy array:

NumPy is a well-known Python module that has gained wide popularity in scientific computing, particularly when dealing with matrices and arrays. In NumPy, matrices can be represented as 2D arrays, and there are several built-in functions that can be used to create, manipulate, and execute operations on matrices.

❖ Function in Python:

Python functions are sections of code that can be reused to perform particular tasks. Functions are incredibly useful for dividing a program into smaller and more manageable segments, as well as improving the readability and organization of the code. In Python, defining a function

requires using the "def" keyword, followed by the function's name and its arguments in parentheses.

The various components of a Python function in more detail:

Function Name: When you create a function in Python, the name of the function is what you will use to reference and call the function from other parts of the program. For this reason, it's crucial to choose a descriptive name that accurately conveys what the function does. By selecting a descriptive name for your function, you make it easier for yourself and other developers to understand the function's purpose and role within the codebase.

Function Arguments: In Python, arguments are values that are passed to a function when it is called. Functions can take any number of arguments, including zero, and each argument should be separated by a comma. In the previous example, the "greet" function takes a single argument, which is the name of the person being greeted.

Function Body: In Python, the function body refers to the block of code that is executed when the function is called. This block of code is defined using indentation, and all the code inside the function should be indented to indicate that it is part of the function body. The function body can contain any valid Python code, including other functions, loops, conditional statements, and variable assignments. When the function is called, the interpreter will execute the code inside the function body according to the instructions provided in the function definition..

Return Statement: A function can optionally return a value or a set of values to the calling code. This is done using the return statement. The returned value can be assigned to a variable or used directly in the calling code.

❖ **OOP's concept:**

Object-Oriented Programming (OOP) is a programming approach that revolves around the concept of objects. Objects are created from classes, which combine data and behavior into a single entity. OOP offers a modular and reusable approach to writing code, and it is widely used in Python programming..

Python has built-in support for OOP, and it includes the following concepts:

- **Classes and Objects:** In programming, a class is a template or blueprint that defines the attributes and methods of an object. Attributes represent the data or variables that

belong to an object, while methods represent the functions or behaviour that an object can perform. When an object is created from a class, it is called an instance, and it has its own unique set of data and behaviour based on the attributes and methods defined in the class.

- **Inheritance:** Inheritance is a programming concept that enables a class to acquire properties and methods from another class. The class being inherited from is referred to as the parent class or base class, while the class that inherits from it is called the child class or derived class. By using inheritance, the child class can reuse code from the parent class, which helps to reduce code duplication and improve code organization.
- **Polymorphism:** Polymorphism is a feature of object-oriented programming that allows objects to take on multiple forms or behaviors. In Python, polymorphism can be achieved through method overloading and method overriding. Method overloading involves defining multiple methods with the same name but different parameters in a class. The correct method is chosen based on the number and types of arguments passed when the method is called. Method overriding involves defining a method in a child class with the same name as a method in the parent class. The method in the child class overrides the behaviour of the method in the parent class for objects of the child class. This allows objects of the child class to exhibit different behaviour than objects of the parent class when calling the same method.
 1. Method overloading is a programming concept that allows a class to have multiple methods with the same name but different parameters. When the method is called, the correct method is chosen based on the number and types of arguments that are passed to it. In Python, method overloading is not directly supported, but it can be achieved through the use of default parameter values and variable-length argument lists.
 2. Method overriding is a feature of object-oriented programming that allows a child class to provide its own implementation of a method that is already defined in the parent class. This allows the child class to customize or extend the behavior of the parent class. When the method is called on an object of the child class, the overridden method is called instead of the parent method. This can be useful in situations where you want to reuse the code of the parent class but need to modify its behavior in some way. The method signature, including the method name,

return type, and parameters, must be the same in both the parent and child classes for method overriding to work correctly.

- **Abstraction:** This refers to the concept of abstraction in programming, which involves hiding the implementation details of a system and only exposing the necessary functionality to the user. This is typically done by providing a simplified and easy-to-use interface that shields the user from the underlying complexity. Abstraction is an important concept in software design, as it allows developers to create more modular and maintainable code by reducing the dependencies between different parts of the system..

❖ **Constructors and its types:**

In object-oriented programming, a constructor is a special method that gets called automatically when an object is created. Its purpose is to initialize the object's attributes and prepare it for use. In Python, the constructor method is named `__init__()`. It is defined like any other method but has a special name and takes the 'self' parameter, which refers to the object being created.

There are two types of constructors in Python:

- **Default Constructor:** A default constructor is a constructor that doesn't take any arguments. If you don't define a constructor for a class in Python, a default constructor is automatically provided. The default constructor initializes the object with no attributes.
- **A constructor that takes one or more arguments and is defined by the programmer to assign specific values to an object's attributes is referred to as a parameterized constructor.** This type of constructor is often utilized in object-oriented programming to provide a more tailored and specific approach to object initialization. The parameterized constructor can receive any number of arguments that the programmer specifies, enabling a higher degree of flexibility and customization in object initialization. By utilizing this type of constructor, the programmer can easily provide specific values to an object's attributes without having to rely on default values or external methods of initialization. Overall, the use of parameterized constructors in programming can lead to a more efficient and precise approach to object initialization and management.

❖ Types of methods:

In object-oriented programming, methods are functions that are associated with a class and operate on the object of that class. There are three types of methods in Python

- **Instance Methods:** In object-oriented programming, instance methods are methods that are associated with an object of a class. These methods require the "self" parameter, which refers to the particular instance of the class that the method is being called on. The main advantage of instance methods is that they have access to and can modify the attributes of the instance, allowing for greater flexibility and customization in the behaviour of the object. By modifying the object's attributes, instance methods can perform various operations that are specific to the object, such as updating its state or performing calculations based on its properties. Overall, instance methods are a powerful tool in object-oriented programming that allows for dynamic and customized behavior of objects.
- **Class Methods:** Class methods in object-oriented programming are associated with a class, rather than a specific instance of that class. Unlike instance methods, which require the "self" parameter to refer to the instance, class methods require the "cls" parameter to refer to the class itself. This allows class methods to manipulate class-level attributes and perform operations that involve the class as a whole. For example, class methods can be used to create a new instance of the class or to initialize a group of class-level attributes. By using class methods, developers can create more efficient and streamlined code that is specific to the class and does not require an instance to be created first. This makes class methods a powerful tool in object-oriented programming, enabling developers to manipulate class-level attributes and behavior in a more controlled and efficient manner. Overall, class methods are an important aspect of object-oriented programming that enable developers to write more flexible and reusable code..
- **Static Methods:** In object-oriented programming, static methods are functions that are not associated with a specific instance or class. They do not require any special parameters, such as "self" or "cls", and can be defined as standalone functions within the class. Unlike instance methods and class methods, static methods do not have access to the attributes of the object or the class. Instead, they are used to define functions that are related to the class but do not depend on any specific state. An example of a static method might be a utility function that performs a generic task,

such as sorting an array or converting data types. Overall, static methods are a useful tool for organizing and encapsulating related functions within a class, without having to rely on the state of any specific object or class.

❖ Introduction to tkinter:

- Tkinter is a Python GUI (Graphical User Interface) toolkit that provides a set of standard GUI elements (widgets) to build desktop applications. It is built on top of the Tcl/Tk GUI toolkit and is implemented as a Python wrapper around it. Tkinter is a popular choice for building GUI applications due to its ease of use, portability, and compatibility with a wide range of platforms.
- Tkinter provides a variety of GUI widgets such as buttons, labels, text boxes, canvas, menus, and more. These widgets can be arranged using geometry managers like grid, pack, and place. Tkinter also provides support for event-driven programming, which allows you to write code that responds to user input events like button clicks or key presses.
- To use Tkinter in your Python program, you need to import the tkinter module. Once you have imported the module, you can create a top-level window by creating an instance of the Tk() class:

This will create a new window with default settings. You can customize the window's appearance by setting its properties, such as the title, size, and background color:

You can also add widgets to the window using the various widget classes provided by Tkinter. For example, to add a label to the window, you can create an instance of the Label class and add it to the window using the pack() method:

Tkinter also provides support for event-driven programming. You can register event handlers to respond to user input events like button clicks or key presses:

4. Week - Three:

❖ Introduction to machine learning and types ML algorithms:

Machine learning is a branch of artificial intelligence that deals with the creation of algorithms and models that enable computers to learn and improve from experience, without requiring explicit programming. Essentially, machine learning enables computers to automatically learn from data, recognize patterns and connections within that data, and use that information to make decisions or predictions..

Machine learning can be divided into three main types: supervised learning, unsupervised learning, and reinforcement learning.

- Supervised learning is a type of machine learning that involves training a model on labeled data. In this approach, the algorithm is provided with inputs and their corresponding correct outputs. The goal of supervised learning is to learn a function that can map inputs to outputs accurately. Once the model is trained, it can then be used to make predictions on new data that it has never seen before. Examples of supervised learning applications include image classification, where the model learns to identify objects in images based on labeled training data, speech recognition, where the model learns to transcribe spoken words into text using labeled audio data, and predicting stock prices, where the model learns to make predictions on future stock prices based on labeled historical data.
- Unsupervised learning involves training a model on unlabeled data, where the goal is to discover hidden patterns and relationships in the data. Clustering, anomaly detection, and dimensionality reduction are examples of unsupervised learning.
- Reinforcement learning involves training a model to make decisions based on feedback received from the environment. The model learns by trial and error, with the goal of maximizing a reward function. Applications of reinforcement learning include game playing, robotics, and recommendation systems.
- Machine learning, an advanced discipline of computer science, is utilized in an extensive array of applications spanning various fields, such as natural language processing, image and video processing, healthcare, finance, and several others. In the domain of machine learning, decision trees, neural networks, support vector machines, and clustering algorithms are some of the customary methodologies that are commonly employed to achieve the desired results.

❖ Introduction to Scikit-learn:

Scikit-learn is an influential open-source library for machine learning, particularly in Python. This library is established on top of the SciPy, NumPy, and matplotlib libraries and renders an uncomplicated and efficient instrument for mining data and analyzing it. Scikit-learn supplies a broad array of tools to assist with different machine learning duties like regression, classification, clustering, and dimensionality reduction. Furthermore, Scikit-learn presents a user-friendly interface that makes it easy to operate with data and apply machine learning algorithms. It encompasses several conventional machine learning algorithms, like decision

trees, random forests, k-nearest neighbors, support vector machines, and gradient boosting, and also incorporates tools for data preprocessing, feature extraction, and model selection.

One of the strengths of scikit-learn is its consistent API. This makes it easy to switch between different machine learning algorithms and to try different models without having to change the code significantly. Scikit-learn also provides a range of evaluation metrics, such as accuracy, precision, recall, and F1 score, to measure the performance of the models.

Scikit-learn is a widely adopted machine learning library that has been embraced by both industry and academia. The library boasts a large and vibrant community of developers and users, contributing to its continuous improvement and the addition of new algorithms and functionalities. It is frequently updated to keep up with the latest advancements in machine learning.

❖ **Types of machine learning:**

Machine learning (ML) is an expansive domain that encompasses various approaches and techniques to analyze and learn from data. Broadly speaking, machine learning can be categorized into three distinct types:

- **Supervised learning:** Supervised learning is a type of machine learning where the model is trained on labelled data, where inputs or features are mapped to corresponding outputs or labels. The primary objective of supervised learning is to predict the labels of unseen data by utilizing the patterns observed in the training data. Regression and classification are the most frequently used supervised learning a, which aim to model continuous and discrete outputs.
- **Regression** is a type of supervised learning that is used to predict continuous values, such as forecasting the price of a house based on its size and location. On the other hand, **classification** is another type of supervised learning that is used to predict categorical values, such as determining whether an email is spam or not based on its content.
- **Unsupervised learning:** In unsupervised learning, the model is trained on unlabeled data, and the primary objective is to identify patterns or structures within the data without any prior knowledge of the labels. Clustering and dimensionality reduction are the most widely used unsupervised learning algorithms to accomplish this task..

- Clustering is a technique used to group similar data points together based on their intrinsic properties. This approach is useful when there is no predefined set of labels or classes for the data. For instance, clustering can be used to group customers based on their purchasing behavior, which can help businesses tailor their marketing strategies to different customer groups.
- Reinforcement learning: Reinforcement learning is a machine learning approach where the model interacts with an environment and learns from the feedback received in the form of rewards or punishments. The goal is to develop a policy that maximizes the cumulative reward obtained over time. Reinforcement learning is widely used in various applications, such as game playing, robotics, and autonomous driving..

❖ **Matplotlib:**

Matplotlib is a widely used Python library for creating visualizations of data. Developed by John Hunter, it is now maintained by a team of developers. Matplotlib offers a variety of customizable and interactive plot types, such as line, scatter, bar, histogram, pie, and many others. It also provides advanced features like animations, 3D plotting, and subplots. With Matplotlib, users can create publication-quality visualizations that can help communicate insights and findings from data analysis.

Here are some of the key features of Matplotlib:

- Easy to use: Matplotlib is designed with ease of use in mind. Its API is straightforward and easy to learn, making it a great choice for beginners as well as advanced users. The library also provides a wide range of customization options, allowing users to create plots that are tailored to their specific needs. This ease of use and customization is one of the key reasons why Matplotlib is so popular among data scientists and analysts.
- Highly customizable: Matplotlib provides a wide range of customization options, allowing users to create plots with their desired styles and aesthetics. For example, you can customize the color, size, and shape of every element of your plot, including the axes, labels, legends, and titles. You can also add text, annotations, and images to your plot to make it more informative and visually appealing.
- Support for multiple output formats: Matplotlib provides the ability to save plots in different formats, including PNG, PDF, SVG, and more, making it effortless to use

Matplotlib plots in various contexts such as reports, presentations, and web applications..

- Wide range of plot types: Matplotlib is a versatile library that supports a wide range of plot types, including line plots, scatter plots, bar plots, histogram, pie charts, and more. Additionally, Matplotlib provides advanced plotting capabilities such as 3D plots, heatmaps, and contour plots.
- Interactive plots: Matplotlib provides interactive plots that allow you to zoom, pan, and hover over data points to see more information, making it easier to explore and analyze data. This feature is particularly useful when working with large datasets or complex visualizations.

❖ Types of matplotlib plots:

Matplotlib provides a wide range of customizable and interactive plot types. Here are some of the most commonly used types of plots:

- Line Plot: A line plot is a basic plot type that displays data as a series of points connected by a straight line. It is commonly used to show trends over time.
- Scatter Plot: A scatter plot is a plot type that displays the relationship between two variables as a set of points on a 2D plane. It is useful for detecting patterns or trends in data.
- Bar Plot: A bar plot is a plot type that displays data using bars of different heights or lengths. It is commonly used to compare data between different categories or groups.
- Histogram: A histogram is a plot type that displays the distribution of a set of continuous data. It groups the data into bins and displays the frequency of each bin as a bar.
- Pie Chart: A pie chart is a plot type that displays data as a set of slices of a circle. It is commonly used to show the proportion of different categories or groups.
- Box Plot: A box plot is a plot type that displays the distribution of a set of continuous data. It shows the median, quartiles, and outliers of the data in a box-and-whisker format.
- Heatmap: A heatmap is a plot type that displays the distribution of data using colors. It is commonly used to visualize correlation between variables or to highlight patterns in large datasets.

- 3D Plot: A 3D plot is a plot type that displays data in three dimensions. It is commonly used to visualize complex data that cannot be displayed in two dimensions

❖ Regression algorithm:

Regression is a supervised machine learning algorithm utilized to predict the continuous value of a target variable. It is based on one or more predictor variables, and its main objective is to model and scrutinize relationships between variables, which are widely applied in the field of statistics.

Regression models have a wide range of applications, such as predicting stock prices, estimating the impact of marketing campaigns, and forecasting sales figures. The main objective of regression is to determine the best-fit line or curve that describes the relationship between the predictor variables and the target variable.

There are several types of regression models, including:

- Simple Linear Regression: This model is used when there is a linear relationship between the predictor and target variables.
- Multiple Linear Regression: This model is used when there are multiple predictor variables that can affect the target variable.
- Polynomial Regression: This model is used when the relationship between the predictor and target variables is non-linear.
- Logistic Regression: This model is used for classification problems, where the target variable is binary (0 or 1).
- Ridge Regression: Ridge Regression is typically utilized to address the problem of multicollinearity, which arises when the predictor variables are highly correlated with each other.
- Lasso Regression: Lasso Regression is used for feature selection, where only the most crucial predictor variables are taken into consideration while building the model.

Several performance metrics are used to evaluate the effectiveness of regression models, including mean squared error, R-squared value, and root mean squared error. The selection of a particular performance metric depends on the problem's nature and the objectives of the analysis.

❖ **Classification algorithm:**

Supervised machine learning algorithms known as classification algorithms have the ability to foretell the group or kind of an objective variable by considering one or more forecast variables. There is an assortment of classification algorithms accessible, each with its individual merits and demerits. Here are several of the commonly used classification algorithms:

- **Logistic regression** is a type of mathematical model that is commonly used for solving binary classification problems, which means predicting one of two possible outcomes. This model makes use of a logistic function to map the input data to a probability value that ranges from 0 to 1. The logistic function is a mathematical formula that helps to model the relationship between the target variable and the predictor variables.
- **Naive Bayes:** Naive Bayes is a statistical model that uses Bayes' theorem to predict the probability of a target variable belonging to a particular class based on the probability of the predictor variables. The model assumes that the predictor variables are independent of each other, which is a simplistic assumption, hence the name "naive". Naive Bayes is often used for tasks such as text classification and spam filtering because it can handle large amounts of data and is relatively fast and accurate.
- **Decision Trees:** Decision trees are a type of model used for solving classification problems, where we want to assign a label or category to each data point. They are called "non-parametric" because they do not make assumptions about the underlying distribution of the data.
- **Random Forest:** Random forest is a machine learning model that combines several decision trees to enhance the accuracy and reliability of the classification. It achieves this by randomly selecting a portion of the predictor variables and constructing a decision tree for each selected subset. The ultimate prediction is based on the collective decision of all the trees in the ensemble through a majority vote.
- **Support Vector Machines (SVM)** is a type of linear machine learning model used for binary and multi-class classification problems. The model finds a hyperplane in the feature space that maximally separates the different classes. If the data is non-linearly separable, SVM uses kernel functions to transform the data into a higher-dimensional space where it can be linearly separated..
- **K-Nearest Neighbors (KNN):** KNN (K-Nearest Neighbors) is a machine learning model that can classify a data point based on the class of its nearest k-neighbors in the

feature space. It is a non-parametric model and is suitable for both binary and multi-class classification problems. KNN can handle non-linearly separable data, and the selection of k determines the bias-variance trade-off of the model.

Chapter 6:

LEARNING EXPERIENCE

Internships provide a unique learning experience that is different from traditional classroom-based learning. During an internship, I had the opportunity to apply the knowledge and skills I have learned in the classroom to real-world situations and to learn from experienced professionals in your field. The learning experience acquired during an internship can be divided into several categories:

- **Hands-on experience:** Aspiring professionals who are eager to acquire practical skills and knowledge in their respective fields often take on internships. These opportunities provide a platform to engage in relevant projects and tasks, thereby affording one the chance to acquire hands-on experience and develop new skills. The benefits of such an experience are two-fold: it enables one to better comprehend the workings of the industry and explore potential career paths. Take, for instance, an intern in a marketing firm; they may be tasked with creating a social media campaign or carrying out market research. By undertaking these responsibilities, they acquire practical experience in the field of marketing and develop an understanding of the opportunities and challenges that underlie this industry.
- **Professional development:** Internships provide a valuable opportunity to cultivate key professional competencies, including communication, teamwork, and time management. Working in a professional environment fosters an understanding of how to interact with colleagues, supervisors, and clients, as well as how to handle workload management effectively. For instance, an intern might need to participate in client meetings or engage in a team project alongside other interns. These experiences serve to enhance communication and teamwork skills.
- **Industry insights:** Internships offer a unique opportunity to gain invaluable insights into the intricacies of the industry. By collaborating with seasoned professionals, one can learn about emerging trends, optimal methodologies, and possible career trajectories. An intern might be exposed to newfangled technologies or marketing tactics employed in the industry and discover how to incorporate these innovations into their work.
- **Networking:** Participating in internships provides a promising avenue to establish a robust professional network. Through collaboration with industry experts, one can form relationships with colleagues, supervisors, and mentors that can prove advantageous in the long run. Building a professional network can help identify job prospects, establish connections with

potential employers, and procure referrals and endorsements. Networking events and opportunities to shadow senior executives can further facilitate this process.

- Self-awareness: Internships are a great opportunity to gain a deeper understanding of your strengths, weaknesses, and interests. Working in a real-world setting allows you to see how your skills and knowledge can be applied in your industry of choice, and helps you identify the type of work you enjoy and excel at. For instance, you may realize that you have a talent for writing marketing copy or analyzing data.

Overall, internships provide a valuable learning experience that can help you to better comprehend your field of study, develop critical professional skills such as communication and teamwork, and build your professional network. By completing an internship, you will be better prepared to enter the workforce and pursue a successful career in your chosen field

APPENDICES

- Data Structure and Algorithmic Thinking with Python Paperback – 1 January 2015 by Narasimha Karumanchi (Author).
- Head-First Python, 2nd edition Paul Barry (*O'Reilly, 2016*).
- Think Python: How to Think Like a Computer Scientist, 2nd edition Allen B. Downey (*O'Reilly, 2015*).
- *The Hundred-Page Machine Learning Book* by Andriy Burkov.
- *Machine Learning For Absolute Beginners* by Oliver Theobald.
- *Machine Learning for Hackers* by Drew Conway and John Myles White.