**Assignment -1**

1 Define Artificial Intelligence (AI) and provide examples of its applications.

A: Artificial intelligence (AI) is essentially the intelligence exhibited by machines. It refers to computer systems that can mimic human cognitive functions like learning and problem-solving.

**Applications of AI:**

* Social media
* E-commerce
* Data security
* Gaming
* Healthcare
* Education
* Entertainment
* Astronomy
* Virtual assistant like Siri, Alexa use AI to understand your voice commands.
* AI is used to analyse the images like X-rays and MRIs.

2. Differentiate between supervised and unsupervised learning techniques in ML.

**A:** Difference between Supervised and unsupervised learning models:

**Supervised data:**

* It uses labeled data
* Supervised learning model takes direct feedback to check if it is predicting the correct output or not.
* Supervised learning model predicts the output
* Supervised learning need supervision to train the model
* Supervised learning is categorised into classification and regression.

**Unsupervised data:**

* Unsupervised learning algorithm are trained using unlabeled data
* Unsupervised learning algorithm does not take the any feedback
* Unsupervised learning algorithm find the hidden pattern
* Unsupervised learning algorithm does not need any supervision to train the model
* Unsupervised learning algorithm can be classified into clustering and association.

3. What is Python? Discuss its main features and advantages.

**A:** Python is an interpreted high-level programming dialect known for its straightforwardness, flexibility and meaningfulness. It is simple to learn and utilize, object-oriented, interpreted, cross-platform, has an broad standard library, is powerfully written, and supports numerous programming standards.

**Features and advantages of python:**

* **High-Level Language:** Python is closer to human language compared to lower-level languages like machine code. This makes it easier to learn and write, allowing you to focus on the problem you’re solving rather than the intricacies of the language itself.
* **Interpreted Language:** Python code doesn’t need to be compiled into machine code before running. Instead, it’s interpreted line by line at runtime, making development faster and more flexible.
* **Dynamically Typed:** You don’t need to declare the data type of variables in Python (like int or float). This simplifies code and makes it more adaptable.
* **Object-Oriented:** Python supports object-oriented programming, allowing you to structure your code around objects and their interactions. This promotes code reusability and maintainability.
* **Extensive Standard Library:** Python comes with a rich collection of built-in modules and functions for various tasks, reducing the need to write everything from scratch.

**Large Third-Party Library:** Ecosystem: Beyond the standard library, there’s a vast collection of third-party libraries available for nearly any imaginable task, from machine learning (TensorFlow, PyTorch) to web development (Django, Flask) to scientific computing (NumPy, SciPy).

**Organizations using python:**

* Google
* Yahoo
* YouTube
* Spotify
* Quora
* Facebook

4. What are the advantages of using Python as a programming language for AI and ML?

**A:** There are several advantages to using Python for artificial intelligence (AI) and machine learning (ML):

* **Extensive Libraries and Frameworks:** Python has a rich ecosystem of libraries and frameworks specifically designed for AI and ML tasks. These tools provide pre-built functions and modules that make it easier to develop and deploy machine learning models.
* **Easy to Learn and Read:** Python’s syntax is known for its clarity and readability, resembling everyday English. This makes it a beginner-friendly language for those new to programming or AI/ML.
* **Flexibility**: Python is a general-purpose language, meaning it can be used for various tasks beyond AI and ML. This versatility makes it suitable for different stages of an AI/ML project.
* **Rapid Prototyping**: Python’s interpreted nature and simple syntax allow for rapid prototyping. You can quickly write and test code snippets without needing to spend a lot of time on compilation or complex debugging processes.

5. Discuss the importance of indentation in Python code.

**A:** Indentation in Python is not just a stylistic preference; it’s a fundamental aspect of the language’s syntax. Unlike many other programming languages that use curly braces ({}) or other keywords to define code blocks, Python relies solely on indentation to determine the structure of your code. Here’s why indentation is so important**:**

1. Define code blocks
2. Mandatory for functionality
3. Improse readability
4. Enforces code maintainability

6.Define a variable in Python. Provide examples of valid variable names.

**A: Assignment**: The act of creating a variable involves assigning a value to it using the assignment operator (=).

The syntax Is:

Variable\_name = value

**Naming Conventions:** Python follows specific rules for naming variables

Variable names must start with a letter (uppercase or lowercase A-Z) or an underscore (\_).

They can contain letters, numbers, and underscores, but no special characters (except underscore).

Variable names are case-sensitive (age is different from Age).

You cannot use reserved keywords (like if, for, while) as variable names.

**Examples of Valid Variable Names:**

Name = “Alice” # String

Age = 30 # Integer

User\_rating = 4.5 # Float

Is\_admin = True # Boolean

Data\_list = [] # List

My\_function = () # Empty tuple

**Data Types:** The value assigned to a variable determines its data type (eg: integer, string, boolean, list). Python is dynamically typed, meaning you don’t need to explicitly declare the data type beforehand. The data type is inferred based on the assigned value.

7. Explain the difference between a keyword and an identifier in Python.

**A:** **Keywords:**

* + Predefined words: Keywords are reserved words with special meanings within the Python language itself. They cannot be used for anything else.
  + Control the flow: Keywords play a crucial role in defining the program’s structure and logic. They control the flow of execution, like loops (for, while), conditional statements (if, elif, else), function definitions (def), and various other functionalities.
  + Always lowercase: By convention, all keywords in Python are written in lowercase letters.

Examples of keywords in Python:

If, else, for, while, def, return, True, False, None

**Identifiers:**

* User-defined names: Identifiers are names you create to identify different elements in your program. These can be variable names, function names, class names, or module names.
* Flexibility: You have more freedom in choosing names for identifiers compared to keywords. As long as they follow naming conventions, you can create meaningful names that reflect the purpose of the element they represent.
  + Naming conventions: Valid identifiers must start with a letter (uppercase or lowercase) or an underscore (\_). They can contain letters, numbers, and underscores, but no other special characters. Additionally, they must be distinct from keywords.

Examples of valid identifiers in Python:

Message (variable name)

Calculate\_area (function name)

Circle (class name)

Data\_reader (module name)

8. List the basic data types available in Python.

**A:** Python has several built-in data types, including numeric types (int, float, complex), string (str), boolean (bool), and collection types (list, tuple, dict, set). Each data type has its own set of properties, methods, and behaviors that allow programmers to manipulate and process data effectively in their programs.

**The following are the standard or built-in data types in Python:**

* + Numeric
  + Sequence Type
  + Boolean
  + Set
  + Dictionary
  + Binary Types
  + List
  + Tuple
  + Float

9. Describe the syntax for an if statement in Python.

**A:** The syntax for if statement in python:

If condition:

# code to be executed if the condition is True

If: This keyword marks the beginning of the if statement.

Condition: This is an expression that evaluates to True or False. It can be a comparison, a logical operation, or any expression that results in a boolean value.

:: The colon (:) Indicates the end of the first line of the if statement and the start of the code block that will be executed if the condition is True.

Indented code block: The code to be executed if the condition is True must be indented. Indentation is crucial in Python as it defines the scope of the if statement. Only the lines indented at the same level as the line after the colon belong to the if block.

Example:

Age = 25

If age >= 18:

Print(“You are eligible to vote.”)

In this example, the if statement checks if the value of the variable age is greater than or equal to 18.

10. Explain the purpose of the elif statement in Python.

**A:** ‘Elif’ stands for ‘else if’ and is used in Python programming to test multiple conditions. It is written following an if statement in Python to check an alternative condition if the first condition is false. The code block under the elif statement will be executed only if its condition is true.

**Example**:

Grade = 85

If grade >= 90:

Print(“Excellent!”)

Elif grade >= 80:

Print(“Very Good!”)

Elif grade >= 70:

Print(“Good!”)

Else:

Print(“Needs Improvement”)