

## Assignment

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

Artificial intelligence, the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. Since the development of the digital computer in the 1940s, it has been demonstrated that computers can be programmed to carry out very complex tasks, such as discovering proofs for mathematical theorems or playing chess - with great proficiency. Still, despite continuing advances in the computer processing speed and memory capacity, there are as yet no programs that can match full human flexibility over wider domains or in tasks requiring much everyday knowledge.

→ Artificial intelligence has become a crucial part of daily human lives today and it assists in almost every scenario whether you realize it or not.

### Examples

- Automated customer support.
- Personalized shopping experience.
- Health care.
- Finance.
- Smart cars and drones.
- Travel and support.
- Social media.

## 2. Differentiate between Supervised and Unsupervised learning techniques in ML.

### Supervised learning

→ Input data is labeled.

→ Has a feedback mechanism.

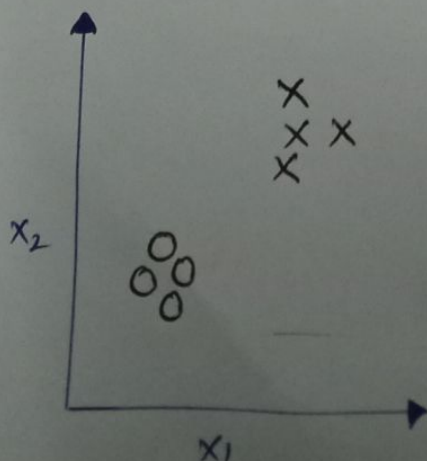
→ Data is classified based on the training dataset.

→ Divided into regression & classification.

→ Used for prediction.

→ Algorithms include: decision trees, logistic regressions, support vector machine.

→ A known number of classes.



### Unsupervised learning

→ Input data is unlabeled.

→ Has no feedback mechanism.

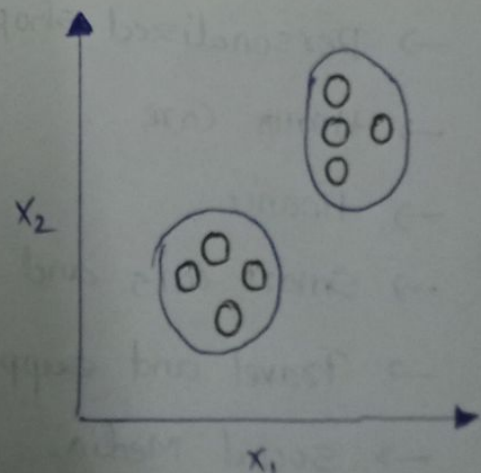
→ Assigns properties of given data to classify it.

→ Dividing into clustering & Association.

→ Used for Analysis.

→ Algorithms include: k-means clustering, hierarchical clustering, apriori algorithm.

→ A unknown number of classes.





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

python is a set of instructions that we give in the form of a programme to our computer to perform any specific task. It is a programming language having properties like it is interpreted, object-oriented and it is high-level too. Due to its beginner-friendly syntax, it became a clear choice for beginners to start their programming journey.

Syntax : # code

Print ("Hello world")

Features : Python has plenty of features that make it the most demanding and popular.

- Easy to read and understand.
- Interpreted language.
- Object-oriented programming language.
- Free and open source.
- Versatile and Extensible.
- Multi-platform.
- Dynamically typed.
- Huge and active community.

Advantages :

- Easy to learn, read and understand.
- Versatile and open-source.
- Improves productivity.
- Supports libraries.
- Strong community.
- Huge library.

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

Python is widely regarded as one of the best programming languages for AI and ML. due to several advantages.

→ Ease of learning and use: Python's syntax is clear, concise, and resembles pseudo-code, making it easy to read and understand.

→ vast Ecosystem of libraries: Python boasts a rich ecosystem of libraries and frameworks specifically tailored for AI and ML tasks.

→ Community Support: Python has a large and active community of developers who contribute to its growth and improvement.

→ Flexibility: Python is a versatile language that supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

→ Interoperability: Python seamlessly integrates with other languages and platforms, allowing developers to leverage existing codebases and infrastructure.

→ Scalability: While Python is often criticized for its performance compared to lower-level languages like C++ or Java, its performance is usually sufficient for most AI and ML tasks.

\* Overall, Python's combination of simplicity, power & Community Support makes it an excellent choice for AI and ML development.



5. Discuss the importance of indentation in python code.

indentation in python is not just a matter of aesthetics; it's a fundamental aspect of language's syntax. Here's why it's so important.

- Readability and clarity: Python emphasizes readability, and indentation plays a crucial role in making code easy to read and understand.
- Enforcement of code structure: Unlike many other programming languages that use braces or keywords to denote code blocks, python uses indentation.
- Consistency: Python's syntax enforces consistency in code formatting.
- Debugging and maintenance: Properly indented code is easier to debug and maintain.
- Style guidelines: Python has PEP 8, the official style guide for python code, which recommends using 4 spaces per indentation level.
- Pythonic code: Writing code that adheres to python's conventions, including proper indentation, is often referred to as writing "pythonic" code.

\* In summary, indentation is not just a matter of style in python; it's a fundamental aspect of language's syntax and plays a crucial role in debugging, code structure, consistency, maintenance, and adhering to python's conventions.

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

In python, a variable is named location in memory used to store data. You can think of it as a label that refers to a particular value.

Syntax: python

variable\_name = value

→ Example of valid variable names in python.

python

age = 25

name = "John"

height\_cm = 180.5

is\_student = True

→ variable names in python can consist of letters, digits and underscores.

\* Variable name cannot begin with a digit.

\* variable names cannot contain spaces or special characters.

Such as !, @, #, \$, %, etc.

\* variable names are case-sensitive, meaning, age, Age and

AGE could be considered different variables.

Examples:

python

my\_variable = 10

Myvariable = 20

My Variable = 30

-my-variable = 40

MY-VARIABLE = 50.



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

→ Keywords: keywords are reserved words in python that have predefined meanings & purposes.

\* These are used to define the structure and logic of python programs, such as control flow statements (if, else, while, for) variable declarations (def, class, import) and so on.

- Examples :

include if, else, for, while, def, class, import, True, False, None, etc.

\* Identifiers: Identifiers are names given to entities in python such as variables, functions, classes, modules etc.

→ They are user-defined and are used to uniquely identify these entities within a program.

→ They must begin with a letter (a-z), (A-Z) or underscore (\_), followed by zero or more letters, digit (0-9), or underscores (\_).

→ They cannot be a python keyword.

→ They are case-sensitive (MyVariable and myvariable are considered different identifiers)

- Examples :

include variable names (age, name, height\_cm),

function names (calculate\_area, print\_result),

class names (person, Employee), etc.

## 8: List the basic data types available in python.

In python, there are several basic data types that are commonly used to represent different kinds of values.

\* Integer (int): Represents whole numbers, both +ve & -ve, without any decimal point.

ex: 10, -5, 1000.

\* Float (float): Represents real numbers.

ex: 3.14, -0.001, 2.71828.

\* String (str): Represents a sequence of characters enclosed within single (') or double (") quotes.

ex: 'hello', "python", '123'.

\* Boolean (bool): Represents one of 2 possible values: True or false. Used in conditional statements and logical operations.

\* None Type: Represents the absence of value or a null value. It is commonly used to simplify that a variable or function returns nothing.

\* List (list): Represents an ordered collection of items, which can be of different types.

ex: [1, 2, 3], ['apple', 'banana', 'cherry'].

\* Tuple (tuple): Similar to lists, tuples are ordered collections of items, but they are immutable, meaning their elements cannot be changed after creation.

ex: (1, 2, 3) ('apple', 'banana', 'cherry').



a. Describe the syntax for an if statement in python.

In python, an if statement is used for conditional execution. It allows you to execute a block of code only if a certain condition is true. Here's the basic syntax for an if statement:

Syntax: python

if condition:

# indented block of code to execute if condition is True

Statement 1

Statement 2

---

- Here's a breakdown of components of if statement

syntax:

\* if: This is the keyword that starts the if statement.

\* condition: This is expression that is evaluated to either True or false.

\* : (colon): The colon is required after the condition to indicate the start of the indented block of code that will be executed if condition is true.

\* indented block of code: This is the block of code that will be executed if the condition is true. It can contain one or more statements.

EX: python

x = 10

if x > 5

print("x is greater than 5")

10. Explain the purpose of `elif` statement in python.

In python, there is no specific "`elif` statement".

Perhaps you meant to refer to the concept of the `pass` Statement, which is used as place holder when a statement is syntactically required but no action is intended or necessary.

Example:

\* python

if condition:

#TODO: Add code here later

Pass.

→ In this example, `pass` statement is used as placeholder within an if statement. Another common use case for `pass` element is when defining empty classes or functions that you intend to implement later.

\* python

class MyClass:

pass

def my-function():

pass

→ In both cases, `pass` is used to indicate that class or function body is empty and will implement later.

→ In summary, the `pass` Statement in python serves the purpose of a placeholder or a no-operation statement, allowing code to pass through without any action when it is encountered.