ASSIGNMENT-1

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

1. Artificial Intelligence (AI) is the theory and development of computer systems capable of performing tasks that typically require human intelligence. These tasks include recognizing speech, making decisions, identifying patterns, and solving problems. AI encompasses a wide range of technologies, such as machine learning, deep learning, and natural language processing (NLP).

   Here are some examples of AI applications across various sectors:

   - **E-commerce**: AI helps in making personalized product recommendations based on user search history and preferences.
   - **Smart Homes:** AI is used in smart assistants like Amazon Alexa or Google Home to control home devices
   - **Finance**: AI is utilized for automated trading, fraud detection, and financial planning.
   - **Robotics**: Manufacturing robots and humanoid social robots are designed to perform tasks and interact with humans.
   - **Healthcare**: AI is used for medical diagnosis, predicting patient outcomes, and personalizing treatment plans.

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

2. In machine learning (ML), supervised and unsupervised learning are two core **Differentiate** techniques, each with its distinct approach to modeling and analyzing data:
3.
4. **Supervised Learning:**
5.
6. <u>Definition:</u> Supervised learning involves training a model on a labeled dataset, which means that each training example is paired with an output label.

7. Goal: The goal is to learn a mapping from inputs to outputs, making predictions based on that learned mapping.
8. Examples: Classification and regression tasks, where the model predicts a discrete label or a continuous output, respectively.
9. Process: It uses patterns to predict the values of the label on additional unlabeled data. Supervised learning is akin to a student learning under the supervision of a teacher.
10. Algorithms: Common algorithms include linear regression, logistic regression, support vector machines, neural networks, and decision trees.
11.
12. **Unsupervised Learning:**
13.
14. Definition: Unsupervised learning, on the other hand, deals with unlabeled data. The system tries to learn the patterns and the structure from the data without any external guidance.
15. Goal: The primary goal is to model the underlying structure or distribution in the data to learn more about the data.
16. Examples: Clustering and association tasks, where the model seeks to group or associate data points based on similarities or frequent patterns.
17. Process: It's like a student trying to learn without a teacher; the patterns emerge naturally from the data.
18. Algorithms: Common algorithms include k-means clustering, hierarchical clustering, and Apriori algorithm.

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

Python is a high-level, interpreted programming language known for its easy-to-read syntax and versatility. It was created by Guido van Rossum and first released in 1991. Here are some of its main features and advantages:

Main Features:

Easy to Read and Write: Python's syntax closely resembles the English language, which makes it accessible to beginners.
Interpreted Language: Python code is executed line by line, which makes debugging easier.

Dynamic Typing: Variables in Python can change type dynamically during execution, which adds flexibility.
Extensive Libraries: Python has a vast standard library and a large ecosystem of third-party packages.
Object-Oriented: Python supports object-oriented programming (OOP) paradigms, allowing for code reuse and modularity.
Portable: Python code can run on various operating systems without requiring changes.

Advantages:
Productivity: Python's simplicity allows developers to focus on solving problems rather than syntax intricacies.
Community Support: A large community contributes to a wealth of tutorials, documentation, and forums.
Versatility: Python is used in web development, data analysis, artificial intelligence, scientific computing, and more.
Integration: Python can be integrated with other languages and technologies, and it supports various protocols and formats.

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

Advantages:

- Less code
- Access to great libraries and frameworks for AI and machine learning (ML)
- Platform independence
- Flexibility
- Simplicity and consistency

5. **Discuss the importance of indentation in Python code.**

Indentation in Python code is not only a matter of style but also a syntactical requirement. Here's why it's important:

Structure: Indentation is used to define the structure of the code. In Python, blocks of code are defined by their indentation level, which replaces the braces {} used in many other languages to define scope.
Readability: Proper indentation makes the code more readable and understandable. It clearly delineates different sections and levels of logic within the code.
Flow Control: Indentation is crucial for control statements like if, for, while, and function definitions. The code within these constructs must be indented to indicate that it's part of the control flow.
Error Prevention: Incorrect indentation can lead to IndentationError, which will stop the code from running. It ensures that the code executes as intended.
Community Standards: The Python community has adopted PEP 8 – Style Guide for Python Code, which includes guidelines for indentation. Following these standards helps maintain consistency across different Python projects.

```
def greet(name):
    if name:
        print(f"Hello, {name}!")
    else:
        print("Hello, World!")

greet("Alice")
```

In this function, the if statement and the print functions are indented to show they are within the greet function's scope. The indentation indicates that the print statements are part of the conditional logic.

Remember, consistency is key. Whether you use spaces or tabs, sticking to one method throughout your code is essential for avoiding errors and maintaining readability.

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

In Python, a variable is a name that refers to a memory location where you can store data values. It acts as a container for data in your code. To define a variable, you simply assign a value to a variable name using the equals sign =.

Here are some examples of valid variable names in Python:

```
# Examples of valid variable names
my_variable = 10
variable2 = "Hello"
user_id = 501
max_speed = 120.5
_is_valid = True
```

Rules for valid Python variable names:

Must start with a letter (a-z, A-Z) or an underscore (_).

Can be followed by letters, numbers (0-9), or underscores.

Cannot start with a number.

Are case-sensitive (myVariable and myvariable are different variables).

Cannot be a reserved word in Python (like if, else, import, etc.).
Remember, meaningful variable names make your code more readable and maintainable.

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

In Python, a keyword and an identifier serve different purposes:

Keywords:

Keywords are the reserved words in Python that have a special meaning to the interpreter.
They cannot be used as identifiers for other elements like variables, functions, classes, etc.
Examples include def, return, if, else, class, and while.
Python has a set list of keywords that cannot be changed or modified.

Identifiers:

Identifiers are the names you assign to entities like variables, functions, classes, etc., to identify them.
They can be chosen by the programmer but must follow certain rules, such as:
They must start with a letter (a to z or A to Z) or an underscore (_).
After the first character, they can contain letters, underscores, or digits (0 to 9).
They cannot be the same as a keyword.
They are case-sensitive (myVar, myvar, and MYVAR are different identifiers).
Examples of identifiers include user_age, total_sum, CarModel, etc.

Here’s a simple illustration in Python code:

```
# 'def' is a keyword used to define a function.
def function_name(parameter):
    # 'return' is a keyword that returns a value from a function.
    return parameter

# 'function_name' and 'parameter' are identifiers representing the function and its argument.
```

choosing meaningful and descriptive identifiers makes your code more readable and maintainable, while respecting the rules and avoiding the use of keywords as identifiers.

8. List the basic data types available in Python.

In Python, the basic data types are:

int: for integers like 1, 42, -10.

float: for floating-point numbers like 3.14, -0.001, 2.0e8.

str: for strings, text enclosed in quotes like 'hello', "world".

bool: for Boolean values True and False.

list: for ordered sequences of values like [1, 2, 3], ['a', 'b', 'c'].

tuple: for ordered, immutable sequences like (1, 2, 3), ('a', 'b', 'c').

dict: for key-value pairs like {'key1': 'value1', 'key2': 'value2'}.

set: for unordered collections of unique elements like {1, 2, 3}.

These are the core data types that you’ll frequently use in Python programming. There are also other types like complex for complex numbers and NoneType for the None object, which represents the absence of a value.

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

In Python, an `if` statement is used to test a condition and execute a block of code if the condition is true. Here's the basic syntax:

```python
if condition:
    # code to execute if the condition is true
```

Here's an example with an actual condition:

```python
if x > 10:
    print("x is greater than 10")
```

In this example, if the variable `x` is greater than 10, the message "x is greater than 10" will be printed to the console. Remember that Python relies on indentation to define the scope of the code blocks, so make sure to indent the code under the `if` statement properly.

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

The purpose of the `elif` statement in Python is to:

- Handle multiple conditions sequentially.
- Execute a specific block of code as soon as a true condition is found.
- Serve as a shortened version of "else if".

BY:
RUDRAVENA SATHWIKA
21UK1A0527