Alliance School of Advanced Computing

Department of Computer Science and Engineering

Class Assignment-1

Course Code: 5CS1025

Course Title: Artificial intelligence Semester: 04

Class : AIML

Name: P.Yashwanth kumar

RegNo:2023BCSE07AED369

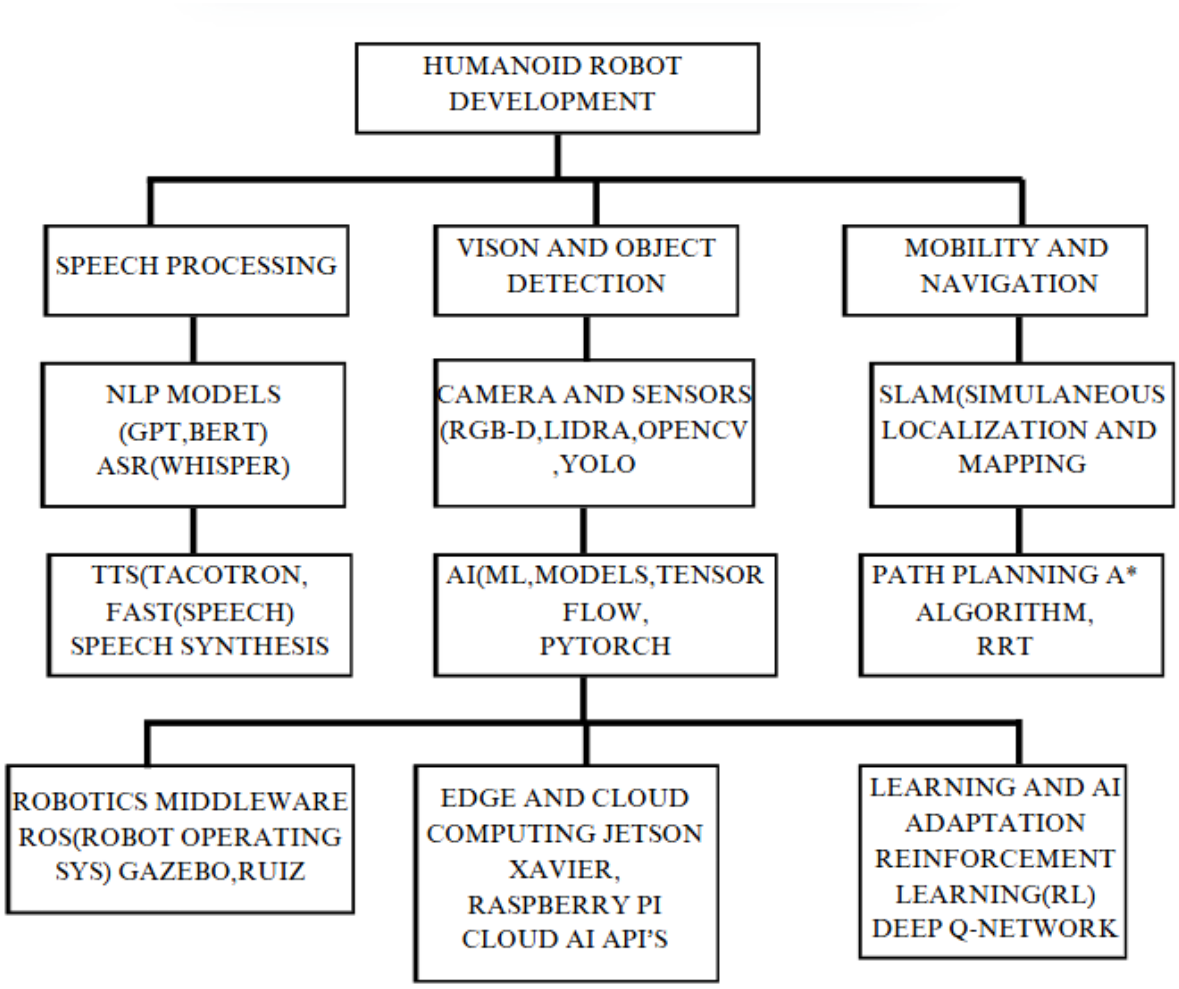
2024-25

AI- ASSIGNMENT-1

Github-Link

https://github.com/siva123h/siva123h/blob/3d215a4ae05e6fbefed83da16b1c9be3885c862b/siva.assignment-1AI.pdf

1.Imagine you are tasked with designing a humanoid robot to assist in a home or office environment. The robot must be capable of interacting with people by talking and listening, walking to different locations, seeing and recognizing objects, and learning from its surroundings to adapt its behaviour . What technologies, tools, and frameworks would you need to build such a robot? Give as flow chart.



2.Calculate and interpret mean, median, mode, variance and standard deviation for a given dataset. Data =[15,21,29,21,15,24,32,21,15,30].

```
import pandas as pd
import numpy as np
data = [15, 21, 29, 21, 15, 24, 32, 21, 15, 30]
df = pd.DataFrame(data, columns=["Values"])
print("Mean of data:",df["Values"].mean())
print("Median of data:",df["Values"].median())
print("Mode of data:",df["Values"].mode())
print("Standard deviation of data:",df["Values"].std())
print("Variance of data:",df["Values"].var())
```

```
Mean of data: 22.3
Median of data: 21.0
Mode of data: 0    15
1    21
Name: Values, dtype: int64
Standard deviation of data: 6.3779132776934
Variance of data: 40.67777777777778
```

3. You are analyzing a dataset that captures the daily performance and activity of a humanoid robot in a simulated environment. The dataset link robot_dataset(robot_dataset)_1.csv includes the following attributes.

Interaction_Count: Number of conversations the robot had daily.
Steps_Walked: Total steps taken each day.
Objects_Recognized: Number of objects successfully identified by the robot.
Learning_Sessions: Number of learning tasks completed.
Energy_Consumption (kWh): Daily energy usage of robots.

Perform Basic Statistical Operations:

1) What is the average (mean) number of conversations the robot has daily?
2) Find the total steps walked by the robot over a given period.
3) Determine the maximum and minimum energy consumption in the dataset.
4) Calculate the correlation between the number of steps walked and energy consumption.
5) Analyze the distribution of objects recognized daily (e.g., histogram or box plot).
6) What is the variance in the number of learning sessions completed?

```python
import numpy as np
import pandas as pd
df = pd.read_csv("robot_dataset(robot_dataset)_1.csv")
print("Mean Accuracy:", df["Accuracy (%)"].mean())
print("Total Robots:", df["Robot_ID"].count())
print(df[["Robot_ID", "Task_Type"]])
print(df.iloc[10:20])
print(df.groupby(['Sensor_Type']).mean(numeric_only=True))
grouping = df.groupby(['Task_Type','Environmental_Status'])
print(grouping.first())
```

```
Mean Accuracy: 94.92060000000001
Total Robots: 500
    Robot_ID   Task_Type
0    RBT_001  Inspection
1    RBT_002    Assembly
2    RBT_003  Inspection
3    RBT_004     Welding
4    RBT_005    Assembly
..       ...         ...
495  RBT_496  Inspection
496  RBT_497  Inspection
497  RBT_498  Inspection
498  RBT_499    Assembly
499  RBT_500    Assembly

[500 rows x 2 columns]
   Robot_ID   Task_Type Component_ID     Sensor_Type            Sensor_Data  \
10  RBT_011    Assembly      CMP_567          Camera         0 (no obstacle)
11  RBT_012    Painting      CMP_268           LIDAR              75.8 (°C)
12  RBT_013  Inspection      CMP_631         Thermal       98% (defect-free)
13  RBT_014  Inspection      CMP_115          Camera              82.4 (°C)
14  RBT_015  Inspection      CMP_932  LIDAR + Camera        92% (visual fit)
```

4.Write a Python program that declares variables of different data types (e.g., string, integer, float,and boolean). Output the variables in a sentence format using print() and f-strings.

```python
name = "siva"
age = 19
height = 4.6
is_student = True

# Output the variables in a sentence format using f-strings
print(f"My name is {name}, I am {age} years old,")
print(f"my height is {height} feet.")
print(f"Am I a student? {is_student}.")
```

```
My name is siva, I am 19 years old,
my height is 4.6 feet.
Am I a student? True.
```

5. Write a Python program that takes an integer input and checks whether the number is positive, negative, or zero using conditional statements (if-else).

```
[7]: num = int(input("Enter an integer: "))
     if num > 0:
         print("The number is positive.")
     elif num < 0:
         print("The number is negative.")
     else:
         print("The number is zero.")
```

```
Enter an integer:  281
The number is positive.
```

6. Write a Python program that takes a number as input and prints the multiplication table for that number (from 1 to 10).

```
[5]: n=int(input("enter a number"))
     print("multiplication table:")
     for i in range(1,11):
          print(n,"x",i,"=",i*n)
```

```
enter a number 8
multiplication table:
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80
```

```
[ ]:
```

7. Create a Python list that contains the names of 5 different fruits. Perform the given operations on the list.

```
fruits = ["Apple", "Banana", "Cherry", "Date", "Elderberry"]
fruits.append("Cherry")
print("after adding Cherry:", fruits)
fruits.remove("Date")
print("after removing Date:", fruits)
fruits.sort()
print("After sorting :", fruits)
fruits.reverse()
print("After reversing:", fruits)
index_Cherry = fruits.index("Cherry")
print("Index of Cherry:", index_Cherry)
third_fruit = fruits[2]
print("The third fruit in the list:", third_fruit)
sliced_fruits = fruits[1:4]
print("Sliced list (index 1 to 3):", sliced_fruits)
```

```
after adding Cherry: ['Apple', 'Banana', 'Cherry', 'Date', 'Elderberry', 'Cherry']
after removing Date: ['Apple', 'Banana', 'Cherry', 'Elderberry', 'Cherry']
After sorting : ['Apple', 'Banana', 'Cherry', 'Cherry', 'Elderberry']
After reversing: ['Elderberry', 'Cherry', 'Cherry', 'Banana', 'Apple']
Index of Cherry: 1
The third fruit in the list: Cherry
Sliced list (index 1 to 3): ['Cherry', 'Cherry', 'Banana']
```

8.Write a Python program that creates a tuple containing 5 numbers. Perform the given operations on the tuple.

```
[10]: numbers = (10, 20, 30, 40, 50)
      print("third element:", numbers[2])
      print("Sliced tuple (1 to 3):", numbers[1:4])
      print("Index of30:",numbers.index(30))
      print("Count of 50:", numbers.count(50))
      print("Sum:", sum(numbers))
      print("Max:", max(numbers), "Min:", min(numbers))
```

```
third element: 30
Sliced tuple (1 to 3): (20, 30, 40)
Index of30: 2
Count of 50: 1
Sum: 150
Max: 50 Min: 10
```

9.Create a dictionary that stores the names of 3 students as keys and their marks in mathematics as values. Perform the given operations.

```
[12]: students ={"vivek": 90, "uday": 97, "bhanu": 80}
      students["vivek"] = 90
      students["uday"] = 97
      students.pop("bhanu")

      print("vivek's marks:", students["vivek"])
      print("Final Marks:", students)
```

```
vivek's marks: 90
Final Marks: {'vivek': 90, 'uday': 97}
```

10. Create two sets of integers. Perform the given set operations.

```
[14]: set1 = {9, 7, 2, 5, 1}
      set2 = {8, 1, 7, 3,4}
      print("Union:",set1 | set2)
      print("Intersection:", set1 & set2)
      print("difference(set1 - set2):", set1 - set2)
      print("symmetric difference:", set1 ^ set2)
```

```
Union: {1, 2, 3, 4, 5, 7, 8, 9}
Intersection: {1, 7}
difference(set1 - set2): {9, 2, 5}
symmetric difference: {2, 3, 4, 5, 8, 9}
```

11. Write a Python function called find_largest() that takes a list of numbers as input and returns the largest number from the list. Test the function with a sample list.

```
[16]: def find_largest(numbers):
          return max(numbers)
      list = [99, 67, 35, 46, 77, 85]
      largest_number = find_largest(list)
      print("The largest number in the list is:", largest_number)
```

```
The largest number in the list is: 99
```

12. Use list comprehension to create a list of squares of all even numbers between 1 and 20.

```
[17]: squares_of_even_numbers = [x**2 for x in range(1, 21) if x % 2 == 0]
      print(squares_of_even_numbers)

      [4, 16, 36, 64, 100, 144, 196, 256, 324, 400]
```

13. Write a Python script that uses a lambda function to calculate the product of two numbers provided by the user.

```
[15]: num1 = int(input("Enter the first number: "))
      num2 = int(input("Enter the second number: "))
      product = (lambda x, y: x * y)(num1, num2)
      print(f"The product of {num1} and {num2} is: {product}")

      Enter the first number:  7
      Enter the second number:  9
      The product of 7 and 9 is: 63

[ ]:
```

14. Write a Python program to create a one-dimensional, two-dimensional, and three-dimensional NumPy array. Print the shape and dimensions of each array.

```
import numpy as np
arr_1d = np.array([1, 2, 3, 4, 5])
print("One-dimensional array:")
print("Array:", arr_1d)
print("Shape:", arr_1d.shape)
print("Dimensions:", arr_1d.ndim)
print()
arr_2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print("Two-dimensional array:")
print("Array:", arr_2d)
print("Shape:", arr_2d.shape)
print("Dimensions:", arr_2d.ndim)
print()
arr_3d = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
print("Three-dimensional array:")
print("Array:", arr_3d)
print("Shape:", arr_3d.shape)
print("Dimensions:", arr_3d.ndim)
```

```
One-dimensional array:
Array: [1 2 3 4 5]
Shape: (5,)
Dimensions: 1

Two-dimensional array:
Array: [[1 2 3]
 [4 5 6]
 [7 8 9]]
Shape: (3, 3)
Dimensions: 2

Three-dimensional array:
Array: [[[1 2]
  [3 4]]

 [[5 6]
  [7 8]]]
Shape: (2, 2, 2)
Dimensions: 3
```

15. Write a Python program to create a 5x5 NumPy array of random integers and Perform array indexing as given.

[17]:
```
import numpy as np
array = np.random.randint(50, 201, (5, 5))
print("5x5 Array of Random Integers:")
print(array)
print("\nElement at row 2, column 3:", array[1, 2])
print("\nSubarray (rows 2-3, columns 1-2):")
print(array[1:3, 0:2])
print("\nRow 4:")
print(array[3])
print("\nColumn 3:")
print(array[:, 2])
```

```
5x5 Array of Random Integers:
[[135  93 180 194 178]
 [106  64 162  67  51]
 [193 110 159  57 111]
 [196 194 150 122  63]
 [ 94 186  99 140 158]]

Element at row 2, column 3: 162

Subarray (rows 2-3, columns 1-2):
[[106  64]
 [193 110]]

Row 4:
[196 194 150 122  63]

Column 3:
[180 162 159 150  99]
```

https://github.com/siva123h/siva123h/blob/3d215a4ae05e6fbefed83da16b1c9be3885c862b/siva.assignment-1AI.pdf

16. create a NumPy array of shape (4, 4) containing numbers from 1 to 16. Use slicing to extract for the given conditions.

```
import numpy as np
arr = np.arange(1, 17).reshape(4, 4)
print("4x4 Array:\n", arr)
print("\nFirst two rows:\n", arr[:2])
print("\nLast two columns:\n", arr[:, -2:])
print("\nDiagonal elements:", arr.diagonal())
print("\nSubarray (2nd and 3rd rows, 2nd and 3rd columns):\n", arr[1:3, 1:3])
```

```
4x4 Array:
 [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]

First two rows:
 [[1 2 3 4]
 [5 6 7 8]]

Last two columns:
 [[ 3  4]
 [ 7  8]
 [11 12]
 [15 16]]

 [ 1  6 11 16]nts:

Subarray (2nd and 3rd rows, 2nd and 3rd columns):
 [[ 6  7]
 [10 11]]
```

17. Write a Python program that creates a 2D array of shape (6, 2) using np.arange() and then reshapes it into a 3D array of shape (2, 3, 2). Flatten the reshaped array and print the result.

```
import numpy as np
arr_2d = np.arange(1, 13).reshape(6, 2)
print("2D Array (6, 2):\n", arr_2d)
arr_3d = arr_2d.reshape(2, 3, 2)
print("\nReshaped 3D Array (2, 3, 2):\n", arr_3d)
arr_flattened = arr_3d.flatten()
print("\nFlattened Array:\n", arr_flattened)
```

```
2D Array (6, 2):
 [[ 1  2]
 [ 3  4]
 [ 5  6]
 [ 7  8]
 [ 9 10]
 [11 12]]

Reshaped 3D Array (2, 3, 2):
 [[[ 1  2]
  [ 3  4]
  [ 5  6]]

 [[ 7  8]
  [ 9 10]
  [11 12]]]

Flattened Array:
 [ 1  2  3  4  5  6  7  8  9 10 11 12]
```

18. Write a Python program to demonstrate broadcasting. Create an array of shape (3, 3) and add a one-dimensional array of shape (1, 3) to it using broadcasting.

```
]: import numpy as np

# Create a 2D array of shape (3, 3)
array_2d = np.array([[1, 2, 3],[4, 5, 6],[7, 8, 9]])
array_1d = np.array([10, 20, 30])
result = array_2d + array_1d

print("\nResult after broadcasting:\n",result)
```

```
Result after broadcasting:
 [[11 22 33]
 [14 25 36]
 [17 28 39]]
```

19. Create two NumPy arrays of the same shape, A and B. Perform the following arithmetic operations:
Element-wise addition.
Element-wise subtraction.
Element-wise multiplication.
Element-wise division

```
import numpy as np
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
B = np.array([[9, 8, 7], [6, 5, 4], [3, 2, 1]])
print("Addition:\n", A + B)
print("\nSubtraction:\n", A - B)
print("\nMultiplication:\n", A * B)
print("\nDivision:\n", A / B)
```

```
Addition:
 [[10 10 10]
 [10 10 10]
 [10 10 10]]

Subtraction:
 [[-8 -6 -4]
 [-2  0  2]
 [ 4  6  8]]

Multiplication:
 [[ 9 16 21]
 [24 25 24]
 [21 16  9]]

Division:
 [[0.11111111 0.25       0.42857143]
 [0.66666667 1.         1.5       ]
 [2.33333333 4.         9.        ]]
```

20. Create a Pandas DataFrame with the given Name and marks of 3 courses:
Add a new column named 'Total' that represents the sum of all the courses. Add 'Grade' based on the values of the 'Total'. Print the updated DataFrame with the new 'Total' and 'Grade' column.

[22]:
```
import pandas as pd
data ={'Name': ['vivek', 'siva', 'uday'],
    'maths': [85, 92, 78],
    'social': [88, 79, 94],
    'AI': [91, 85, 80]}
df = pd.DataFrame(data)
df['Total'] = df['maths'] + df['social'] + df['AI']
df['Grade'] = df['Total'].apply(lambda x:'A' if x >= 240 else 'B' if x >= 180 else 'C' if x >= 120 else 'd')
print(df)
```

```
   Name  maths  social  AI  Total Grade
0  vivek     85      88  91    264     A
1   siva     92      79  85    256     A
2   uday     78      94  80    252     A
```