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-E

Name: Sushmitha.S

Roll no: 2023BCSE07AED415

Link: https://github.com/Sushmitha2375/AI-assignment.git

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

Technology	**Tool**	**Framework**
Speech Recognition	Google Speech API	CMU Sphinx
Natural Language Processing	NLTK, Rasa	SpaCy
Computer Vision	OpenCV, RealSense	TensorFlow
Machine Learning	Scikit-learn	PyTorch
Robotics Control	ROS, Arduino	Gazebo
Navigation & Mapping	LIDAR, GPS	SLAM
Object Recognition	YOLO, SSD	OpenCV
Human-Robot Interaction	Unity3D	ROS Interaction Libraries
Emotion Recognition	Affect Net Dataset	TensorFlow
Mobility Systems	Servo Motors, Stepper Motors	Arduino

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]

```python
#2
import pandas as pd
df=[15,21,29,21,15,24,32,21,15,30]
df_series=pd.Series(df)
a=df_series.mean()
print("Mean: ",a,"\n")
b=df_series.median()
print("Median: ",b,"\n")
c=df_series.mode()[0]
print("Mode: ",c,"\n")
d=df_series.var()
print("Variance: ",d,"\n")
e=df_series.std()
print("Standard deviation: ",a)
```

```
Mean:  22.3

Median:  21.0

Mode:  15

Variance:  40.67777777777778

Standard deviation:  22.3
```

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?

```
#3
import pandas as pd
df=pd.read_csv("robot_dataset(robot_dataset)_1(in).csv")
df
```

	Robot_ID	Task_Type	Component_ID	Sensor_Type	Sensor_Data	Processing_Time (s)	Acc
0	RBT_001	Inspection	CMP_460	LIDAR	1 (obstacle detected)	67.0	
1	RBT_002	Assembly	CMP_252	Thermal	85.3 (°C)	71.2	
2	RBT_003	Inspection	CMP_248	Thermal	92% (visual fit)	49.2	
3	RBT_004	Welding	CMP_433	Camera	98% (defect-free)	74.5	
4	RBT_005	Assembly	CMP_992	Camera	92% (visual fit)	64.5	
...	...	...	...	...	...	...	
495	RBT_496	Inspection	CMP_834	LIDAR	85.3 (°C)	66.3	
496	RBT_497	Inspection	CMP_851	LIDAR	92% (visual fit)	45.1	
497	RBT_498	Inspection	CMP_657	LIDAR	82.4 (°C)	75.4	
498	RBT_499	Assembly	CMP_562	Camera	98% (defect-free)	48.7	
499	RBT_500	Assembly	CMP_465	LIDAR + Camera	75.8 (°C)	73.5	

500 rows × 17 columns

```
#3
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 17 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Robot_ID                    500 non-null    object
 1   Task_Type                   500 non-null    object
 2   Component_ID                500 non-null    object
 3   Sensor_Type                 500 non-null    object
 4   Sensor_Data                 500 non-null    object
 5   Processing_Time (s)         500 non-null    float64
```

```
df.describe
```

```
499  RBT_500    Assembly      CMP_465  LIDAR + Camera            75.8 (°C)

     Processing_Time (s)  Accuracy (%) Environmental_Status  \
0                   67.0          90.4              Stable
1                   71.2          98.1              Stable
2                   49.2          95.3            Unstable
3                   74.5          90.2              Stable
4                   64.5          97.2            Unstable
..                   ...           ...                 ...
495                 66.3          96.2            Unstable
```

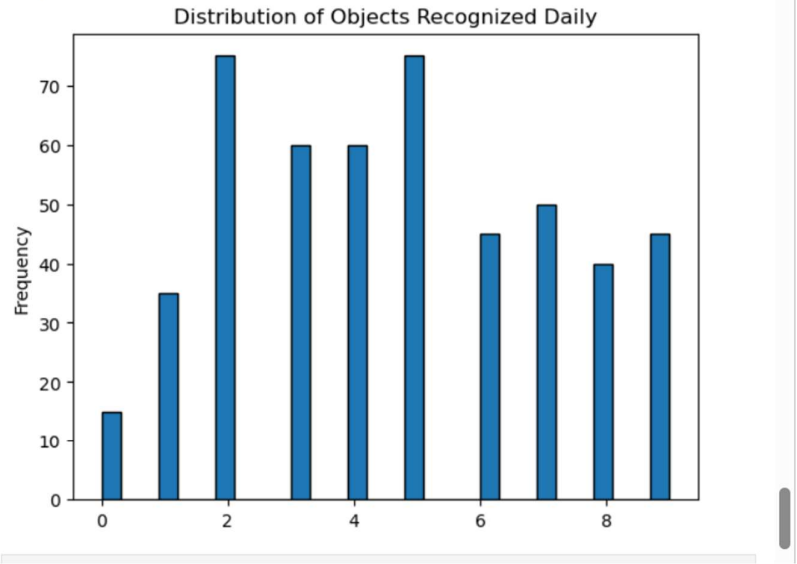
```
correlation = df['Steps_Walked'].corr(df['Energy_Consumption (kwh)'])
print("Correlation between steps and energy consumption:", correlation)
```

Output:

Correlation between steps and energy consumption: 0.001547813739331415

```
[17]: df['Objects_Recognized'].plot(kind='hist', bins=30, edgecolor='black', title='Distr
```

```
[17]: <Axes: title={'center': 'Distribution of Objects Recognized Daily'}, ylabel='Frequ
ency'>
```




```
variance = df['Learning_Sessions'].var()
print("Variance is the number of learning sessions:",variance)
```

```
Variance is the number of learning sessions: 391.9422845691385
```

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
#4
city = "Bangalore"
population = 8436675
temperature = 29.5
raining = False
print(f"I live in {city}, which has a population of {population}.")
print(f"Today's temperature is {temperature}°C.")
```

```
I live in Bangalore, which has a population of 8436675.
Today's temperature is 29.5°C.
```

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).

```python
#5
num = int(input("Enter an integer number: "))
if num < 0:
    print("The number is negative.")
elif num > 0:
    print("The number is positive.")
else:
    print("The number is zero.")
```

```
Enter an integer number:  4
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).

[4]:
```python
#6
num=int(input("Enter number:"))
for i in range(1,10,1):
    print(i*num)
```

```
Enter number: 3
3
6
9
12
15
18
21
24
27
```

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

[6]:
```
#7
fruits=["apple","mango","papaya","orange","pineapple"]
fruits.append("strawberry")
print(fruits)
fruits.reverse()
print(fruits)
fruits.insert(5,"kiwi")
print(fruits)
fruits.pop()
print(fruits)
fruits.clear()
print(fruits)
```

```
['apple', 'mango', 'papaya', 'orange', 'pineapple', 'strawberry']
['strawberry', 'pineapple', 'orange', 'papaya', 'mango', 'apple']
['strawberry', 'pineapple', 'orange', 'papaya', 'mango', 'kiwi', 'apple']
['strawberry', 'pineapple', 'orange', 'papaya', 'mango', 'kiwi']
[]
```

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

```
#8
t=(1,2,3,4,5)
print(t)
print(t[-1])
print(t[1:3])
print(t.count(1))
```

```
(1, 2, 3, 4, 5)
5
(2, 3)
1
```

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

[18]:
```
#9
stu_marks = {"anu": 80, "david": 78, "Charles": 92}
print("Students and their marks:", stu_marks)
print("Marks of anu:", stu_marks["anu"])
stu_marks["alice"] = 88
print("After adding alice:", stu_marks)
del stu_marks["david"]
print("After removing david:", stu_marks)
```

```
Students and their marks: {'anu': 80, 'david': 78, 'Charles': 92}
Marks of anu: 80
After adding alice: {'anu': 80, 'david': 78, 'Charles': 92, 'alice': 88}
After removing david: {'anu': 80, 'Charles': 92, 'alice': 88}
```

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

```
#10
set1={1,2,3,4}
set2={7,5,6,8,2,4}
print("Union",set1.union(set2))
print("Intersection",set2.intersection(set1))
print("symmetric difference",set1.symmetric_difference(set2))
```

```
Union {1, 2, 3, 4, 5, 6, 7, 8}
Intersection {2, 4}
symmetric difference {1, 3, 5, 6, 7, 8}
```

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.

```
#11
def find_largest(number):
    return max(number)
list=[45,11,28,54,96,68]
largest=find_largest(list)
print("Largest number is:",largest)
```

```
Largest number is: 96
```

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

```
#12
square=[i**2 for i in range(1, 21) if i%2==0]
print(square)
```

```
[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.

```
#13
num1=int(input("enter number1:"))
num2=int(input("enter number2:"))
product=(lambda a,b:a*b)(num1,num2)
print("Answer:",product)
```

```
enter number1: 34
enter number2: 21
Answer: 714
```

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.

```
#14
import numpy as np
one=np.array([1,2,3,4])
print(one)
print("shape:",one.shape)
print("dimension:",one.ndim)
two=np.array([[1,2,3,4],[2,4,6,8]])
print(two)
print("shape:",two.shape)
print("dimension:",two.ndim)
three=np.array([[[2,3,4,5],[3,5,7,8],[1,3,5,6,]]])
print(three)
print("shape:",three.shape)
print("dimension:",three.ndim)
```

```
snape: (4,)
dimension: 1
[[1 2 3 4]
 [2 4 6 8]]
shape: (2, 4)
dimension: 2
[[[2 3 4 5]
  [3 5 7 8]
  [1 3 5 6]]]
shape: (1, 3, 4)
dimension: 3
```

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

```
#15
import numpy as np
np.random.random((5,5))
```

```
array([[0.22144473, 0.66967354, 0.35202056, 0.41736461, 0.43927234],
       [0.39118427, 0.29598921, 0.60471064, 0.96722964, 0.32642538],
       [0.28043237, 0.32181735, 0.450185  , 0.03711261, 0.59843511],
       [0.90557179, 0.3615549 , 0.52777284, 0.20371533, 0.36446674],
       [0.38604645, 0.93385879, 0.13488197, 0.5806696 , 0.71159399]])
```

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

```
#16
import numpy as np
arr=np.arange(1, 17).reshape((4,4))
print(arr)
arr1=arr[1:3,2:4]
print(arr1)
```

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

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.

[7]:
```
#17
twod=np.arange(12).reshape(6,2)
print(twod)
thred=twod.reshape(2,3,2)
print(thred)
three=thred.flatten()
print(three)
```

```
[[ 0  1]
 [ 2  3]
 [ 4  5]
 [ 6  7]
 [ 8  9]
 [10 11]]
[[[ 0  1]
  [ 2  3]
  [ 4  5]]

 [[ 6  7]
```

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.

```
#18
import numpy as np
arroned=np.array([1,2,3])
arrtwod=np.array([[[4,6,8],[3,5,6],[5,6,7]]])
relt=arroned+arrtwod
print(relt)
```

```
[[[ 5  8 11]
  [ 4  7  9]
  [ 6  8 10]]]
```

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.

```
#19
a=np.array([3,4,8])
b=np.array([2,7,6])
addition=a+b
subtraction=a-b
multiplication=a*b
division=a/b
print(addition)
print(multiplication)
print(division)
print(subtraction)
```

```
[ 5 11 14]
[ 6 28 48]
[1.5        0.57142857 1.33333333]
[ 1 -3  2]
```

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.

```
#20
import pandas as pd
data={'Name':["david","charles","anu"],
      'course1':[88,90,92],
      'course2':[78,99,90],
      'course3':[99,100,85]}
df=pd.DataFrame(data)
df['Total']=df[['course1','course2','course3']].sum(axis=1)
df['Grade']=df['Total'].apply(lambda a:'A' if a>=270 else ('B' if a>=240 else 'c'))
print(df)
```

```
      Name  course1  course2  course3  Total Grade
0    david       88       78       99    265     B
1  charles       90       99      100    289     A
2      anu       92       90       85    267     B
```