National University of Computer and Emerging Sciences,
Lahore Campus

	Course:	**AI Lab**	**Course Code:**	**CL1002**
	Program:	**BS (CS, DS)**	**Semester:**	**Spring 2023**
	Duration:	**2 Hours 30 Minutes**	**Total Marks:**	**40**
	Paper Date:	**6th June, 2023**	**Weight:**	**40%**
	Section:	**All Sections**	**Page(s):**	**3**
	Exam:	**Lab Final**	**Reg. No.**	

Read below Instructions Carefully:

- Understanding the question statement is also part of the exam, so do not ask for any clarification. In case of any ambiguity, make suitable assumptions.
- You have to complete exam in specified time. No extra time will be given for submission.
- Submit a single **.ipynb file** for all questions.
- Submit folder on **cactus** by following path: \\cactus1\Xeon\Spring 2023\Junaid Hussain\AI Lab Final Slot 1
- Your code should be **intended** and **commented** properly. Use **meaningful variable names**.
- It is your responsibility to save your code from being copied. All matching codes will be considered cheating cases. **PLAGIARISM** will result in forwarding of **case to Disciplinary Committee**.

Q No. 1: Backpropagation or Feed Froward implementation (15 marks)

Implement a feedforward neural network using Python. Your network should have one input layer with 3 nodes, one hidden layer with 4 nodes, and one output layer with 2 nodes. The activation function for all layers is the sigmoid function. Write a function called feedforward that takes an input vector and returns the output vector of the neural network.

Your function should follow the steps of a feedforward process:

1. Compute the weighted sum of inputs for each node in the hidden layer.
2. Apply the sigmoid activation function to the hidden layer outputs.
3. Compute the weighted sum of inputs for each node in the output layer.
4. Apply the sigmoid activation function to the output layer outputs.
5. Return the output vector.
6. Run the Example in the main fucntion

You can assume that the weights and biases for the network are pre-defined as the following lists:

```
input_weights = [[0.1, 0.2, 0.3],
                 [0.4, 0.5, 0.6],
                 [0.7, 0.8, 0.9],
                 [1.0, 1.1, 1.2]]

hidden_weights = [[1.3, 1.4, 1.5, 1.6],
                  [1.7, 1.8, 1.9, 2.0]]

output_weights = [[2.1, 2.2],
                  [2.3, 2.4],
                  [2.5, 2.6],
                  [2.7, 2.8]]
```

The biases for the hidden and output layers are defined as follows:

```
hidden_biases = [0.9, 1.0, 1.1, 1.2]
output_biases = [1.3, 1.4]
```

You can use the sigmoid function sigmoid(x) defined below for the activation function:

```
import math

def sigmoid(x)
    return 1 / (1 + math.exp(-x))
```

Example usage of the function:

```
input_vector = [0.1, 0.2, 0.3]
output_vector = feedforward(input_vector)
print(output_vector)
#Sample output: [0.7874101769784412, 0.8946330293610033]
```

Q No. 2: Genetic Algorithms (5+5+5 marks)

According to the **Salesman Problem**, The problem says that a salesman is given a set of cities, he has to find the shortest route to as to visit each city exactly once and return to the starting city. Implement a genetic algorithm in Python to solve the **Traveling Salesman Problem (TSP)** . Provide the code for the fitness_function, crossover, and mutation steps. Additionally, explain how the fitness function is defined and the termination condition is determined in this particular implementation. Use the helping code provided in the cactus.

Q No. 3: k-mean Clustering (10 marks)

As an ML expert, your task is to create a cluster model to group e-vehicle drivers based on their driving data. Lithionpower, the largest provider of electric vehicle batteries, wants to incentivize drivers based on these clusters. The clustering should be accurate to ensure fair and appropriate incentives for drivers.

1. Read the dataset driver.csv, where each line represents a driver's data point with the mentioned attributes.
2. Describe the dataset.
3. Plot histogram of the data.
4. Preprocess the data and handle missing values using appropriate method. **You can use built-in method or write the code from scratch, both carry equal marks**.
5. Scale the features id needed, if not comment why?
6. Apply K-Means clustering algorithm to group the drivers based on their driving data.
7. Determine the optimal number of clusters using an appropriate evaluation metric.
8. Assign each driver to a specific cluster based on the clustering results.
9. Visualize the clusters in a scatter plot, with mean_dist_day on the x-axis and mean_over_speed_perc on the y-axis.
10. Use different colors or markers for each cluster.