Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Spring, Year:2025), B.Sc. in CSE (Day)

Lab Report NO # 03
Course Title: Artificial Intelligence Lab
Course Code: CSE-316 Section:222-D2

Lab Experiment Name:
1. Neural Network using Multilayer Perceptron Algorithm in Weka.
2. K-Means Clustering Using Real Data.

<u>Student Details</u>

	Name	ID
1.	Md. Abu Hurayra	222902071

Submission Date : 27-4-2025
Course Teacher's Name : Md. Abu Rumman Refat

Lab Report Status
Marks: ………………………………… Signature:.....................
Comments:.. Date:.............................

1. TITLE OF THE LAB REPORT EXPERIMENT

1. Neural Network using Multilayer Perceptron Algorithm in Weka.
2. K-Means Clustering Using Real Data

2. OBJECTIVES/AIM

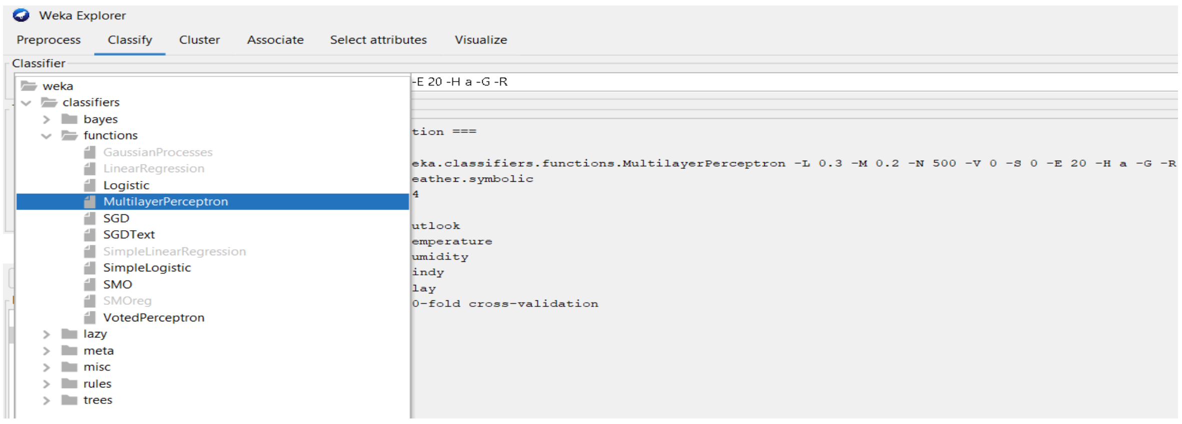
- To create and train a neural network model utilizing MLP within the Weka platform.
- To organize real-world entities into distinct groups using the K-Means clustering method.

3. PROCEDURE / ANALYSIS / DESIGN

1. **Neural Network (MLP in Weka):**
    - **Chose an appropriate dataset for classification tasks.**
    - **Uploaded the dataset into Weka's environment.**
    - **Selected Multilayer Perceptron from the available algorithms.**
    - **Modified parameters such as hidden layer structure and learning rate.**
    - **Executed the model training and assessed its accuracy.**
2. **K-Means Clustering:**
    - **Collected a real-world dataset involving university attributes.**
    - **Developed a Python-based K-Means clustering application.**
    - **Randomized initial cluster centers.**
    - **Reassigned points and recalculated centers repeatedly until stability.**

4. IMPLEMENTATION

1.



```
-E 20 -H a -G -R
```

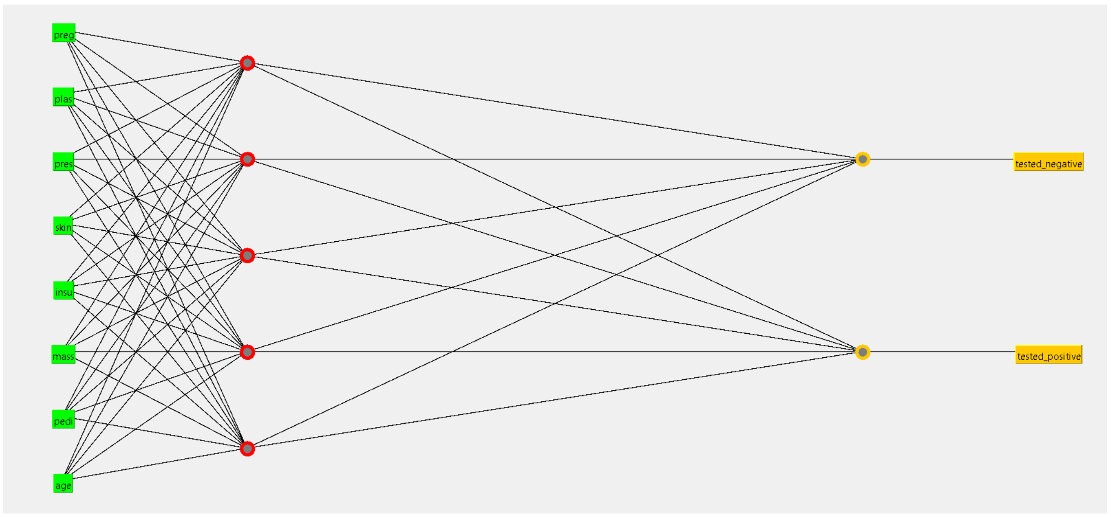
```
tion ===

eka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R
eather.symbolic
4

utlook
emperature
umidity
indy
lay
0-fold cross-validation
```

2.

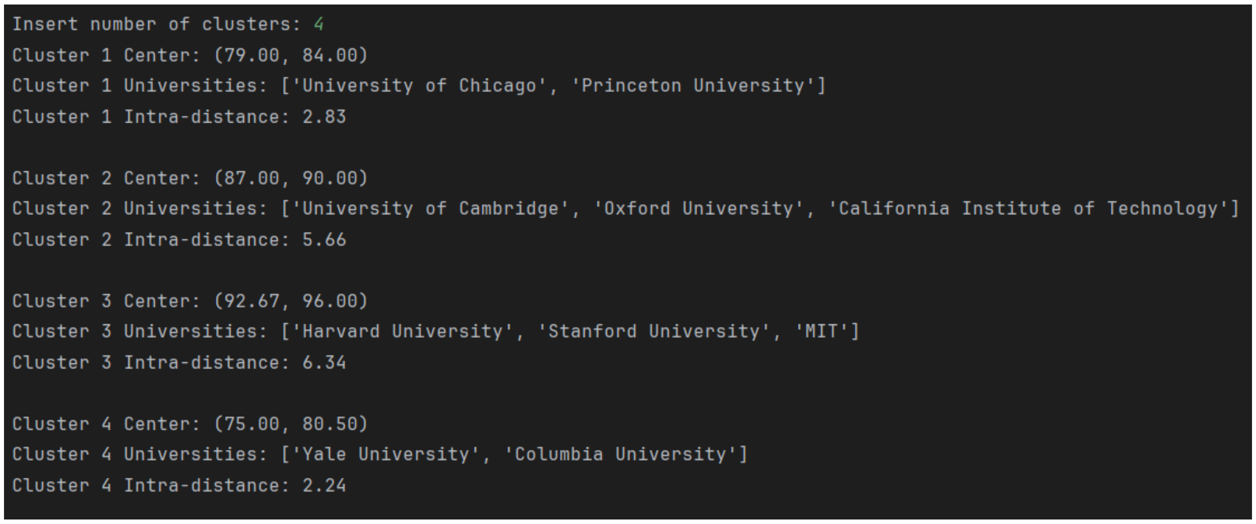
```
import math
import random

class Point:
    def __init__(self, x, y, name):
        self.x = x
        self.y = y
        self.name = name
        self.cluster = None

class KMeans:
    def __init__(self, points, clusters):
        self.points = points
        self.clusters = clusters
        self.centers = []
        self.start()

    def start(self):
        self.centers = random.sample(self.points, self.clusters)

        while True:
            for p in self.points:
                min_dist = float('inf')
                for idx, center in enumerate(self.centers):
                    d = math.sqrt((center.x - p.x) ** 2 + (center.y
- p.y) ** 2)
                    if d < min_dist:
                        p.cluster = idx
                        min_dist = d

            old_centers = [Point(c.x, c.y, c.name) for c in
self.centers]

            for idx in range(self.clusters):
                cluster_points = [p for p in self.points if
p.cluster == idx]
                if cluster_points:
                    avg_x = sum(p.x for p in cluster_points) /
len(cluster_points)
                    avg_y = sum(p.y for p in cluster_points) /
len(cluster_points)
                    self.centers[idx] = Point(avg_x, avg_y,
f"Center {idx+1}")

            error = sum(
                math.sqrt((self.centers[i].x - old_centers[i].x) **
2 + (self.centers[i].y - old_centers[i].y) ** 2)
                for i in range(self.clusters)
            )
            if error == 0:
                break

        for idx in range(self.clusters):
            cluster_points = [p for p in self.points if p.cluster
== idx]
            universities = [p.name for p in cluster_points]
            intra_distance = sum(
                math.sqrt((self.centers[idx].x - p.x) ** 2 +
(self.centers[idx].y - p.y) ** 2)
                for p in cluster_points
            )
            print(f"Cluster {idx + 1} Center:
({self.centers[idx].x:.2f}, {self.centers[idx].y:.2f})")
            print(f"Cluster {idx + 1} Universities:
{universities}")
            print(f"Cluster {idx + 1} Intra-distance:
{intra_distance:.2f}\n")

def main():
    university_data = [
        (95.0, 98.0, 'Harvard University'),
        (92.0, 96.0, 'Stanford University'),
        (91.0, 94.0, 'MIT'),
        (89.0, 92.0, 'University of Cambridge'),
        (87.0, 90.0, 'Oxford University'),
        (85.0, 88.0, 'California Institute of Technology'),
        (80.0, 85.0, 'University of Chicago'),
        (78.0, 83.0, 'Princeton University'),
        (76.0, 81.0, 'Yale University'),
        (74.0, 80.0, 'Columbia University')
    ]

    points = [Point(x, y, name) for x, y, name in university_data]
    clusters = int(input("Insert number of clusters: "))
    KMeans(points, clusters)

if __name__ == "__main__":
    main()
```

5. TEST RESULT / OUTPUT

1.



2.



```
Insert number of clusters: 4
Cluster 1 Center: (79.00, 84.00)
Cluster 1 Universities: ['University of Chicago', 'Princeton University']
Cluster 1 Intra-distance: 2.83

Cluster 2 Center: (87.00, 90.00)
Cluster 2 Universities: ['University of Cambridge', 'Oxford University', 'California Institute of Technology']
Cluster 2 Intra-distance: 5.66

Cluster 3 Center: (92.67, 96.00)
Cluster 3 Universities: ['Harvard University', 'Stanford University', 'MIT']
Cluster 3 Intra-distance: 6.34

Cluster 4 Center: (75.00, 80.50)
Cluster 4 Universities: ['Yale University', 'Columbia University']
Cluster 4 Intra-distance: 2.24
```

6. ANALYSIS AND DISCUSSION

The MLP model built in Weka delivered strong classification outcomes with relatively straightforward configuration. The performance signified that the network effectively captured the underlying data patterns.

K-Means clustering successfully categorized universities sharing similar performance indicators, forming logical and interpretable groups. Despite some variation in distances, the clusters revealed meaningful relationships.

7. SUMMARY:

This exercise demonstrated effective use of both supervised and unsupervised learning techniques. The MLP provided solid predictive capabilities, while K-Means revealed natural groupings within the dataset. The practical sessions improved understanding of both theoretical concepts and their real-world applications.