# Abhishek Murthy

**21BDS0064**

# Fall Sem 2024-2025

**DA - 4**

# Machine Learning Lab 25-09-2024

## K Means

**Dataset:**

|  |  |
| --- | --- |
| **X** | **Y** |
| 1 | 2 |
| 2 | 1 |
| 4 | 5 |
| 5 | 4 |
| 8 | 9 |
| 9 | 8 |
| 20 | 21 |
| 21 | 20 |
| 24 | 25 |
| 25 | 24 |
| 50 | 50 |

**Code:**

import random import math

*def* euclidean\_distance(*point1*, *point2*):

return math.sqrt(sum((p1 - p2) \*\* 2 for p1, p2 in zip(*point1*, *point2*)))

*def* assign\_clusters(*data\_points*, *centroids*): clusters = [[] for \_ in *centroids*]

for point in *data\_points*:

distances = [euclidean\_distance(point, centroid) for centroid in *centroids*] closest\_centroid\_index = distances.index(min(distances))

clusters[closest\_centroid\_index].append(point) return clusters

*def* update\_centroids(*clusters*): new\_centroids = []

for cluster in *clusters*: if cluster:

new\_centroid = [sum(dim) / len(cluster) for dim in zip(\*cluster)] new\_centroids.append(new\_centroid)

else:

new\_centroids.append([random.random() for \_ in *clusters*[0][0]]) return new\_centroids

*def* k\_means\_clustering(*data\_points*, *k*, *max\_iterations*=100): centroids = random.sample(*data\_points*, *k*)

for iteration in range(*max\_iterations*):

clusters = assign\_clusters(*data\_points*, centroids) new\_centroids = update\_centroids(clusters)

if new\_centroids == centroids:

print(*f*"Converged in {iteration} iterations.") break

centroids = new\_centroids return clusters, centroids

data\_points = [

[1, 2], [2, 1], [4, 5], [5, 4], [8, 9], [9, 8],

[20, 21], [21, 20], [24, 25], [25, 24], [50, 50]

]

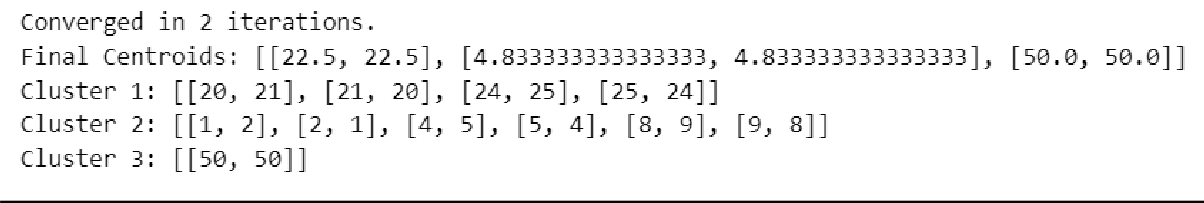
k = 3

clusters, centroids = k\_means\_clustering(data\_points, k)

print("Final Centroids:", centroids)

for idx, cluster in enumerate(clusters): print(*f*"Cluster {idx+1}: {cluster}")

**Output:**



## Principal Component Analysis:

**Dataset:**

|  |  |
| --- | --- |
| **X** | **Y** |
| 4 | 11 |
| 8 | 4 |
| 13 | 5 |
| 7 | 14 |

**Code:**

**Output:**

