National University of Computer and Emerging Sciences

Lab Manual
CL461-Artificial Intelligence Lab

Department of Computer Science
FAST-NU, Lahore, Pakistan

3. Machine Learning Concepts:

Unsupervised Learning is a machine learning technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabelled data.

3.1 Algorithms:

Unsupervised Learning Algorithms allow users to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning methods. Unsupervised learning algorithms include clustering, anomaly detection, neural networks, etc

3.2 Why Unsupervised learning?:

Here, are prime reasons for using Unsupervised Learning:

- Unsupervised machine learning finds all kind of unknown patterns in data.
- Unsupervised methods help you to find features which can be useful for categorization.
- It is taken place in real time, so all the input data to be analyzed and labeled in the presence of learners.
- It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention.

3.3 Types of unsupervised learning

Unsupervised learning problems further grouped into clustering and association problems.

3.4 Clustering:

Clustering is an important concept when it comes to unsupervised learning. It mainly deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data. You can also modify how many clusters your algorithms should identify. It allows you to adjust the granularity of these groups.

3.5 Types of Clustering:

- **Exclusive (partitioning)**

In this clustering method, Data are grouped in such a way that one data can belong to one cluster only.

Example: K-means

- **Agglomerative**

In this clustering technique, every data is a cluster. The iterative unions between the two nearest clusters reduce the number of clusters.

Example: Hierarchical clustering

4- Clustering Types

- Hierarchical clustering
- K-means clustering
- K-NN (k nearest neighbors)
- Principal Component Analysis
- Singular Value Decomposition
- Independent Component Analysis

4.1 KMeans Clustering

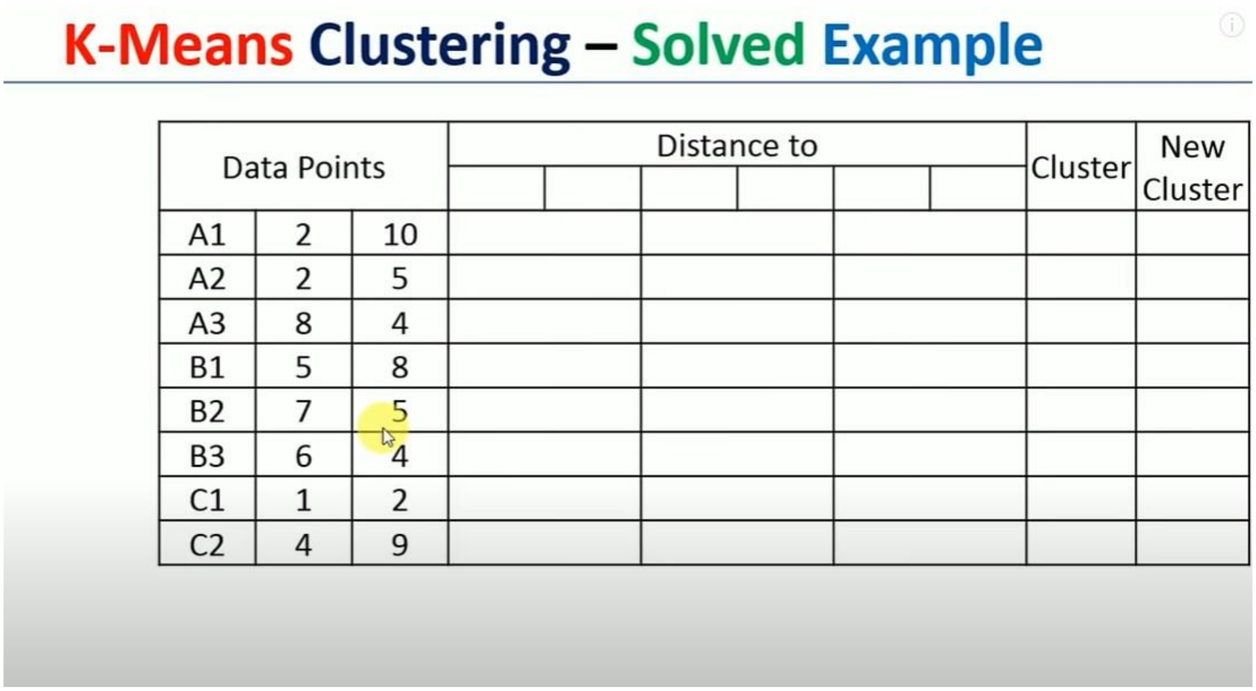
K means it is an iterative clustering algorithm which helps you to find the highest value for every iteration. Initially, the desired number of clusters are selected. In this clustering method, you need to cluster the data points into k groups. A larger k means smaller groups with more granularity in the same way. A lower k means larger groups with less granularity.

The output of the algorithm is a group of "labels." It assigns data point to one of the k groups. In kmeans clustering, each group is defined by creating a centroid for each group. The centroids are like the heart of the cluster, which captures the points closest to them and adds them to the cluster.

Code Example:

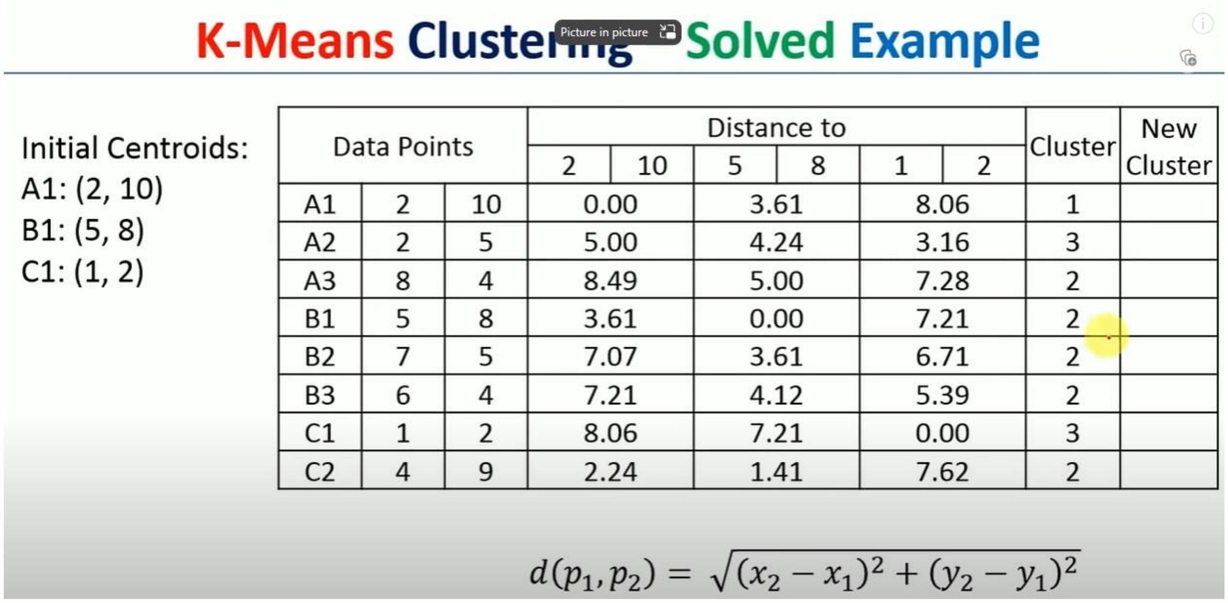
```
from sklearn.cluster import KMeans kmeans
= KMeans(n_clusters=4, max_iter=50)
kmeans.fit(dataframe)
```

```
For plotting:
 import matplotlib.pyplot as
plt
%matplotlib inline

plt.figure(figsize=(10, 7))
plt.scatter(df['var1'], df['var2'], c=cluster.labels_)
```



K-Means Clustering – Solved Example

Data Points			Distance to						Cluster	New Cluster
A1	2	10								
A2	2	5								
A3	8	4								
B1	5	8								
B2	7	5								
B3	6	4								
C1	1	2								
C2	4	9								

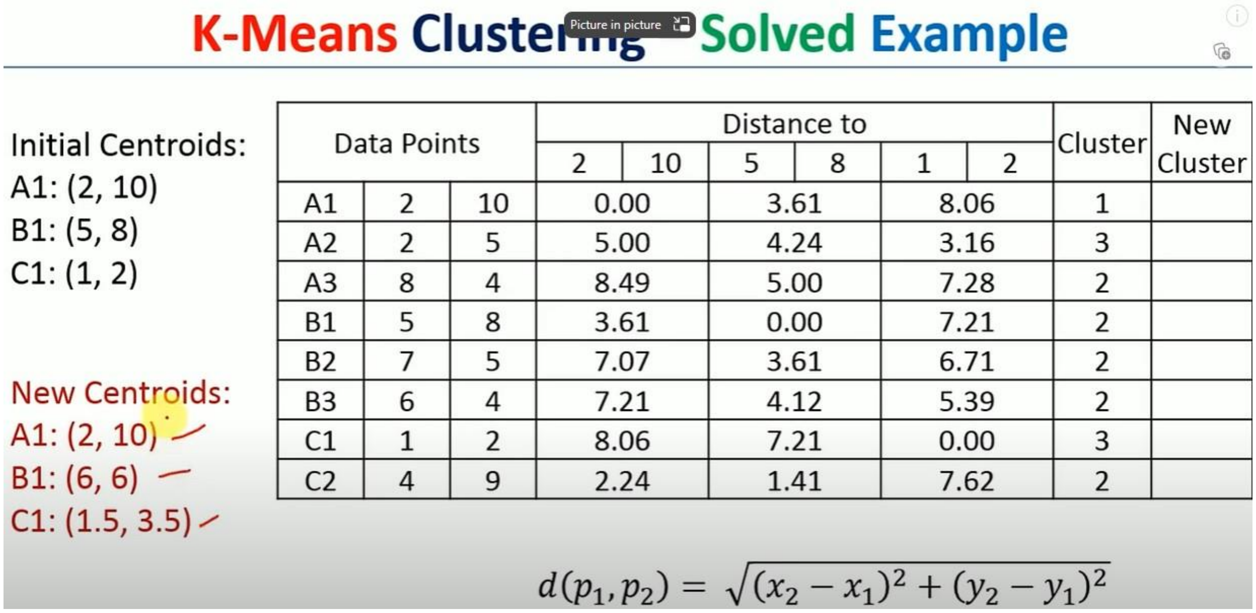


K-Means Clustering – Solved Example

Initial Centroids:
A1: (2, 10)
B1: (5, 8)
C1: (1, 2)

Data Points			Distance to						Cluster	New Cluster
			2	10	5	8	1	2		
A1	2	10	0.00		3.61		8.06		1	
A2	2	5	5.00		4.24		3.16		3	
A3	8	4	8.49		5.00		7.28		2	
B1	5	8	3.61		0.00		7.21		2	
B2	7	5	7.07		3.61		6.71		2	
B3	6	4	7.21		4.12		5.39		2	
C1	1	2	8.06		7.21		0.00		3	
C2	4	9	2.24		1.41		7.62		2	

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



K-Means Clustering Solved Example

Initial Centroids:
A1: (2, 10)
B1: (5, 8)
C1: (1, 2)

Data Points			Distance to						Cluster	New Cluster
			2	10	5	8	1	2		
A1	2	10	0.00		3.61		8.06		1	
A2	2	5	5.00		4.24		3.16		3	
A3	8	4	8.49		5.00		7.28		2	
B1	5	8	3.61		0.00		7.21		2	
B2	7	5	7.07		3.61		6.71		2	
B3	6	4	7.21		4.12		5.39		2	
C1	1	2	8.06		7.21		0.00		3	
C2	4	9	2.24		1.41		7.62		2	

New Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- ### 5. Hierarchical Clustering:

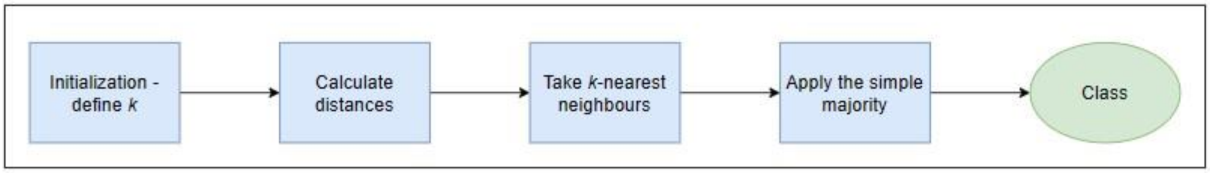
Hierarchical clustering is an algorithm which builds a hierarchy of clusters. It begins with all the data which is assigned to a cluster of their own. Here, two close cluster are going to be in the same cluster. This algorithm ends when there is only one cluster left.

Code Sample:

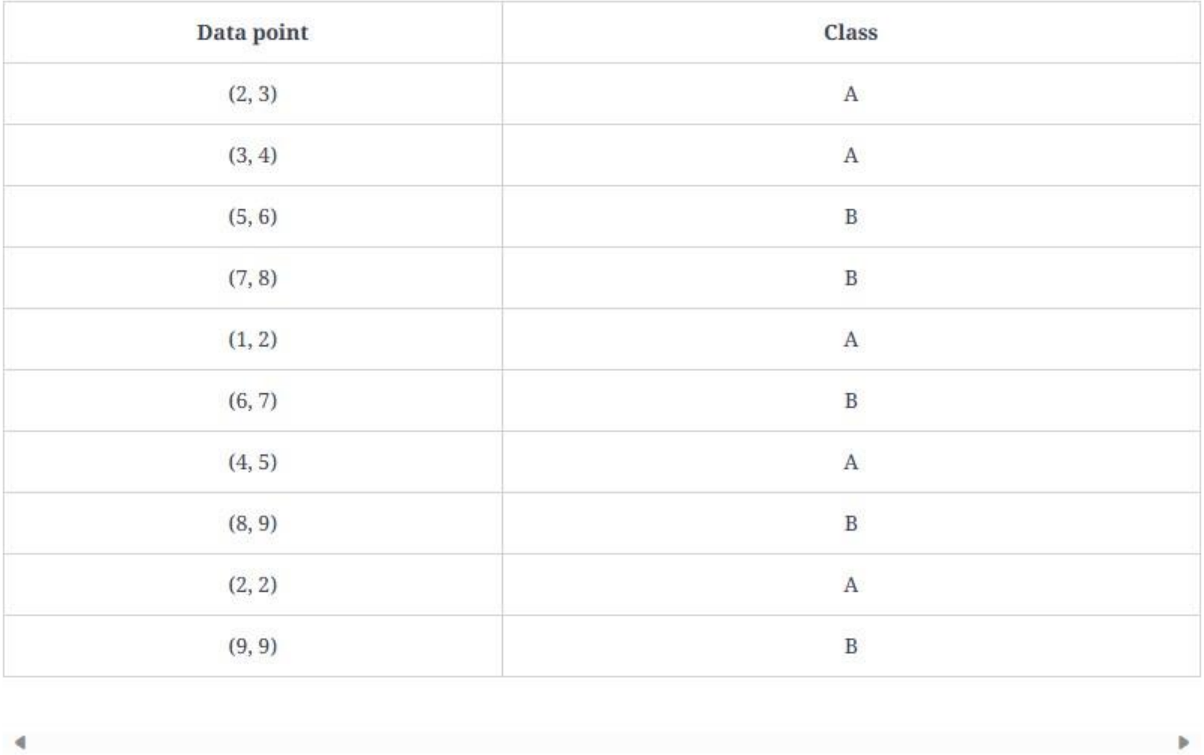
```
from sklearn.cluster  import
AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=2, affinity='euclidean',
linkage='ward')
cluster.fit_predict(data_scaled)

For plotting:
 import matplotlib.pyplot as
plt
%matplotlib inline

plt.figure(figsize=(10, 7))
plt.scatter(df['var1'], df['var2'], c=cluster.labels_)
```

Introduction to *k*-NN

What is *k*-NN? As mentioned above, *k*-NN is a widely recognized classification technique used to assign items to categories based on how similar they are to nearby data points. It falls under the category of instance-based or lazy learning algorithms. Unlike some other algorithms that build explicit models during training, *k*-NN makes predictions by finding the most similar data points in the training dataset to the item being classified.

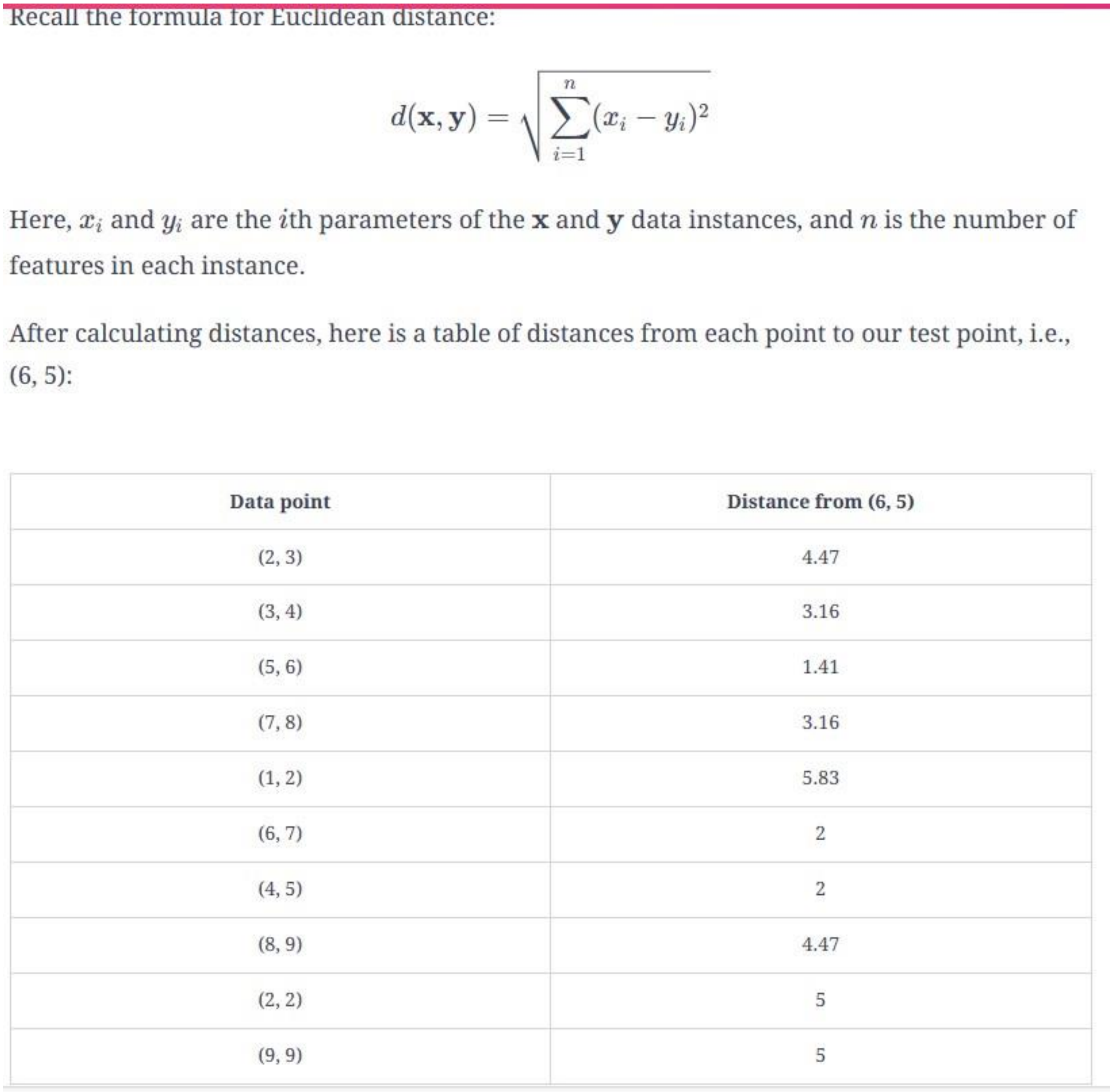

Initialization - define *k*
Calculate distances
Take *k*-nearest neighbours
Apply the simple majority
Class

k-NN in steps



Data point	Class
(2, 3)	A
(3, 4)	A
(5, 6)	B
(7, 8)	B
(1, 2)	A
(6, 7)	B
(4, 5)	A
(8, 9)	B
(2, 2)	A
(9, 9)	B

We also have the following test data point: (6, 5)



Recall the formula for Euclidean distance:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Here, x_i and y_i are the ith parameters of the $\mathbf{x}$ and $\mathbf{y}$ data instances, and n is the number of features in each instance.

After calculating distances, here is a table of distances from each point to our test point, i.e., (6, 5):

Data point	Distance from (6, 5)
(2, 3)	4.47
(3, 4)	3.16
(5, 6)	1.41
(7, 8)	3.16
(1, 2)	5.83
(6, 7)	2
(4, 5)	2
(8, 9)	4.47
(2, 2)	5
(9, 9)	5

Lab Task:

- Apply the K-means algorithm to the provided dataset and provide code implementations using both a relevant library and from scratch.

- Implement the k-NN algorithm both from scratch and utilizing a library

- Implement agglomerative **Hierarchical Clustering on given dataset** to the provided dataset and provide code implementations using both a relevant library and from scratch.