Module 4

Unsupervised learning

→ no supervision is provided to the algorithm
→ deals with unlabeled dataset

Clustering - The process of dividing datasets into groups consisting of similar data points.

Applications -

1) Market Segmentation
2) Statistical data analysis
3) Social network analysis
4) Image Segmentation, etc.

Customer segmentation for targeted marketing is one of the most vital applications of the clustering algorithm. Here, as a manager of the online store he/she could want to group the customers into different clusters, so that he can make a customised marketing campaign for each of the group. He does not have any label in mind such as good or bad customer. He want to just look at patterns in customer data & then to try & find segments.

Clustering techniques use the raw data to form clusters based on common factors among various data points

Difference b/w segmenting and clustering

segmenting is the process of putting customers into groups based on similarities and clustering is the process of finding similarities in customers so that they can be grouped, and therefore segmented.

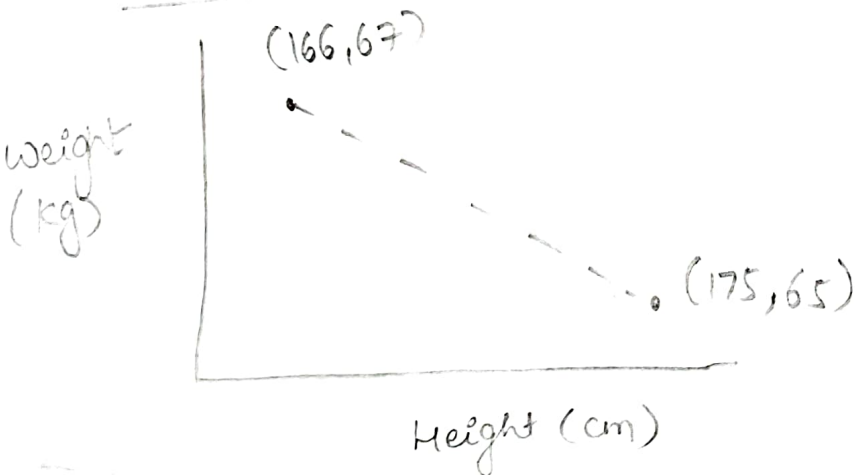
For successful segmentation, the segments formed must be stable. This means that the same person should not fall under different segments upon segmenting the data on the same criteria. Segments should have intra-segment homogeneity and inter segment heterogeneity

Clustering Algorithm

1. K-Means algorithm

clustering works on the basis of grouping the observations which are the most similar to each other.
In simple terms, the algorithm needs to find data points whose values are similar to each other & therefore these points would then belong to the same cluster. The method in which any clustering algorithm goes about doing that is through the method of finding something called 'distance measure'. The distance measure that is used in K-means clustering is called the Euclidean Distance measure.

observation	Height (cm)	weight (kg)
A	175	65
B	166	67



Euclidean distance is simply the length of the straight line joining the 2 points

Point $X = (X_1, X_2) = (175, 65)$

Point $Y = (Y_1, Y_2) = (166, 67)$

$$D = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2} = \sqrt{(175-166)^2 + (65-67)^2}$$

$$= \sqrt{(9)^2 + (-2)^2} = \sqrt{81+4} = \sqrt{85}$$

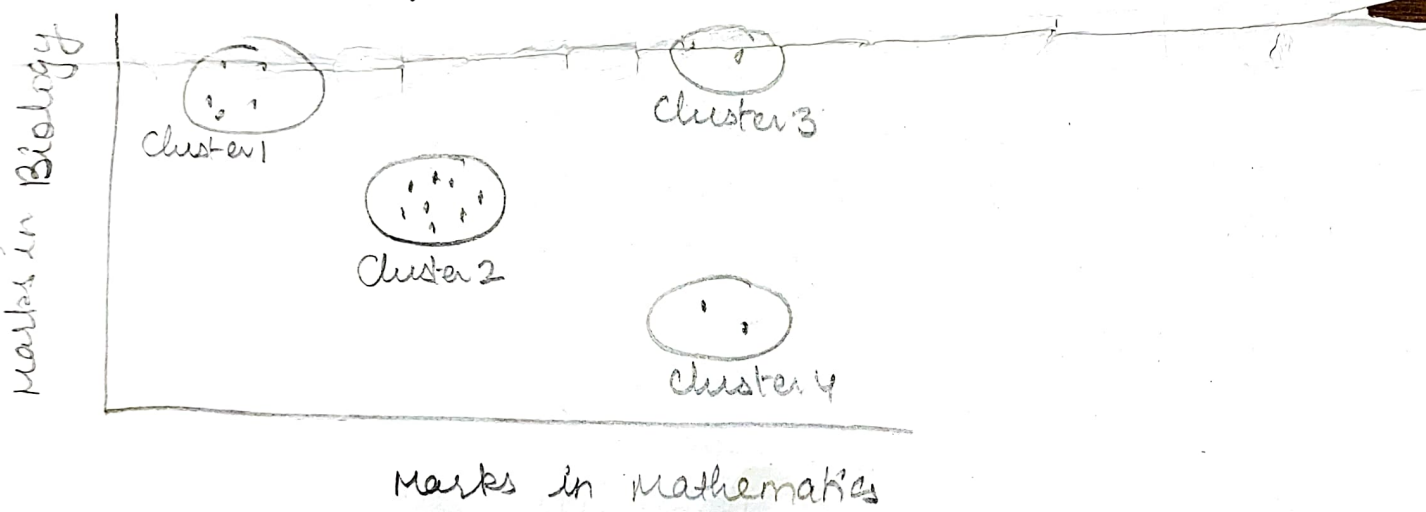
$$= 9.22$$

General eqn when there are n dimensions -

$$D = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \cdots (X_n - Y_n)^2}$$

Centroids

→ center point of the clusters



* missing one crucial information – the numerical order. eg if we want to compare two clusters we can't say how much marks on average do the students from Cluster 1 outperform or underperform the Cluster 2 students in a particular subject just by taking a look at the above visualisation alone. Is it by 10 marks? or 15?

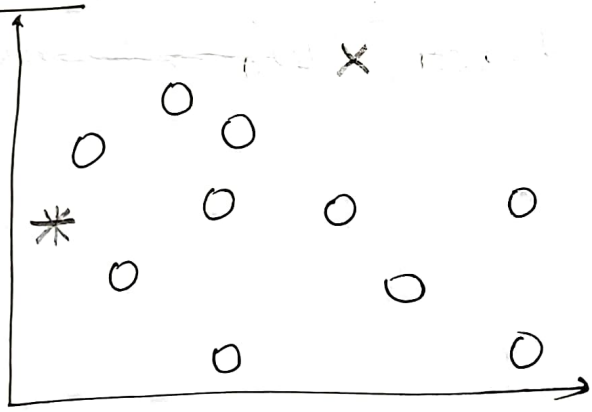
This is where the concept of Centroids comes.

Computing Centroids

The cluster centres for a particular cluster that we compute in K-means Algorithm is given by the Centroid value for those cluster points

Observation	Height	weight	Age
A	175	83	22
B	165	74	25
C	183	98	24
D	172	80	24
Centroid	$\frac{(175+165+183+172)}{4}$ = 173.75	$\frac{(83+74+98+80)}{4}$ = 83.75	$\frac{(22+25+24+24)}{4}$ = 23.75

Algorithm



n=10 datapoint
We want to divide them in two clusters
K in K means is the no. of clusteres
Here K=2

1) start by choosing k random points which will be initial centroids
2) Allocate each point in the data set to the nearest cluster center. We do so by calculating the distance of each data from the two centres & allocate point to the ~~closed~~ centroid with least distance. Euclidean distance is used. (Assignment step)
3) Recompute the centre of each of these 2 clusters which is calculate by taking mean of individual datapoints in each of the cluster. Then we will get new centroids. (Optimization step)

4) Again go to step 2 & perform the same steps.
5) Keep iterating through the process of assignment & optimisation till the centroids no longer updates. This is the step where algo has reached optimal grouping

The algorithm inner loop iterates over two steps:

1. Assign each observation X_i to the closest cluster centroid μ_K

$$Z_i = \text{argmin} \| X_i - \mu_K \|^2$$

2. Update each centroid to the mean of the points assigned to it.

$$\mu_K = \frac{1}{n_K} \sum_{i : Z_i = K} X_i$$

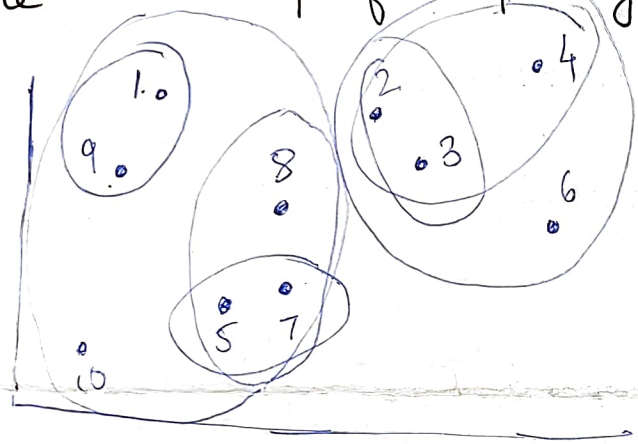
Hierarchical Clustering Algorithm

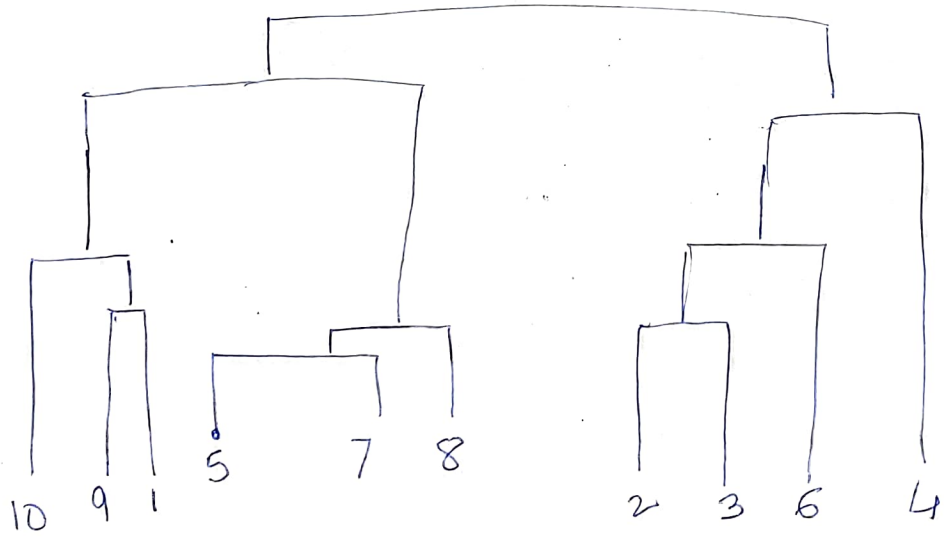
One of the major considerations in using k-means algorithm is deciding the value of k beforehand. The hierarchical clustering algorithm does not have this restriction.

The output of hierarchical clustering algorithm is quite different from the k-mean algorithm. It results in an inverted tree-shaped structure called dendogram

Given a set of N items to be clustered, the steps in hierarchical clustering are:

1. Calculate the N X N distance (similarity) matrix, which calculates the distance of each data point from the other.
2. Each item is first assigned to its own cluster i.e N clusters are formed
3. The clusters which are closest to each other are merged to form a single cluster.
4. The same step of computing





Bottom up approach - Hierarchical Agglomerative Clustering

K-medoids Clustering

→ slightly modified from K-means

→ a medoid can be defined as that object of a cluster whose average dissimilarity to all the objects in the cluster is minimal.

example

i	x	y
X_1	2	6
X_2	3	4
X_3	3	8
X_4	4	7
X_5	6	2
X_6	6	4
X_7	7	3
X_8	7	4
X_9	8	5
X_{10}	7	6

Step 1 Select two random representative objects ($K=2$)

C_1 (3, 4) → X_2

C_2 (7, 4) → X_8.

step 2

i	x	y	C_1	C_2	Distance / cost	C
X_1	2	6	3	4	$\|2-3\| + \|6-4\|$	3
X_3	3	8	3	4	$\|3-3\| + \|8-4\|$	4
X_4	4	7	3	4		4
X_5	6	2	3	4		5
X_6	6	4	3	4		3
X_7	7	3	3	4		5
X_9	8	5	3	4		6
X_{10}	7	6	3	4		6

* Manhattan distance

C	x	y	C_2		Distance
x_4	2	6	7	4	7
x_3	3	8	7	4	8
x_4	4	7	7	4	6
x_5	6	2	7	4	(3)
x_6	6	4	7	4	(1)
x_7	7	3	7	4	(1)
x_9	8	5	7	4	(2)
10	7	6	7	4	(2)

step3 Compare cost of Cost(C_4) and cost (C_2) for every i & select minimum cost

step4 Calculate total cost
= (3+4+4) + (3+1+1+2+2)
= 20

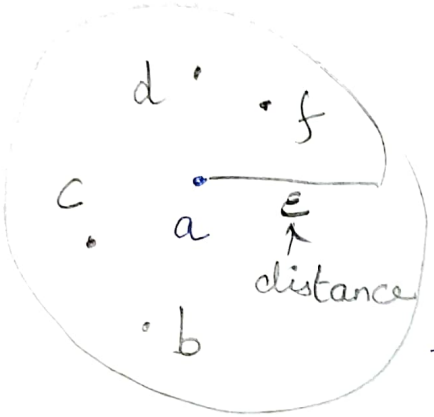
step5 select another point to be a medoid.
say C_1 (3,4) and C_2 (7,3)

Repeat same task
Total cost when (7,3) is the medoid > total cost when (7,4) was the medoid earlier
Hence choose (7,4)
Since there is no change in the medoid set, the algorithm ends here.

Density Based Spatial clustering of Applications with Noise (DB SCAN) -

Density - no. of points which are in a given area

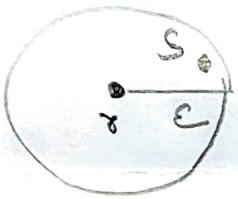
Two inputs are given < Epsilon (circle radius) ε Epsom (2)
Min points



When we take a datapoint as a center & using this centre & epsilon radius if we draw a circle then how many min pts must be there in the circle is defined by Min points

* If this datapoint satisfies min points cond" then it becomes core point

Boundary point (s) → pt which is neighbour of core point



Noise point → the point which is not a core point nor a boundary point. These are outliers which are excluded from the cluster

That is why DBSCAN is Robust.

Algorithm

(1) The algo proceeds by arbitarily picking up a point in the dataset (until all points have been visited)
(2) If there are atleast 'minpoints' within the radius of 'ε' to the point then we consider all these points to be a part of the same cluster.
(3) The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point.

a

[Gaussian Clustering Model — SL
Bayesian Natural Network — SL]