We can divide Machine Learning into two parts

① Supervised ML ② Unsupervised ML

→ when ever we talk about Supervised ML algo we will have target/dependent/ Supervisor.

Algorithms inside unsupervised

① k-means —updated→ k-mean++ cluster
② hierachical clustering
③ DBScan clustering

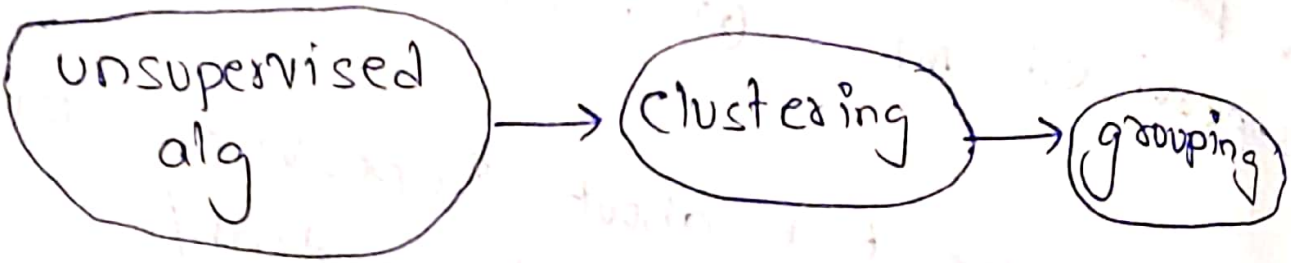
→ Most of time we use this three algo for clustering

Agenda behind taking Unsupervised data

→ If we see below example based on "Height" and "weight" we can predict the "BMI" because "BMI" is Supervisor

Height	weight	BMI
170	60	21
180	65	22
160	70	20

→ In unsupervised algo we will not perform prediction. we will do grouping the dataset.



→ In the below example we are clustering the data based on country.

ex:-

Height	weight	BMI	Country
170	60	21	IND
180	65	22	UK
160	70	20	USA

→ To do this clustering we will use matical eqn

1. k-mean
2. hiearchical
3. DB-Scan

KMEANS

cluster formation

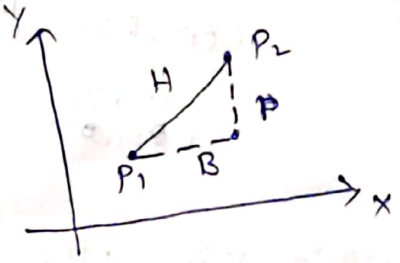
→ Each cluster is formed based on the distance

① Euclidean distance measure
② Manhatten " "
③ Cosine " "
④ Tanimoto " "
⑤ Squared Euclidean " "

K-mean

→ we will have dataset. we need to findout similarity b/w dataset by using distance formula like Euclidean

Equation for Euclideans distance



pythagoras

$H^2 = P^2 + B^2$

$H = \sqrt{P^2 + B^2}$

$$D(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The main Concepts in k-mean are

① Centroid
② Distance
③ mean

→ In k-mean "k" is no. of centroids
→ "k" is desides based on
  ① ELbow
  ② WCSS (within cluster sum of Square)
→ ① Inter cluster
  ② Intra cluster

Evalution matrix

① dunn index
② Silhouette Index

process

step1:- Select two data points as centroids. k=2

$C_1(x_1, y_1)$ $C_2(x_2, y_2)$

step2:- Find the E.D from C_1, C_2 to remaining points. which is nearest C_1 we add it into that cluster.

ED: C_1, NP$_1$ ED: C_2, NP(x_3, y_3)

$ED = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

step 3:- After adding new point into cluster we update the centroid of that cluster.

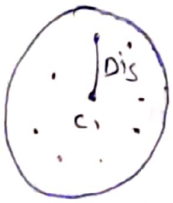
$$N_C = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$$

EL-blow method

→ This method is used to find the "K" value

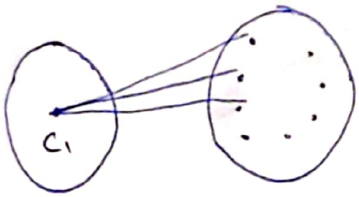
Intra cluster distance:-

→ The distance blw a data item and the cluster centroid within a cluster



Inter cluster distance

→ The distance blw centroid of one cluster to data point in another cluster

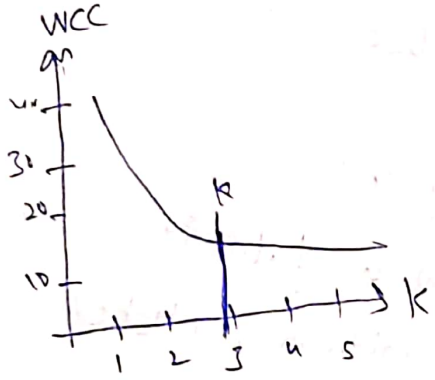


WCSS:- With in cluster sum of square

$$WCC = \sum_{i=1}^{n} d(c_i x_i)^2$$

→ To find "k" we will start with one cluster and cal WCC then one more cluster and so.. on

→ Finally we see graph like below and choose "k" value



Diff b/w k-mean & k-mean ++

→ In k-mean centroid initialation is random

→ In k-mean ++, we take ~~one~~ point based on longest distance

→ In k-mean++ it may choose outlier

How to Validate cluster

These two ways to Validate

① dunn index (Explain in nxt class)

② Silhouette score

Silhouette score

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad , \text{if } |C_I| > 1$$

→ The score come in b/w "-1 to +1"

→ If score near to +1 it's best model

→ If score near to -1 it's woost model

a(i) = mean distance from centroid of the cluster to another point in same cluster

b(i) = mean distance from centroid of one cluster to another points in different cluster

For more detail cheat wiki

How to create best model

Custom model (or) Custom learning =

Supervised + Unsupervised

weight	height	Gender

→ Create cluster based on Gender

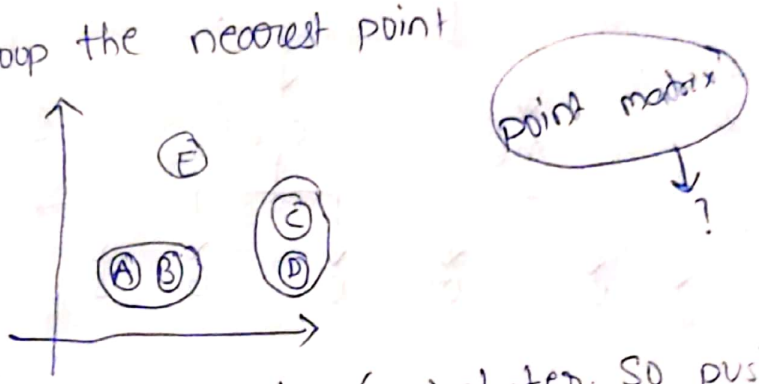
→ Then build supervised model on weight, height.

② Hierarchical clustering

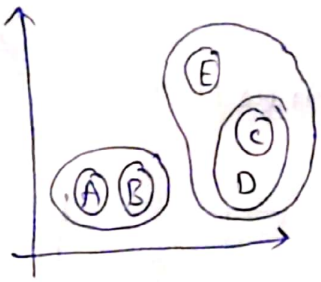
→ Let Consider few data point

Ⓐ, Ⓑ, Ⓒ, Ⓓ, Ⓔ

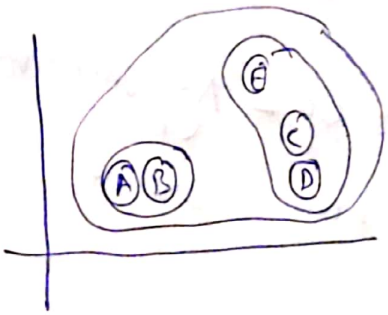
→ Group the nearest point



→ 'E' is near to (C,D) cluster. So push 'E' to (C,D) cluster



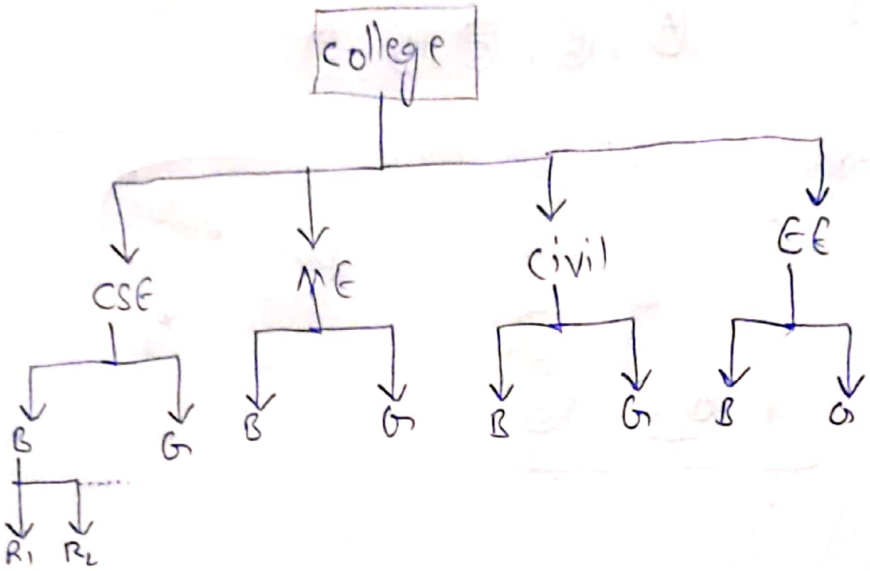
→ (E, C,D) cluster is near to (A,B) cluster. So group all the point into one cluster



→ we will create cluster at each and every point.

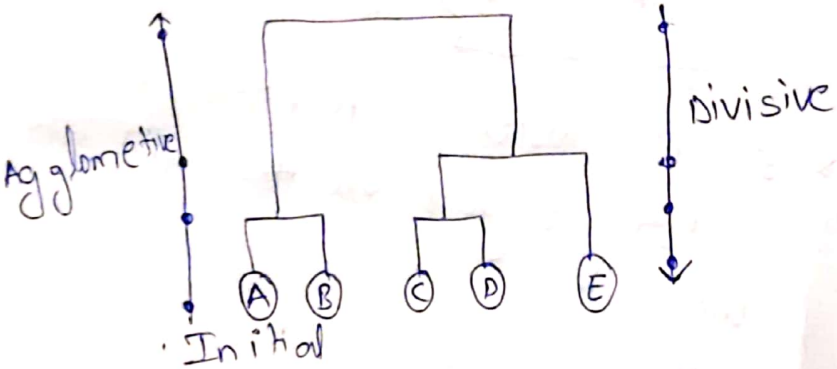
→ Here we don't need 'k' value.

Example of Hierarchical cluster



Den do gram

→ Representation of Hierichical cluster.



★ going down to up - Agglometive

★ going up to down - Divisive

① each point is cluster

② Bottom to up approach

③ Combing all the point as a single cluster

Note: Generally we will prefer Agglomerative method than Divisive

③ DB-Scan (Density Base spatial clustering with application with noise)

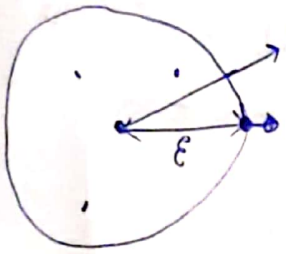
→ It is Density based approach
→ we can find outliers in it
We will create cluster based on below terminology

① eps (ε) epsilon distance
② Core point
③ Border point
④ Noise point-(Outlier)
⑤ min-point

DB-Scan steps

Step-1 : Initialise eps, min point (Hyp)

Ex:- EPS - Distance = ε } Hyperparameter
min-point = 4



→ Core point (If min-point condition is statisfied)

Step-2 :- We will take one point from the circle and draw circle with 'ε' distance

→ If does not contain min-point in circle. either It will be noise point (or) Border point

noise point :- If circle dose not have Core point
Border point :- If circle have Core point

→ we will use DBScan where k-mean not able to find cluster

[Read wiki DBScan]

[check sklearn cluster to find best algo to do cluster]

Overview of algos

① k-mean
→ centroid based approach

② Hierarical cluster
→ check each and every possibility of clusto

③ DB-Scan
→ Densit base

point matrix

→ Consider 5 points

P_1, P_2, P_3, P_4, P_5

find distance b/w the point and create cluster

step 1

	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	3	7	0		
P_4	6	5	9	0	
P_5	11	10	2		0

step 2

	P_1	P_2	$[P_3, P_5]$	P_4
P_1	0			
P_2	9	0		
P_3, P_5	3	7	0	
P_4	6	5	8	0

$$d\left(P_1, [P_3, P_5]\right) = \min\left[d(P_1, P_3), d(P_1, P_5)\right]$$

min