* Clustering

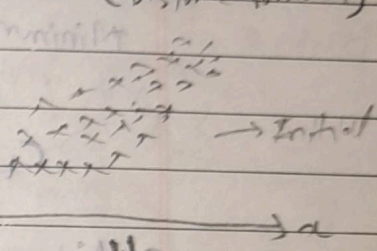
{clustering is Data Mining Approach} ← it is not ML

→ clustering (group)
- → kmeans and Kmeans++ (Enhance version of kmeans)
- → Mean shift algo
- → Hirarchical algo
- → DB Scan algo
- → Mini batch algo

we are going to create a group

Q) How to grouping the data?
→ Based on Distance.
→ Between Data point we are going to calculate distance

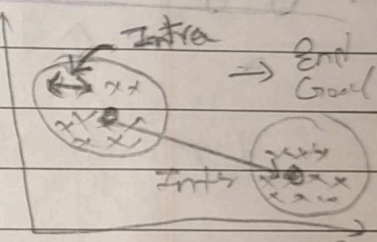
Definition: clustering → is a → Grouping of data → based on → Distance → Distance can be many type (Distance formula)



* K-Means Algo

① Concept of the Centroid
② Distance
(i) Intra (within cluster)
(ii) Inter (Between cluster)

Evaluvation of cluster ⇒ WCSS (dist Sum of Squared) → elbow plot



K-Means++ ← conversion of mean
(No. of clusters) ↑ (mean or Average)

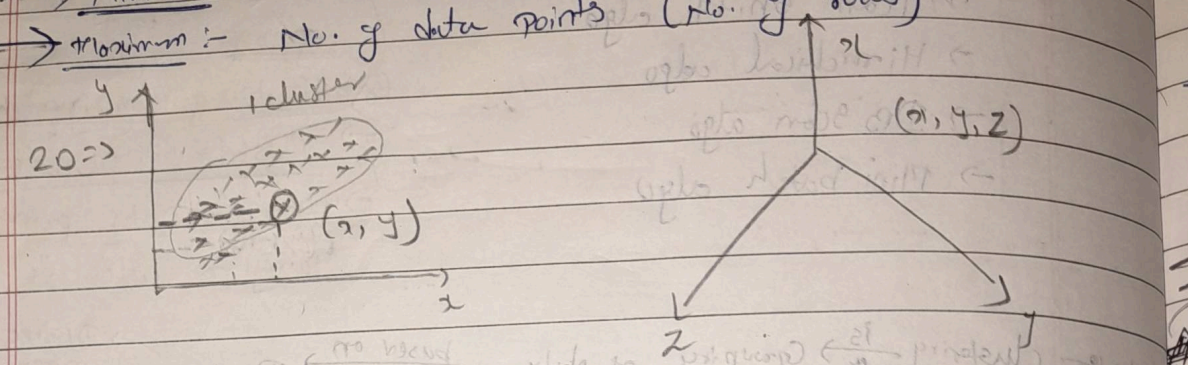
Datapoints	x_1 Height	x_2 weight
1	180	60
2	170	50
3	160	40
4	165	30
5	185	45
6	160	65
7	167	75
8	190	80
9	195	90
10	200	100

⇒ K-means
- it is all about centroid
- Centroid
  - → for initializing the centroid the method is random method

① Calculate a centroid (datapoint) → Initialize centroid randomly

→ Minimum :- 1 cluster we can create

→ Maximum :- No. of data points (No. of rows)



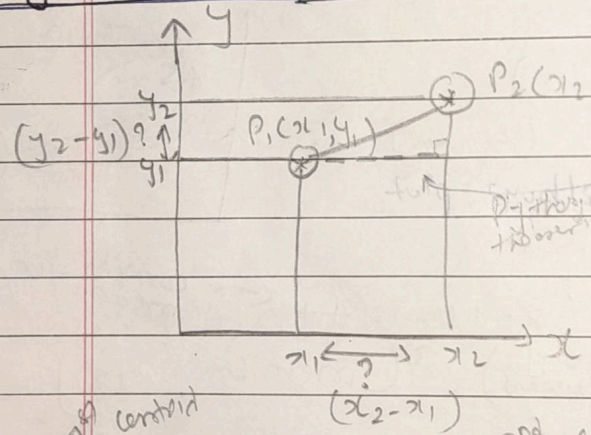
Minimum two => K = no. of centroid ∴ K = 2

1st centroid → C_1 (180, 60) This is my 1st cluster

2nd centroid → C_2 (170, 50) This is my 2nd cluster

(Randomly we are choosing 2 point)

Proof of Euclidian Distance



$H^2 = B^2 + P^2$ (perpendicular)^2

$H^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2$

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

1st centroid → C_1 (180, 60)

2nd centroid → C_2 (170, 50)

3rd point → (160, 40)

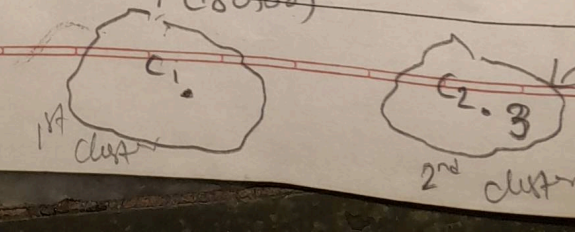
(from C_1 and C_2 we will calculate distance for the 3rd point)

Centroid formula

① C_1 to 3 $\sqrt{(160-180)^2 + (60-40)^2}$ = 28.28 (distance is v.v. high)

C_2 to 3 $\sqrt{(170-160)^2 + (50-40)^2}$ = 14.14 (less distance)

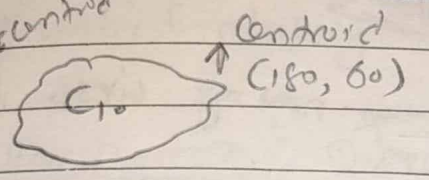
In C_2 the distance is v.v. near

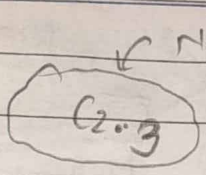


New point query over here. So, we have to update this cluster

How we are initializing centroid in K-means? (Interview Question)

Steps:-
1) Initilize a centroid (Randomly)
2) Distance
3) fill the cluster
4) Update the centroid ← (Here we Conversion of centroid)





New point going here. Cie 3rd point
So, we have to update this centroid

(170, 50) (160, 40)

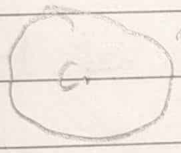
AVG

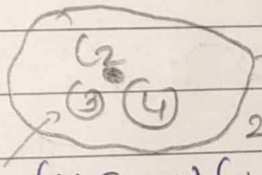
$\left(\frac{170+160}{2}, \frac{50+40}{2}\right)$

Now my point will be (165, 45) ← updated centroid

1st centroid — C1 (180, 60)
2nd centroid — C2 (165, 45)
4th (165, 30)

② C1 to 4 $\sqrt{(180-165)^2+(60-30)^2}$ = 33.54
C2 to 4 $\sqrt{(165-115)^2+(45-30)^2}$ = 15





(165, 45) ← centroid
we have to update this

Inside this we have 2 point (165, 45) (165, 30)

$\left(\frac{165+165}{2}, \frac{45+30}{2}\right)$

C1 (180, 60) C2 (165, 37.5) (165, 37.5) new centroid value of C2

③ Q) If my Intra-Distance is less, so my cluster is Good or Bad?

⇒ Good

Dense cluster ⇒ less Intra distance (within cluster distance) it is good cluster

• Greater or Big or more Inter distance is bad (Sparse cluster)

WORLD STAR™
Date :
Page :

We can see in graph if we have 1 cluster my wcss is v.v. high

wcss value

cluster is v.v. high
High wcss

This is called Elbow method

Elbow method

optimized value

now my wcss is constant

0 1 2 3 4 5 6 7 K-value

choose K=4 for our data my model

- K=4 → validation using mathematically

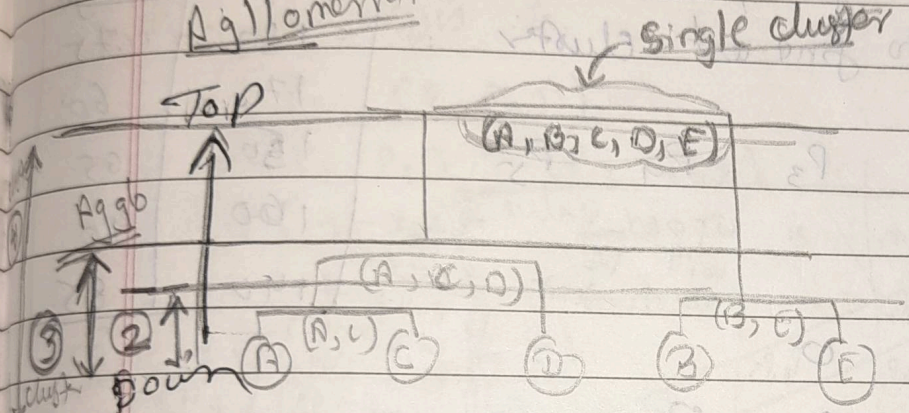
* Hierarchical Clustering

Hierarchy of cluster — grouping of data (No. of cluster = No. of data point)

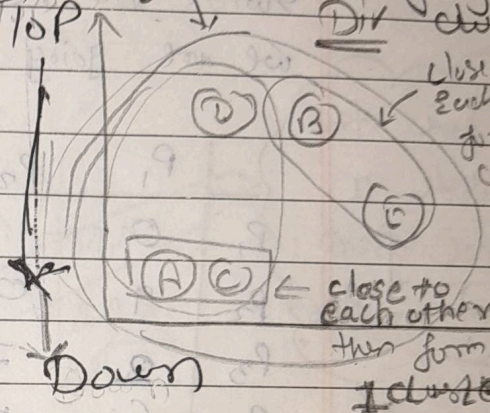
Hierarchical clustering → Agg hierarchical (Bottom up)
→ Divisive (Top Down)
(Initially we are going to consider Single cluster)

⇒ Dendogram : Diagram which shows is representing cluster

Agglomerative



Div



* Difference between Hierarchical and K-means (Refer Intro)

• Single cluster (Individual)

↳

x_1 Height	x_2 weight	
160	60	← This data point we will consider as a single cluster
170	70	←
160	40	←

Note:- Hierarchical :- Only once we are going to define a hierarchi and, easily we are going to define no. of cluster

Different

K-means

In, K = 4, 5, 6, 7 &

2nd May

5 cluster to 1 cluster

* Agglomerative → Bottom to Top → 5 cluster → 1 cluster
  Divisive → Top to Down → 1 cluster → 5 cluster

* Representation cluster : Dendrogram

Distance between a point → Point Matrix

we can calculate distance from these data's with the help of Euclidean distance

With the help of point matrix we are going to find out cluster

	x_1 (Height)	x_2 (weight)
P_1	180	75
P_2	170	60
P_3	150	65
P_4	160	90
P_5	140	85

we have to calculate a distance from 1 point another point

Randomly putting values

	P_1	P_2	P_3	P_4	P_5
P_1	O_{P_1,P_1}			upper value will be Redundent	
P_2	$9_{P_2,P_1}$	O_{P_2,P_2}			
P_3	2 P_3,P_1	$7_{P_3,P_2}$	O_{P_3,P_3}		
P_4	$6_{P_4,P_1}$	$5_{P_4,P_2}$	9	O_{P_4,P_4}	
P_5	11	10	3	8	0

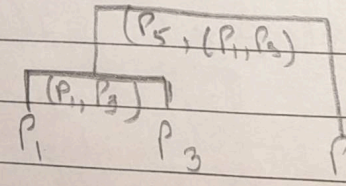
No need to put value of the next it will be similar only

heatmap

P → Points

Diagonal will be zero

Step#1 → least distance in P_1-clm is → 2



Step2 ⇒ Now we are going to reduce this cluster (P_1, P_3)

	(P_1, P_3)	P_2	P_4	P_5
(P_1, P_3)	0			
P_2	7	0		
P_4	6	5	0	
P_5	3 minimum	10	8	0

$P_5 \longleftrightarrow P_4$ { we can Easily calculate this point }

But,

$P_5 \longleftrightarrow (P_1, P_3)$ =? How we can calculate distance

$$\frac{d(P_5, (P_1, P_3))}{d((P_5, P_1), (P_5, P_3))} \longleftarrow \{\text{Single Linkage Method}$$

$$\underset{(11)}{P_5, P_1}) \quad \underset{(3)}{P_5, P_3})$$

$[d(11, 3)] \longleftarrow 3$ is minimum distance

$$\frac{d(P_2, (P_1, P_3))}{d((P_2, P_4), (P_2, P_3))}$$

$[d(9, 7)] \longleftarrow 7$ is minimum

$$\frac{d(P_4, (P_1, P_3))}{d((P_4, P_1), (P_4, P_3))}$$

$[d(6, 9)]$

Now, least distance in (P_1, P_3) - clm $\longrightarrow$ is $\boxed{3}$

dense ↔ sparse

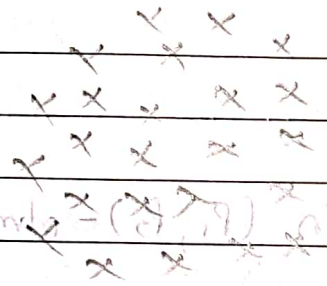
* DBSCAN ⇒ Clustering [(density-based)(spatial clustering) of (application with noise)] ← Noise ≈ outlier

full form

→ It is a soft clustering method to incorporate noisy data (outlier) based on density.

- DBSCAN :- 1) Epsilon Distance
  2) No. of points
  3) Core point
  4) Boundary point
  5) Noise Point (i.e. Outlier)

Explanation ⇒ whenever we are talking about clustering these data has been given. (Drawing some randomly data point over here)



{ I have to create a cluster with this data

Q) How we can create a cluster?

→ Over here we have some requirements

① Epsilon distance
② No. of points → These both are Hyperparameter tuning (we can select different Epsilon value and No. of points)

{ we can select a optimize value for both

→ we can select Epsilon distance → {1, 2, 3, 4, 5 ... n}

→ we can select No. of points → {1, 2, 3, 4, 5 ... n}

Epsilon Distance is also called Radius → for circle

Imp ⇒ # It is used to finding out the outliers

→ Suppose, I have Epsion = 1 and (No. of Points = 4)

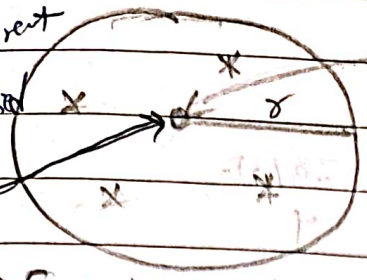
(This is minimum requirement)

we can say this my (centroid)

However this requirement is going to be satisfied we can say

This is a core point

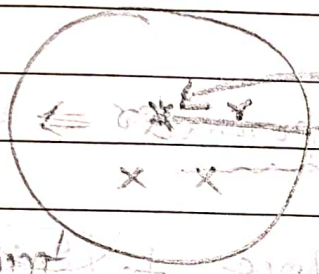
This is my point should be inside circle



Total → 5 points

#Core point is responsible for creating a cluster

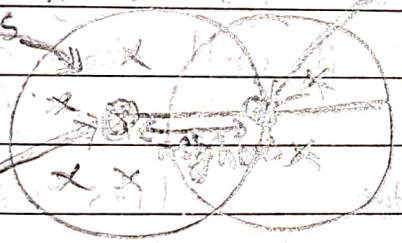
Core Point → If we are taking any random point and we are taking Epsion distance (i.e radius) and we are drawing one circle, and it is fulfilling the requirement



Now it is Boundary point (or) Noise point

Total = 3 point

Inside this one my conditions is going to be satisfied

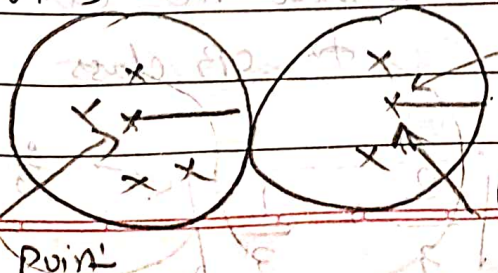


Boundary point (Circle in grey circle) #Now it is Boundary point Inside the cluster

This is clearly visible this is my Core point

Condition for Boundary Point :- The point is called as boundary point if that point comes inside the circle where core point is available. (or) If, if the point is neighbour Core point. then we can say it is Boundary point

Noise Point (Outlier)



This is not a core point and not a Boundary point

Noise point

Core Point