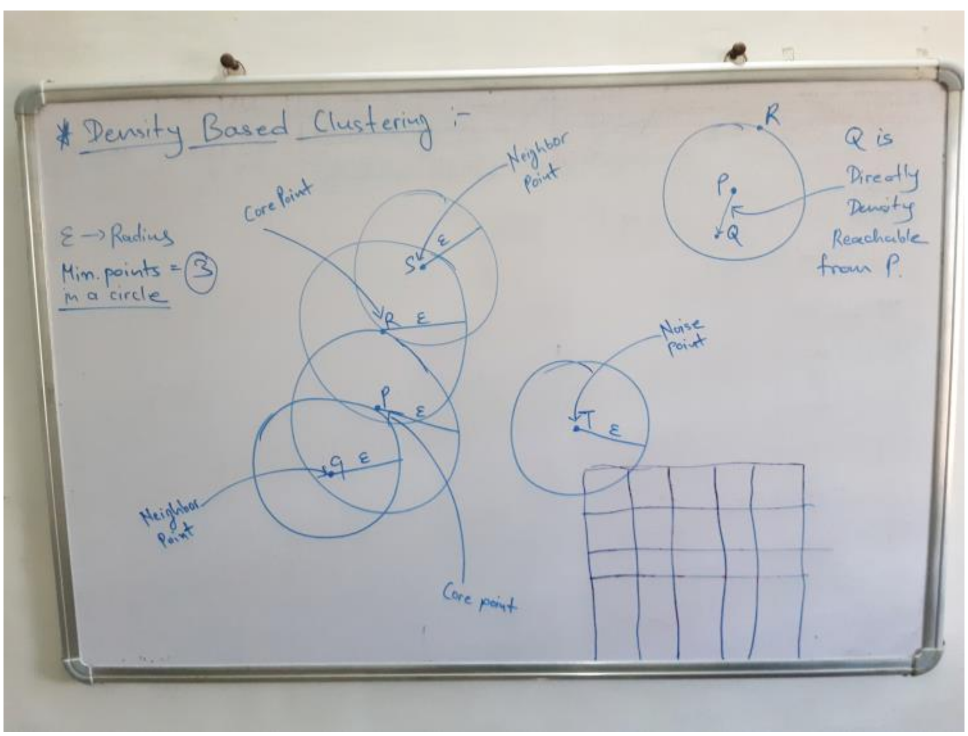
Unsupervised Learning

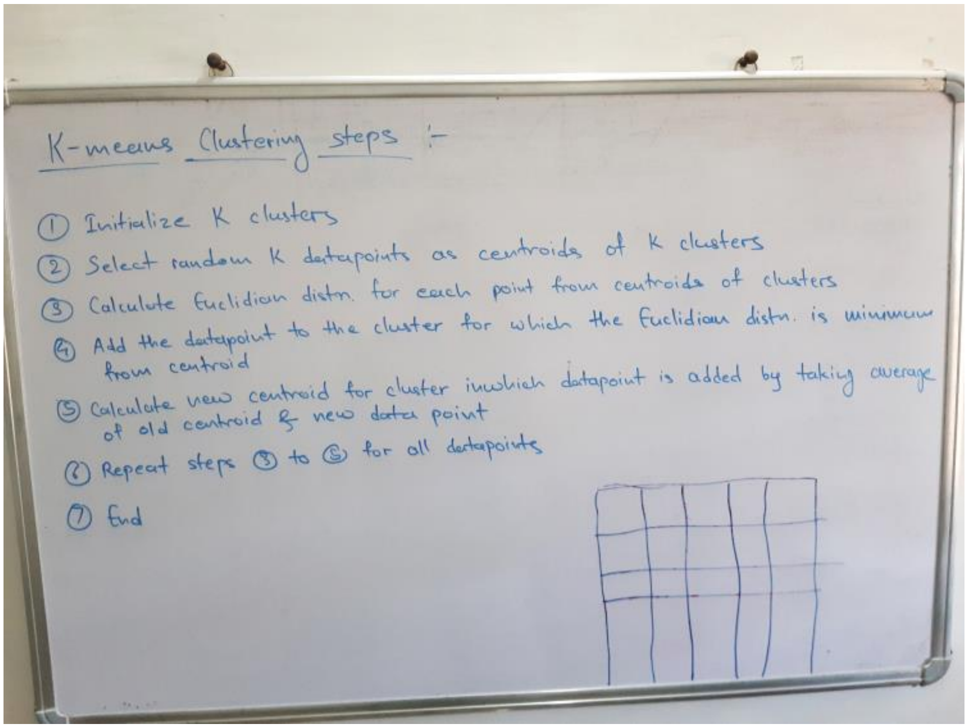
Monday, December 19, 2022 11:27 AM

Density based clustering -

- It is a unsupervised learning approach to form clusters of data point
- It forms clusters on the basis of density of data points
- Those point which are far from density clusters are treated as outliers
- This outliers are called as Noise points
- The two factors are considered as input,
  1.Epsilon
  2.Minimum points
- Epsilon is simply the radius of the circle drawn from a data point as center
- Minimum points are number of points that should be in the circle of a point treated as center
- If min points are present in the circle of a point then it is called as Core Point
- The points present in the circle are called as Neighbor Points
- If a point is neighbor of core point then it is called as Directly Density Reachable
- DBC is very robust in finding outliers and forming clusters
- How clusters are made ?
  1.All the data points are visited one by one
  2.If a point is Core point then it is marked as visited
  3.If it is a neighbor point then also marked as visited
  4.Noise points are ignored
  5.If two points are core points and neighbor of each other then they are added in the same cluster
  6.Cluster is made around core points



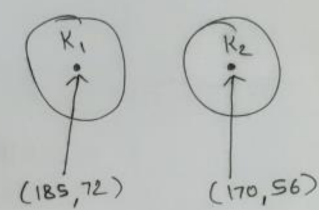
K Means Clustering -


K-means Clustering steps :-
1 Initialize K clusters
2 Select random K datapoints as centroids of K clusters
3 Calculate Euclidian distn. for each point from centroids of clusters
4 Add the datapoint to the cluster for which the Euclidian distn. is minimum from centroid
5 Calculate new centroid for cluster inwhich datapoint is added by taking average of old centroid & new data point
6 Repeat steps 3 to 5 for all datapoints
7 End

* K-means clustering Example :-

No.	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76

Consider, points 1 & 2 as centroid of clusters K_1 & K_2



Euclidian distn.,

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

for pt. 3,

$$K_1 \longrightarrow \sqrt{(168-185)^2 + (60-72)^2} = 20.80$$

$$K_2 \longrightarrow \sqrt{(168-170)^2 + (60-56)^2} = 4.48 \checkmark$$

Distn. of 3 is min from K_2

$\therefore$ $K_1 = \{1\}$

$K_2 = \{2, 3\}$

New centroid for K_2,

$$= \left(\frac{168+170}{2}, \frac{60+56}{2}\right) = (169, 58)$$

$$\left.\begin{array}{l} \therefore K_1 = (185, 72) \\ \therefore K_2 = (169, 58) \end{array}\right\} \text{Centroids}$$

for pt. 4,

$K_1 = \sqrt{(179-185)^2 + (68-72)^2} = \sqrt{36+16} = 7.2$ ✓

$K_2 = \sqrt{(179-169)^2 + (68-58)^2} = \sqrt{100+100} = 14.14$

point 4 is near to K_1

$\therefore K_1 = \{1,4\}$

$\therefore K_2 = \{2,3\}$

$\therefore$ New centroid for $K_1 = \left(\frac{179+185}{2}, \frac{68+72}{2}\right)$

$K_1 = (182, 70)$

$$\left.\begin{aligned} \therefore K_1 &= (182, 70) \\ K_2 &= (169, 58) \end{aligned}\right\} \text{Centroids}$$

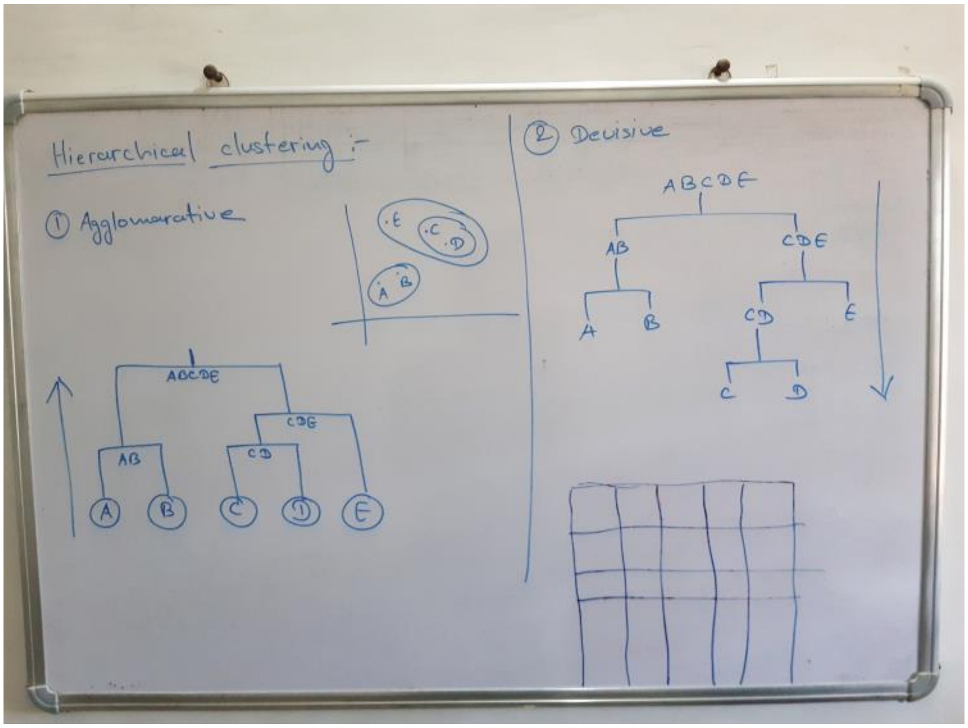
Similarly calculate for other points,

$\therefore K_1 = \{1,4,5,6,7,8,9,10,11,12\}$

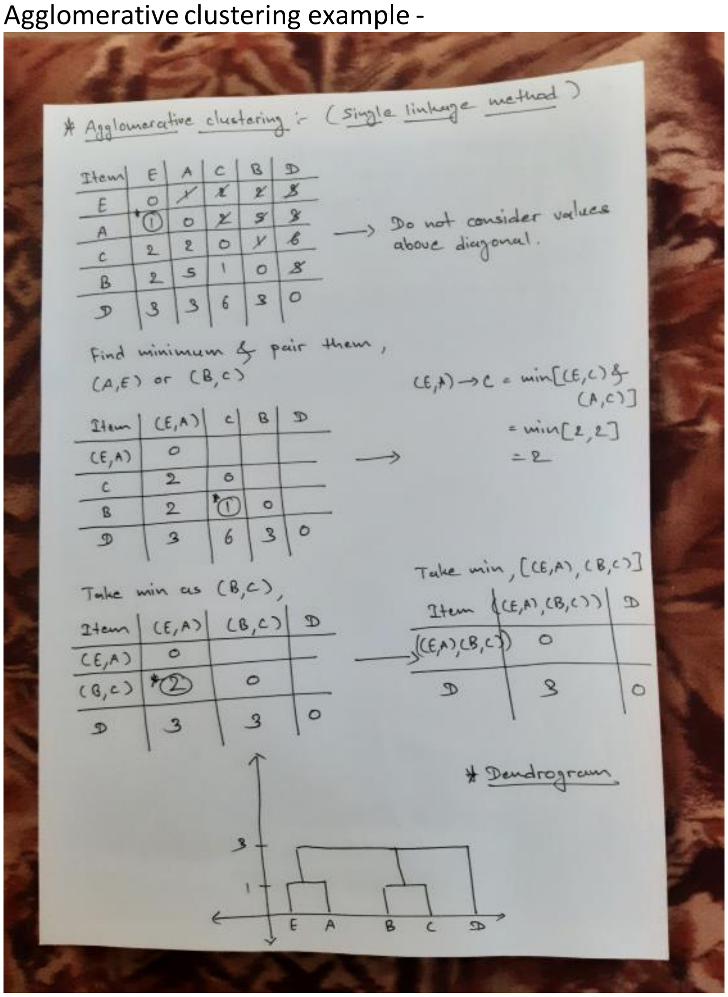
$\therefore K_2 = \{2,3\}$

Hierarchical clustering -

- Clustering is the process of grouping data which have similarities between them
- We create cluster of similar data points
- Hierarchical clustering is type of clustering which is a unsupervised learning method
- There are two types of it,
  1.Agglomerative clustering
  2.Devisive clustering
- In agglomerative we treat each single data point as cluster
- Then according to similarities between them we merge these clusters
- We merge all the clusters or data points until a final cluster containing all cluster is formed
- This is a bottom up approach
- In divisive we have a single big cluster in beginning
- We then divide it into small clusters until we get separate clusters
- This is a top down approach.



Agglomerative clustering example -



* Agglomerative clustering :- (Single linkage method)

Item	E	A	C	B	D
E	0	~~1~~	~~2~~	~~2~~	~~3~~
A	1	0	~~2~~	~~5~~	~~3~~
C	2	2	0	~~1~~	~~6~~
B	2	5	1	0	~~3~~
D	3	3	6	3	0

⟶ Do not consider values above diagonal.

Find minimum & pair them,
(A,E) or (B,C)

Item	(E,A)	C	B	D
(E,A)	0			
C	2	0		
B	2	1	0	
D	3	6	3	0

⟶ (E,A) ⟶ C = min[(E,C) & (A,C)]
= min[2,2]
= 2

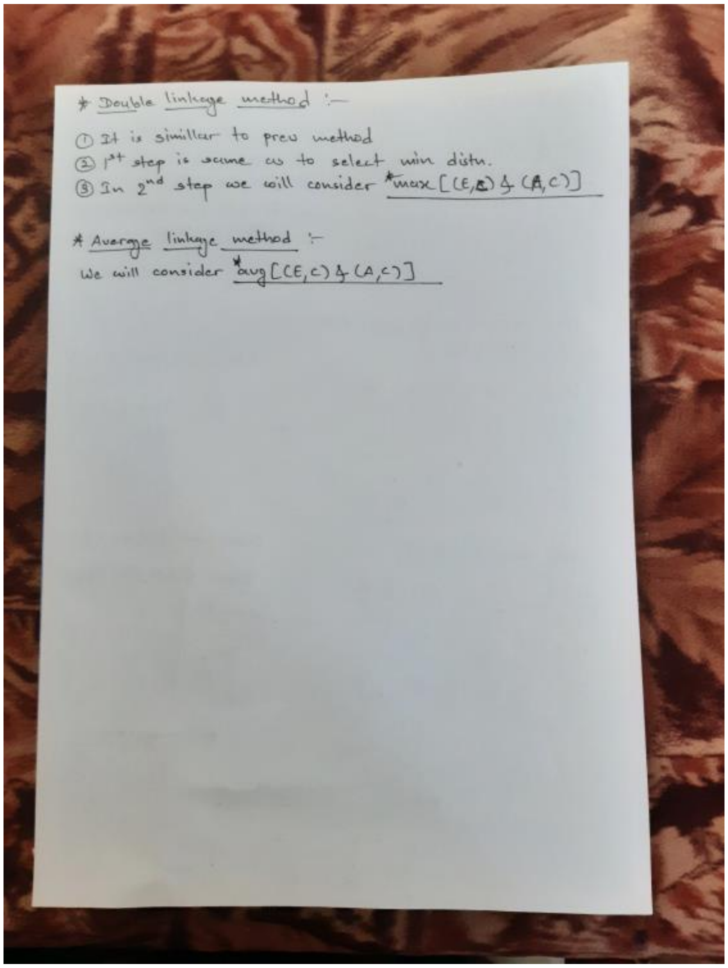
Take min as (B,C),

Item	(E,A)	(B,C)	D
(E,A)	0		
(B,C)	2	0	
D	3	3	0

⟶ Take min, [(E,A), (B,C)]

Item	((E,A),(B,C))	D
((E,A),(B,C))	0	
D	3	0

* Dendrogram



* Double linkage method :-

(1) It is simillar to prev method
(2) 1st step is same as to select min distn.
(3) In 2nd step we will consider *max [(E,C) & (A,C)]

* Average linkage method :-
We will consider *avg [(E,C) & (A,C)]

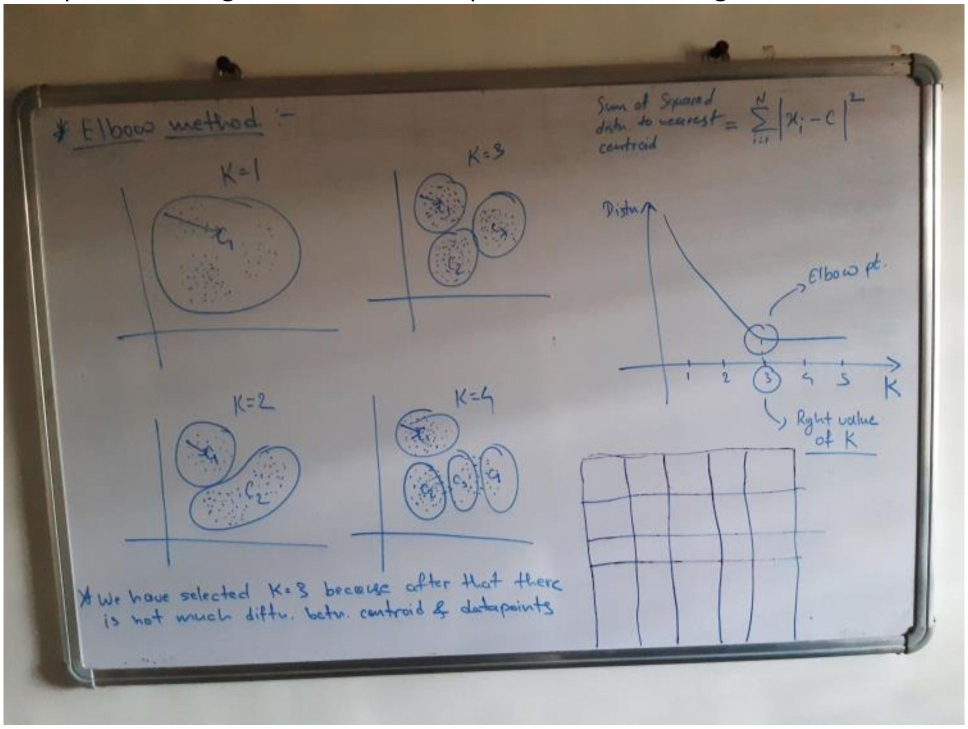
K-medoids clustering -

- Similar to the K-means clustering
- In in here instead of centroids we select medoids
- A medoid is a data point which is most similar to other data points
- Then we partition the dataset using medoids and form clusters
- Steps -
  1.Select K medoids
  2.Calculate distance between each datapoint and medoid using, |x2-x1|+|y2-y1|
  3.Compare these distances of medoids from datapoints and select minimum distances
  4.According to minimum distances group the points in cluster
  5.Again select another medoid and repeat steps 2 to 4
  6.Compare total cost of both iterations
  7.If we get positive difference then terminate else again repeat.

Elbow method -

- This method is used to determine right K value for K-means clustering
- For that we take 2 to 3 values of K for a dataset
- We calculate Sum of Square distances to nearest centroid
- It means that for each point in cluster we calculate its distance from centroid and square it
- We do this for all data points and sum them up
- As we increase value of K the sum value decreases
- In beginning the value decreases drastically but after some point it decreases gradually

- That point of change is called as Elbow point and taken as right value of K for a dataset



Extrinsic method -

- It is method used to evaluate clustering algo
- In this we use clustering for to be used in some other model
- Then we evaluate performance of that model and from that we got to know whether our clustering is good or not

Intrinsic method -

- In this we try to evaluate cluster itself
- This can be done by treating clusters as classes or by manual check the clusters