K-Mean Algorithm:-

Step: 1 Take Mean Value

step:-2 Find the Nearest number to mean and put it in the Cluster.

Step:-3 Repeat steps ① & ② Untill we ge same Mean.

Example:-

$S = \{2, 3, 4, 10, 11, 12, 20, 25, 30\}$

→ Divide the set into 2 groups (i,e) $k = 2$

→ Take any 2 random Number from the set S

(i,e) $m_1 = 4$ & $m_2 = 12$

Now, $K_1 = \{2, 3, 4\}$ $K_2 = \{10, 11, 12, 20, 25, 30\}$

$m_1 = \frac{2+3+4}{3} = 3$ $m_2 = \frac{108}{6} = 18$

$\therefore m_1 = 3$ & $m_2 = 18$

Repeat the process untill we get same means

$K_1 = \{2, 3, 4, 10\}$ $K_2 = \{11, 12, 20, 25, 30\}$

$m_1 \simeq 5$ $m_2 \simeq 20$

$K_1 = \{2, 3, 4, 10, 11, 12\}$ $K_2 = \{20, 25, 30\}$

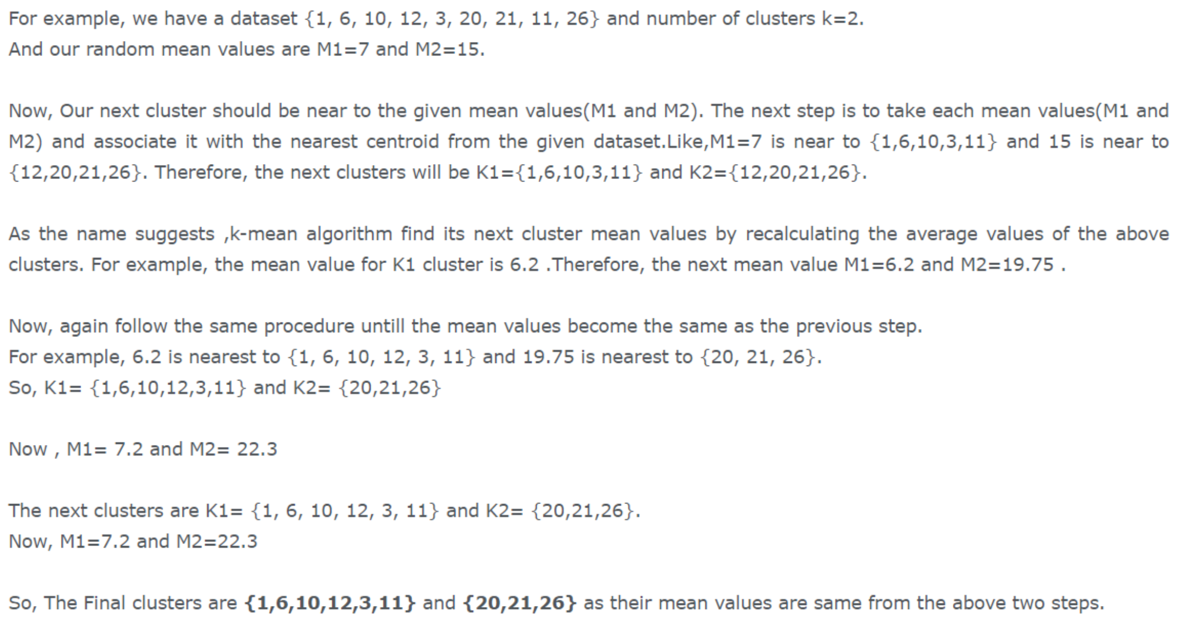
$m_1 = 7$ $m_2 = 25$

Again, $K_1 = \{2, 3, 4, 10, 11, 12\}$ $K_2 = \{20, 25, 30\}$

$m_1 = 7$ $m_2 = 25$

Thus we get same mean values,

Finaly we form two clusters $K_1 = \{2, 3, 4, 10, 11, 12\}$ & $K_2 = \{20, 25, 30\}$ from set S.



For example, we have a dataset {1, 6, 10, 12, 3, 20, 21, 11, 26} and number of clusters k=2.
And our random mean values are M1=7 and M2=15.

Now, Our next cluster should be near to the given mean values(M1 and M2). The next step is to take each mean values(M1 and M2) and associate it with the nearest centroid from the given dataset.Like,M1=7 is near to {1,6,10,3,11} and 15 is near to {12,20,21,26}. Therefore, the next clusters will be K1={1,6,10,3,11} and K2={12,20,21,26}.

As the name suggests ,k-mean algorithm find its next cluster mean values by recalculating the average values of the above clusters. For example, the mean value for K1 cluster is 6.2 .Therefore, the next mean value M1=6.2 and M2=19.75 .

Now, again follow the same procedure untill the mean values become the same as the previous step.
For example, 6.2 is nearest to {1, 6, 10, 12, 3, 11} and 19.75 is nearest to {20, 21, 26}.
So, K1= {1,6,10,12,3,11} and K2= {20,21,26}

Now , M1= 7.2 and M2= 22.3

The next clusters are K1= {1, 6, 10, 12, 3, 11} and K2= {20,21,26}.
Now, M1=7.2 and M2=22.3

So, The Final clusters are **{1,6,10,12,3,11}** and **{20,21,26}** as their mean values are same from the above two steps.

1. **K means Problem**

consider the following data set consisting of the scores of two variables on each of seven individuals. Apply k-means algorithm and solve

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Solution

This data set is to be grouped into two clusters. As a first step in finding a sensible initial partition, let the A & B values of the two individuals furthest apart (using the Euclidean distance measure), define the initial cluster means, giving:

	Individual	Mean Vector (centroid)
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

The remaining individuals are now examined in sequence and allocated to the cluster to which they are closest, in terms of Euclidean distance to the cluster mean. The mean vector is recalculated each time a new member is added. This leads to the following series of steps:

	Cluster 1		Cluster 2	
Step	Individual	Mean Vector (centroid)	Individual	Mean Vector (centroid)
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1, 2	(1.2, 1.5)	4	(5.0, 7.0)
3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
4	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
5	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

Now the initial partition has changed, and the two clusters at this stage having the following characteristics:

	Individual	Mean Vector (centroid)
Cluster 1	1, 2, 3	(1.8, 2.3)

Cluster 2	4, 5, 6, 7	(4.1, 5.4)

But we cannot yet be sure that each individual has been assigned to the right cluster. So, we compare each individual’s distance to its own cluster mean and to that of the opposite cluster. And we find:

Individual	Distance to mean (centroid) of Cluster 1	Distance to mean (centroid) of Cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

Only individual 3 is nearer to the mean of the opposite cluster (Cluster 2) than its own (Cluster 1). In other words, each individual's distance to its own cluster mean should be smaller that the distance to the other cluster's mean (which is not the case with individual 3). Thus, individual 3 is relocated to Cluster 2 resulting in the new partition:

	Individual	Mean Vector (centroid)
Cluster 1	1, 2	(1.3, 1.5)
Cluster 2	3, 4, 5, 6, 7	(3.9, 5.1)

The iterative relocation would now continue from this new partition until no more relocations occur. However, in this example each individual is now nearer its own cluster mean than that of the other cluster and the iteration stops, choosing the latest partitioning as the final cluster solution.

Also, it is possible that the k-means algorithm won't find a final solution. In this case it would be a good idea to consider stopping the algorithm after a pre-chosen maximum of iterations.

2. Suppose we want to group the visitors to a website using just their age (a one-dimensional space) as follows:

15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65

Solution

Initial clusters:

Centroid (C1) = 16 [16]
Centroid (C2) = 22 [22]

Iteration **1**:

C1 = 15.33 [15,15,16]
C2 = 36.25 [19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65]

Iteration **2**:

C1 = 18.56 [15,15,16,19,19,20,20,21,22]
C2 = 45.90 [28,35,40,41,42,43,44,60,61,65]

Iteration **3**:

C1 = 19.50 [15,15,16,19,19,20,20,21,22,28]
C2 = 47.89 [35,40,41,42,43,44,60,61,65]

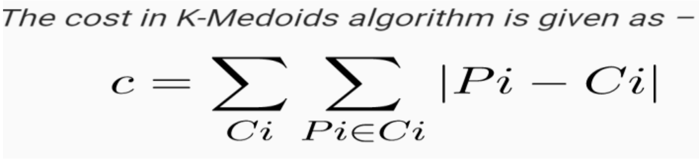
Iteration **4**:

C1 = 19.50 [15,15,16,19,19,20,20,21,22,28]
C2 = 47.89 [35,40,41,42,43,44,60,61,65]

No change between iterations 3 and 4 has been noted. By using clustering, 2 groups have been identified 15-28 and 35-65. The initial choice of centroids can affect output clusters, so the algorithm is often run multiple times with different starting conditions in order to get a fair view of what the clusters should be.

K-Medoids (also called as Partitioning Around Medoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw. **A medoid can be defined as the point in the cluster, whose dissimilarities with all the ot points in the cluster is minimum.**

The dissimilarity of the `medoid(Ci)` and `object(Pi)` is calculated by using **`E = |Pi - Ci|.`**

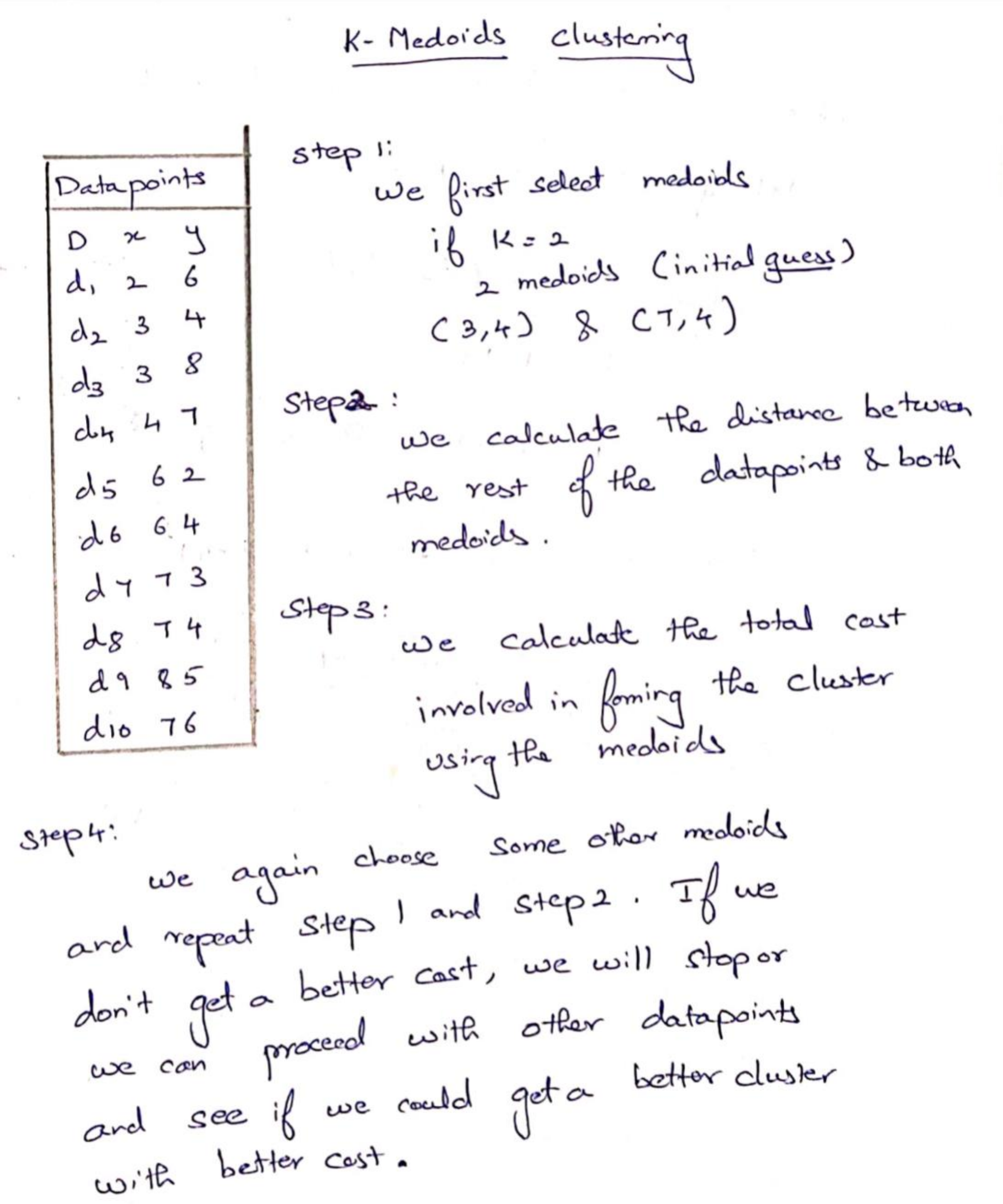


The cost in K-Medoids algorithm is given as –

$$c = \sum_{Ci} \sum_{Pi \in Ci} |Pi - Ci|$$

ALGORITHM

```
1. Initialize: select k random points out of the n data points as the medoids.
2. Associate each data point to the closest medoid by using any common distance metric methods.
3. While the cost decreases:
        For each medoid m, for each data o point which is not a medoid:
                1. Swap m and o, associate each data point to the closest medoid, recompute the cost.
                2. If the total cost is more than that in the previous step, undo the swap.
```



K- Medoids clustering

Datapoints		
D	x	y
d_1	2	6
d_2	3	4
d_3	3	8
d_4	4	7
d_5	6	2
d_6	6	4
d_7	7	3
d_8	7	4
d_9	8	5
d_{10}	7	6

step 1:
we first select medoids
if K = 2
2 medoids (initial guess)
(3,4) & (7,4)

Step2 :
we calculate the distance between the rest of the datapoints & both medoids.

Step 3:
we calculate the total cost involved in forming the cluster using the medoids

step 4:
we again choose some other medoids and repeat step 1 and step 2. If we don't get a better cost, we will stop or we can proceed with other datapoints and see if we could get a better cluster with better cost.

$= |x_1 - x_2| + |y_1 - y_2|$

D	x	y	Distance from (3,4)	Distance from (7,4)
d_1	2	6	$\|2-3\| + \|6-4\| = \underline{3}$	$\|2-7\| + \|6-4\| = 7$
d_3	3	8	$\|3-3\| + \|8-4\| = \underline{4}$	$\|3-7\| + \|8-4\| = 8$
d_4	4	7	$\|4-3\| + \|7-4\| = \underline{4}$	$\|4-7\| + \|7-4\| = 6$
d_5	6	2	$\|6-3\| + \|2-4\| = 5$	$\|6-7\| + \|2-4\| = \underline{3}$
d_6	6	4	$\|6-3\| + \|4-4\| = 3$	$\|6-7\| + \|4-4\| = \underline{1}$
d_7	7	3	$\|7-3\| + \|3-4\| = 5$	$\|7-7\| + \|3-4\| = \underline{1}$
d_9	8	5	$\|8-3\| + \|5-4\| = 6$	$\|8-7\| + \|5-4\| = \underline{2}$
d_{10}	7	6	$\|7-3\| + \|6-4\| = 6$	$\|7-7\| + \|6-4\| = \underline{2}$

Note: d_2 & d_8 not used because they are same as initial guess.

Total cost = Σ (min values of last two columns)

$= 3+4+4+3+1+1+2+2$

$= 20$

cluster {3,4} medoid = { {3,4}, {2,6}, {4,7}, {3,8} }

{7,4} medoid = { {7,4}, {7,3} {6,2} {6,4} {8,5} {7,6} }

we will choose some other medoid
(7,3)

we will recalculate the distane between each of data points and the current medoids.
(7,3) (3,4)

D	x	y	Distance from (3,4)	Distance from (7,3)
d_1	2	6	3	$\|2-7\| + \|6-3\| = 8$
d_3	3	8	4	$\|3-7\| + \|8-3\| = 9$
d_4	4	7	4	$\|4-7\| + \|7-3\| = 7$
d_5	6	2	5	$\|6-7\| + \|2-3\| = 2$
d_6	6	4	3	$\|6-7\| + \|4-3\| = 2$
d_8	7	4	4	$\|7-7\| + \|4-3\| = 1$
d_9	8	5	6	$\|8-7\| + \|5-3\| = 3$
d_{10}	7	6	6	$\|7-7 + \|6-3\| = 3$

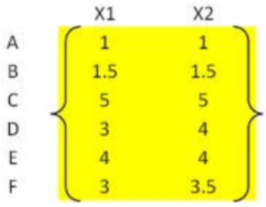
Total cost = 3+4+4+2+2+1+3+3

= 22 > 20 (Previous cluster cust)

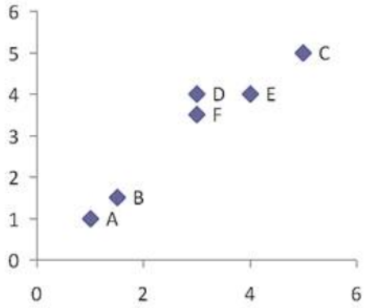
Clusters

$C_1 = \{(3,4), (2,6)\ (3,8)\ (4,7)\}$

$C_2 = \{(7,4), (7,3), (6,2), (6,4), (8,5), (7,6)\}$



	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



The proximity between object can be measured as distance matrix. Suppose we use Euclidean distance , we can compute the distance between objects using the following formula

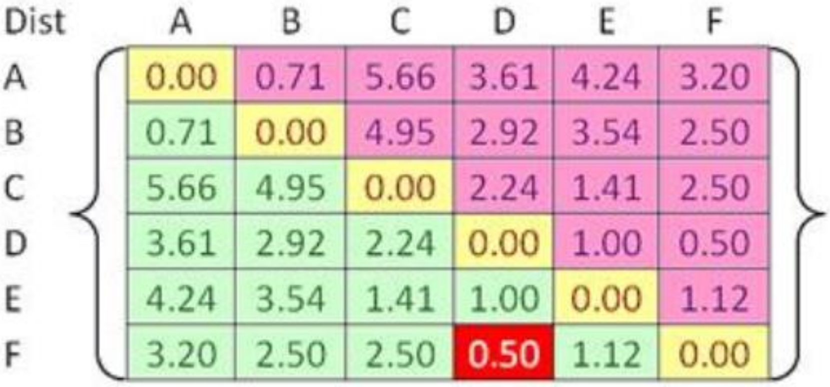
$$d_{ij} = \left(\sum_k \left(x_{ik} - x_{jk}\right)^2\right)^{\frac{1}{2}}$$

For example, distance between object A = (1, 1) and B = (1.5, 1.5) is computed as

$$d_{AB} = \left((1-1.5)^2 + (1-1.5)^2\right)^{\frac{1}{2}} = \sqrt{\tfrac{1}{2}} = 0.7071$$

Another example of distance between object D = (3, 4) and F = (3, 3.5) is calculated as

$$d_{DF} = \left((3-3)^2 + (4-3.5)^2\right)^{\frac{1}{2}} = 0.5$$

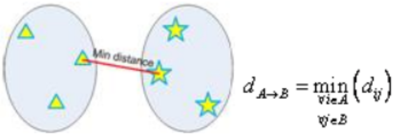


Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Linkages Between Objects

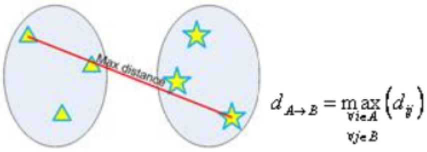
The rule of hierarchical clustering lie on how objects should be grouped into clusters. Given a distance matrix, linkages between objects can be computed through a criterion to compute distance between groups. Most common & basic criteria are

Single Linkage: minimum distance criterion



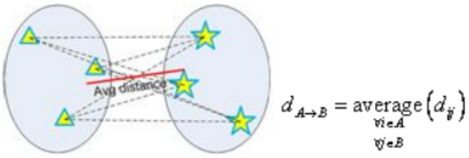
$$d_{A\to B} = \min_{\substack{\forall i \in A \\ \forall j \in B}} \left(d_{ij}\right)$$

Complete Linkage: maximum distance criterion



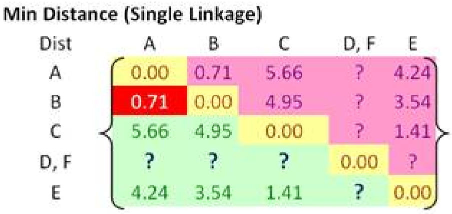
$$d_{A\to B} = \max_{\substack{\forall i \in A \\ \forall j \in B}} \left(d_{ij}\right)$$

Average Group: average distance criterion



$$d_{A\to B} = \underset{\substack{\forall i \in A \\ \forall j \in B}}{\text{average}} \left(d_{ij}\right)$$

In each step of the iteration, we find the closest pair clusters. In this case, the closest cluster is between cluster F and D with shortest distance of 0.5. Thus, we group cluster D and F into cluster (D, F). Then we update the distance matrix (see distance matrix below). Distance between ungrouped clusters will not change from the original distance matrix. Now the problem is how to calculate distance between newly grouped clusters (D, F) and other clusters?



Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

That is exactly where the linkage rule comes into effect. Using single linkage, we specify minimum distance between original objects of the two clusters.

Using the input distance matrix, distance between cluster (D, F) and cluster A is computed as

$$d_{(D,F)\mapsto A} = \min\left(d_{DA}, d_{FA}\right) = \min\left(3.61, 3.20\right) = 3.20$$

Distance between cluster (D, F) and cluster B is

$$d_{(D,F)\mapsto B} = \min\left(d_{DB}, d_{FB}\right) = \min\left(2.92, 2.50\right) = 2.50$$

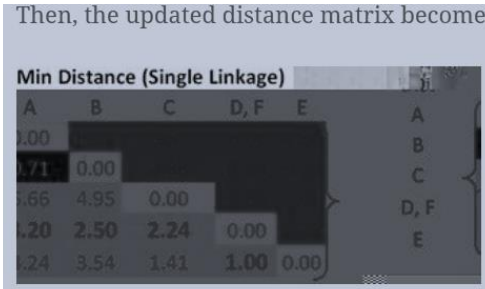
Similarly, distance between cluster (D, F) and cluster C is

$$d_{(D,F)\mapsto C} = \min\left(d_{DC}, d_{FC}\right) = \min\left(2.24, 2.50\right) = 2.24$$

Finally, distance between cluster E and cluster (D, F) is calculated as

$$d_{E\to(D,F)} = \min\left(d_{ED}, d_{EF}\right) = \min\left(1.00, 1.12\right) = 1.00$$

Then, the updated distance matrix becomes

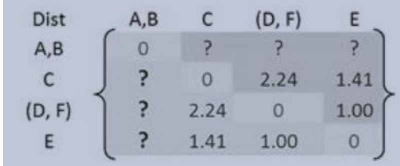


Min Distance (Single Linkage)

A	B	C	D, F	E	
.00					A
.71	0.00				B
5.66	4.95	0.00			C
.20	2.50	2.24	0.00		D, F
.24	3.54	1.41	1.00	0.00	E

Looking at the lower triangular updated distance matrix, we found out that the closest distance between cluster B and cluster A is now 0.71. Thus, we group cluster A and cluster B into a single cluster name (A, B).

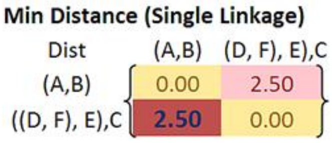
Now we update the distance matrix. Aside from the first row and first column, all the other elements of the new distance matrix are not changed.



Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

Using the input distance matrix (size 6 by 6), distance between cluster C and cluster (D, F) is computed as $d_{C\to(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$

The updated distance matrix is shown in the figure below



Min Distance (Single Linkage)

Dist	(A,B)	(D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	2.50	0.00

The minimum distance of 2.5 is the result of the following computation

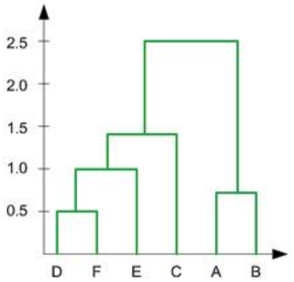
$$d_{(((D,F),E),C)\rightarrow(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}, d_{EA}, d_{EB}, d_{CA}, d_{CB}\right)$$

$$d_{(((D,F),E),C)\rightarrow(A,B)} = \min\left(3.61, 2.92, 3.20, 2.50, 4.24, 3.54, 5.66, 4.95\right) = 2.50$$

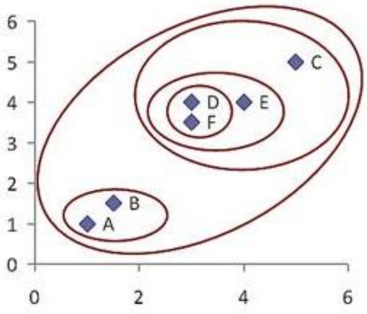
Now if we merge the remaining two clusters, we will get only single cluster contain the whole 6 objects. Thus, our computation is finished. We summarized the results of computation as follow:

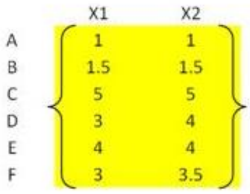
1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge cluster D and F into cluster (D, F) at distance **0.50**
3. We merge cluster A and cluster B into (A, B) at distance **0.71**
4. We merge cluster E and (D, F) into ((D, F), E) at distance **1.00**
5. We merge cluster ((D, F), E) and C into (((D, F), E), C) at distance **1.41**
6. We merge cluster (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance **2.50**
7. The last cluster contain all the objects, thus conclude the computation

Using this information, we can now draw the final results of a dendogram. The dendogram is drawn based on the distances to merge the clusters above.



The hierarchy is given as (((D, F), E),C), (A,B). We can also plot the clustering hierarchy into XY space





	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

A	B	C	D	E	
A	0	2	1	3	4
B	2	0	3	6	5
C	1	3	0	7	5
D	3	6	7	0	4

E	4	5	5	4	0

The nearest pair is A and C as they have the smallest distance, at distance 1.The level of the new cluster is 1 and they are merged into a single cluster called "AC"

Then, we compute the distance from this new compound object to all other objects.

In Single Link Clustering, the rule is to find the shortest distance object and merge them together. Moreover, to find the distance from the compound object to another object is equal to the shortest distance from any member of the cluster to another object.

Note: The distance from the same object will be 0.Therefore, distance from A-A is 0 and so on.

Now, the distance from "AC" to "B" is 2 which is the shortest one. Because, according to the Single Link Clustering rule, the distance from A-B is 2 and C-B is 3. Therefore, 2 will be the distance between AC-B. The same procedure applies for the rest of the objects.

After merging C with A, the new Cluster is called AC, at distance 1.

	AC	B	D	E
AC	0	2	3	4
B	2	0	6	5
D	3	6	0	4
E	4	5	4	0

Now, the nearest pair is AC and B. Therefore, we apply the same rule as above. So, the new compound cluster will be ABC (Order does not matter).

To find the distance with other objects/clusters are same as above. The distance between AC-D is 3 and the distance from B-D is 6.Therefore, the distance from ABC to D will be considered as 3 and so on.

After merging AC with B, the new cluster is called ABC, at distance 2.

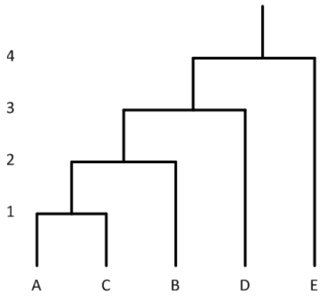
	ABC	D	E
ABC	0	3	4
D	3	0	4
E	4	4	0

The nearest pair of object in above matrix is ABC and D at distance 3.They are merged into a cluster called ABCD.

After merging ABC with D, the cluster is ABCD, at distance 3.

	ABCD	E
ABCD	0	4
E	4	0

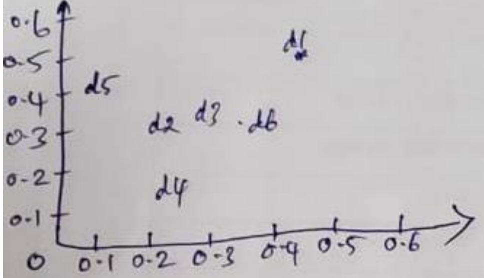
Finally, we merge the last two clusters at level 4.so the final cluster is "ABCDE". This process is summarized by the clustering diagram as below:



Hierarchial Agglomerative Clustering (HAC)

Data set

	x(p)	y(q)
d1	0.4	0.53
d2	0.22	0.38
d3	0.35	0.32
d4	0.26	0.19
d5	0.08	0.41
d6	0.45	0.30



① Find the distance matrix.

Manhattan distance / Euclidean distance

(P_1, P_2) (q_1, q_2)

$|P_1 - q_1) + |P_2 - q_2|$

d1d2 = |0.4 − 0.22| + |0.53 − 0.38| = 0.18 + 0.15 = 0.33

d1d3 = |0.4 − 0.35| + |0.53 − 0.32| = 0.05 + 0.21 = 0.26

d1d4 = |0.4 − 0.26| + |0.53 − 0.19| = 0.14 + 0.34 = 0.45

d1d5 = |0.4 − 0.08| + |0.53 − 0.41| = 0.32 + 0.12 = 0.44

d1d6 = |0.4 − 0.45| + |0.53 − 0.30| = 0.05 + 0.23 = 0.28

	d1	d2	d3	d4	d5	d6
d1	0	0.33	0.26	0.45	0.44	0.28
d2	0.33	0	0.19	0.35	0.01	0.31
d3	0.26	0.19	0	0.22	0.36	0.12
d4	0.45	0.35	0.22	0	0.40	0.30
d5	0.44	0.01	0.36	0.40	0	0.48
d6	0.28	0.31	0.12	0.30	0.48	0

d2d3 = |0.22 − 0.35| + |0.38 − 0.32) = 0.13 + 0.06
= 0.19

d2d4 = |0.22 − 0.38| + |0.38 − 0.19| = 0.16 + 0.19
= 0.35

d3d4 = |0.35 − 0.26| + |0.32 − 0.19| = [illegible]
= 0.09 + 0.13 = 0.22

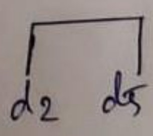
d5d6 = |0.08 − 0.45| + |0.41 − [illegible]01| ⇒ 0.37 + 0.11 ⇒ 0.48
−0.30

② Find the minimum distance

d2d5 = 0.01

d2d5 are closed to each other.

③ update the distance matrix using the first cluster (d2, d5)



methods to update the distance matrix

1) Single Link
2) complete Link
3) Average Link

Single link ; complex link ; Average link

a) Distance between (d2,d5) and d1 using single link ②

$\text{Min}((d2,d1) ; (d5,d1))$

$= \text{Min}((0.33 , 0.44) = 0.33$

Distance between (d2,d5) and d1 using complex link

$= \text{Max}((d2,d1), (d5,d1)$

$= \text{max}(0.33, 0.44) = 0.44$

Distance between (d2,d5) and d1 using Average link

$= \frac{1}{2}(\text{dist}(d2,d1) + \text{dist}(d5,d1))$

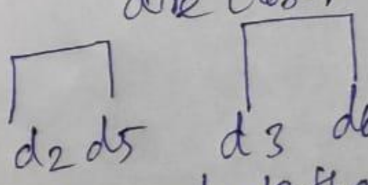
$= \frac{1}{2}(0.33 + 0.44) = \frac{77}{2} \Rightarrow 0.38$

So, choose the single link for the distance matrix updates.

updated matrix (d2,d5) to , d3, d4, d6

	d1	d2d5	d3	d4		d6
d1	0	0.33	0.26	0.35		0.28
d2d5	0.33	0	0.19	0.23		0.31
d3	0.26	0.19	0	0.22		$\boxed{0.12}$
d4	0.35	0.23	0.22	0		0.3
~~d5~~						
d6	0.28	0.31	0.12	0.3		0

d3 & d6 are closer



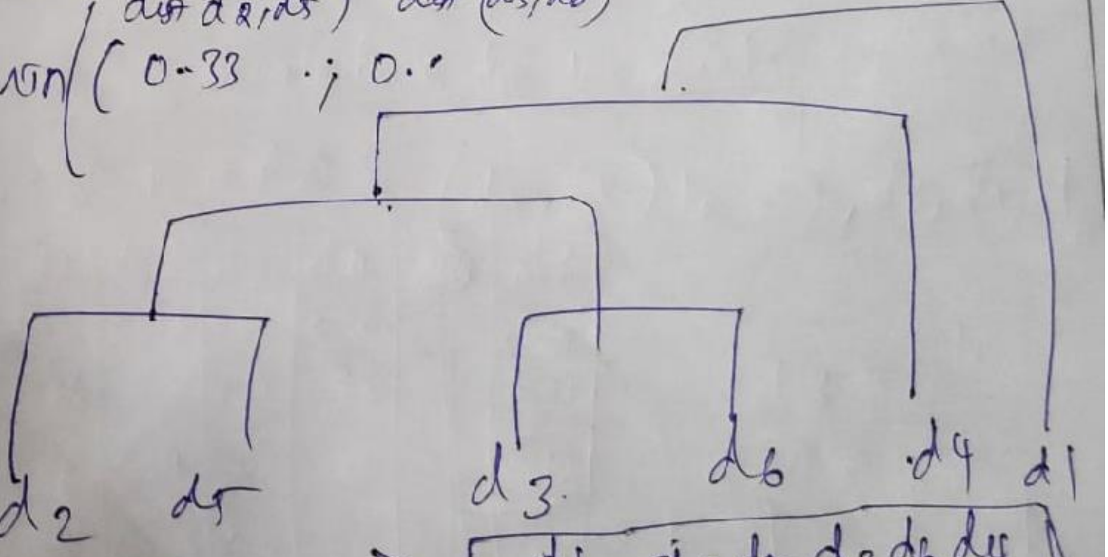
again update the distance matrix

	d1	d2d5	d3d6	d4
d1	0	0.33	0.26	0.35
d2d5	0.33	0	:	
d3d6	0.26		0	
d4				0

d2d5 to d3d6 using single link

dist d2,d5) dist (d3,d6)

Min (0.33 .; 0..



	d1	d2d5 d3d6 d4
d1	0	0.33
d2d5 d3d6 d4	0.33	0

Andrey	0.48	3.33
Weifeng	4.54	0.71

Clusters are

C1 = A, B, D and C2 = C, E | **1.5 marks|**

Since the clustering didn't change we will not compute the clusters again. [1 mark for the answer and justification]

Question 4

a) Cluster the following points { P[2,3], Q[2,4], R[6,5], S[5,8], T[6,3] } using complete linkage hierarchical clustering algorithm. Assume Manhattan distance measure. [5]

b) There are 100 points in a 2-D area. All points are uniformly spaced. What is the expected result from running DBSCAN on this data. [3]

c) Comment on the clustering quality by computing Silhouette coefficient. The distance matrix is given below. Cluster1 contains point A and C and Cluster2 contains points B and D. [3]

	A	B	C	D
A	0	2.0	2.3	3.2
B	2.0	0	2.2	1.4
C	2.3	2.2	0	2.2
D	3.2	1.4	2.2	0

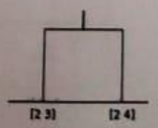
Solution

A. Hierarchical – complete linkage algorithm

1. Manhattan distance matrix (0.5 mark)

	P[2,3]	Q[2,4]	R[6,5]	S[5,8]	T[6,3]
P[2,3]	0	1	6	8	4
Q[2,4]	1	0	5	7	5
R[6,5]	6	5	0	4	2
S[5,8]	8	7	4	0	6
T[6,3]	4	5	2	6	0

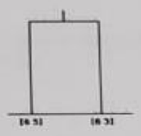
2. Find the minimum distance for within cluster similarity. dmin = 1 for the points P and Q. So group them together and draw dendogram, for the points P and Q at level = 1. (0.5 mark)



3. Re compute the distance matrix using complete linkage distance. (0.5 mark)

	PQ	R	S	T
PQ	0	6	8	5
R	6	0	4	2
S	8	4	0	6
T	5	2	6	0

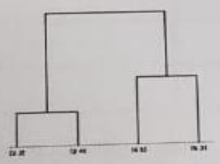
4. Find the minimum distance for within cluster similarity. dmin = 2 for the points R and T. So group them together and draw dendogram, for the points R and T at level = 2. (0.5 mark)



5. Re compute the distance matrix using complete linkage distance. (0.5 mark)

	PQ	RT	S
PQ	0	6	8
RT	6	0	6
S	8	6	0

6. Find the minimum distance for within cluster similarity. dmin = 6 for the points S and RT. So group them together and draw dendogram, for the points S and RT at level = 6. (0.5 mark)



7. Re compute the distance matrix using complete linkage distance. (0.5 mark)

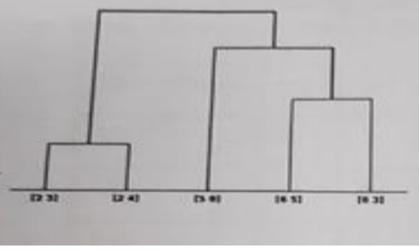
	PQ	RTS
PQ	0	8
RTS	8	0

8. Alternatively group PQ and RT together and draw dendogram, for the points PQ and RT at level = 6.

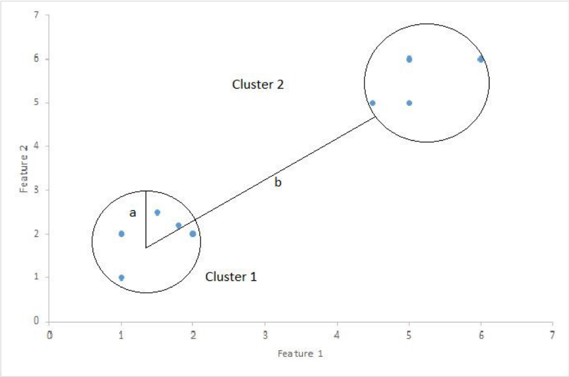
9. Re compute the distance matrix using complete linkage distance

	PQRT	S
PQRT	0	8
S	8	0

10. Draw dendogram, for all points at level = 8. (0.5 mark)

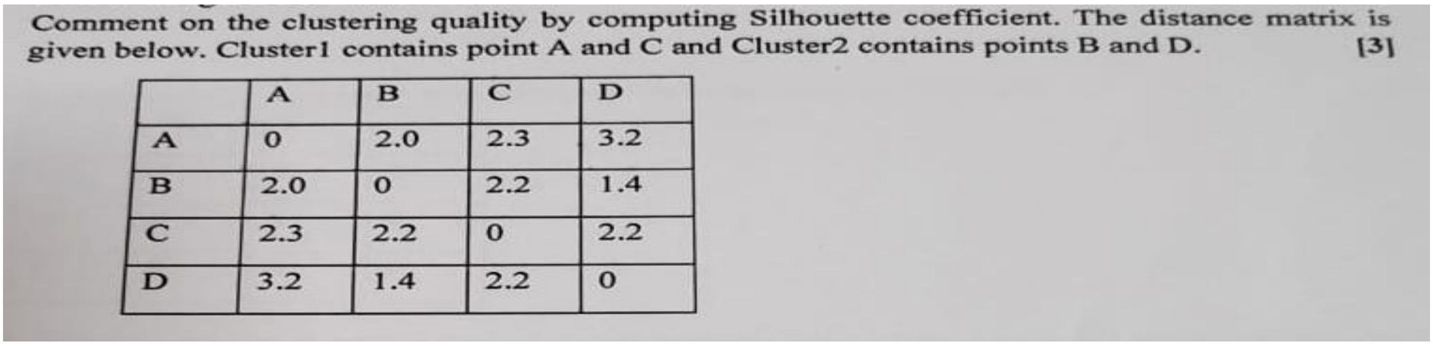


Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.
1: Means clusters are well apart from each other and clearly distinguished.
0: Means clusters are indifferent, or we can say that the distance between clusters is not significant.
-1: Means clusters are assigned in the wrong way.



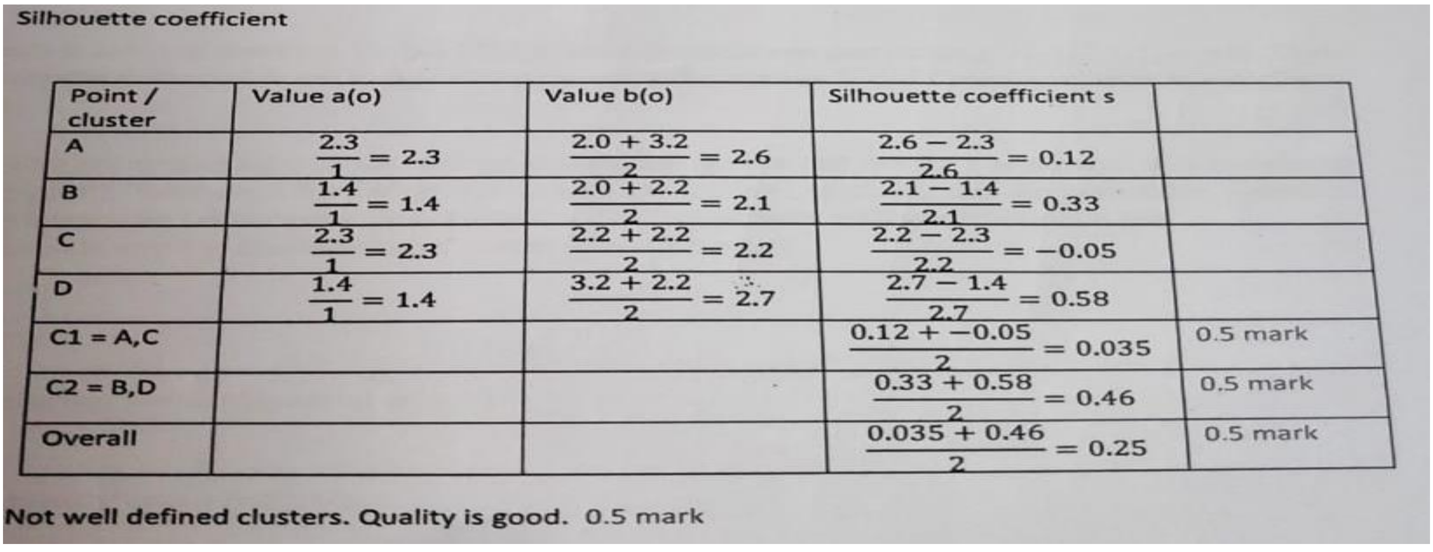
Silhouette Score = (b-a)/max(a,b)

where
a= average intra-cluster distance i.e the average distance between each point within a cluster.
b= average inter-cluster distance i.e the average distance between all clusters.



Comment on the clustering quality by computing Silhouette coefficient. The distance matrix is given below. Cluster1 contains point A and C and Cluster2 contains points B and D. [3]

	A	B	C	D
A	0	2.0	2.3	3.2
B	2.0	0	2.2	1.4
C	2.3	2.2	0	2.2
D	3.2	1.4	2.2	0



Silhouette coefficient

Point / cluster	Value a(o)	Value b(o)	Silhouette coefficient s	
A	$\frac{2.3}{1} = 2.3$	$\frac{2.0 + 3.2}{2} = 2.6$	$\frac{2.6 - 2.3}{2.6} = 0.12$	
B	$\frac{1.4}{1} = 1.4$	$\frac{2.0 + 2.2}{2} = 2.1$	$\frac{2.1 - 1.4}{2.1} = 0.33$	
C	$\frac{2.3}{1} = 2.3$	$\frac{2.2 + 2.2}{2} = 2.2$	$\frac{2.2 - 2.3}{2.2} = -0.05$	
D	$\frac{1.4}{1} = 1.4$	$\frac{3.2 + 2.2}{2} = 2.7$	$\frac{2.7 - 1.4}{2.7} = 0.58$	
C1 = A,C			$\frac{0.12 + -0.05}{2} = 0.035$	0.5 mark
C2 = B,D			$\frac{0.33 + 0.58}{2} = 0.46$	0.5 mark
Overall			$\frac{0.035 + 0.46}{2} = 0.25$	0.5 mark

Not well defined clusters. Quality is good. 0.5 mark