CSC522: Automated Learning and Data Analysis
Homework 4
Rajshree Jain (rjain27), Srujana Rachakonda (srachak)

Virtual Meeting Schedules:

Date/Meeting Time	**Attendee 1**	**Attendee 2**
4/14/2020 (5:00pm-6:00pm)	Rjain27 (attended)	Srachak (attended)
4/19/2020 (5:00pm-6:00pm)	Rjain27 (attended)	Srachak (attended)

Q1 SVM Theory [Yang Shi]

(a)(i)

Class 1 instances:

X1	**X2**	**Y**
6	4	+1
5	6	+1

Class 2 Instances:

X1	**X2**	**Y**
7	8	-1
8	9	-1

According to the graph we can tell that the (5,6) and (7,8) are the support vectors.

Therefore the margin lines through these points will be as follows:

Margin line 1: +1(5w1 + 6w2 + w0) - 1 = 0 (1)
Margin line 2: -1(7w1+ 8w2 + w0) - 1 = 0 i.e 7w1+ 8w2 + w0 + 1 = 0 (2)

Inorder to get a third system of equations lets require that the slope of the hyperplane be perpendicular to the line joining our support vectors.

Slope of our line joining the support vectors = (7-5)/(8-6) = 2/2 = 1
Therefore, the slope of the line perpendicular to this line would be = -1/1 = -1

In 2D SVMs the slope of the hyperplane is -w1/w2, equating these two we get the following equation.

-w1 = -w2 i.e w1 = w2 (3)

Solving the above system of equations 1, 2 and 3 we get the following values for w1, w2 and w0

$w1 = -\frac{1}{2} \quad w2 = -\frac{1}{2} \quad w0 = 13/2$

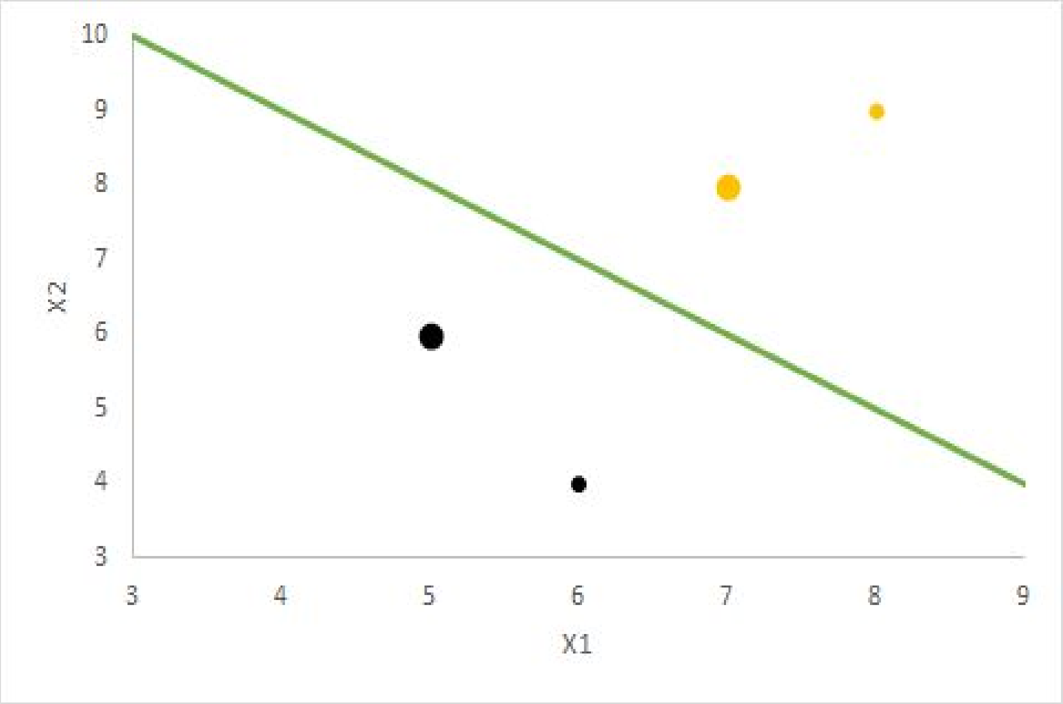
We know that the equation of the hyperplane is given as follows:

$(w^T x + w0) = 0$

Substituting the values of weights derived from the above equations to get the equation of hyperplane as follows:

$-\frac{1}{2} X1 - \frac{1}{2} X2 + 13/2 = 0$

(ii)



The green line is the decision boundary.
The support vectors are the black and yellow points which are bigger in size.

Support vectors: (5,6) and (7,8)

(b)

(i) We cannot make a hard margin svm, since the data is not linearly separable in 1D. As the figure shows, the two class instances are mixed and the class2 instances lie in between class1 instances. Therefore, we cannot make a hard margin svm. Since, hard margin svm cannot handle error and is very sensitive to outliers.

(ii) If we trained a soft margin SVM on this, we would go for C -> 0 since, C -> inf effectively means that it is a hard margin SVM and we do not want that since this is very sensitive to outliers.

(iii) $K(xi, xj) = phi(Xi). phi(X2) = (1+2xixj)^2$

On expanding this equation we get the following:

$K(Xi, Xj) = 1 + 4X_i^2.X_j^2 + 4X_i.X_j$

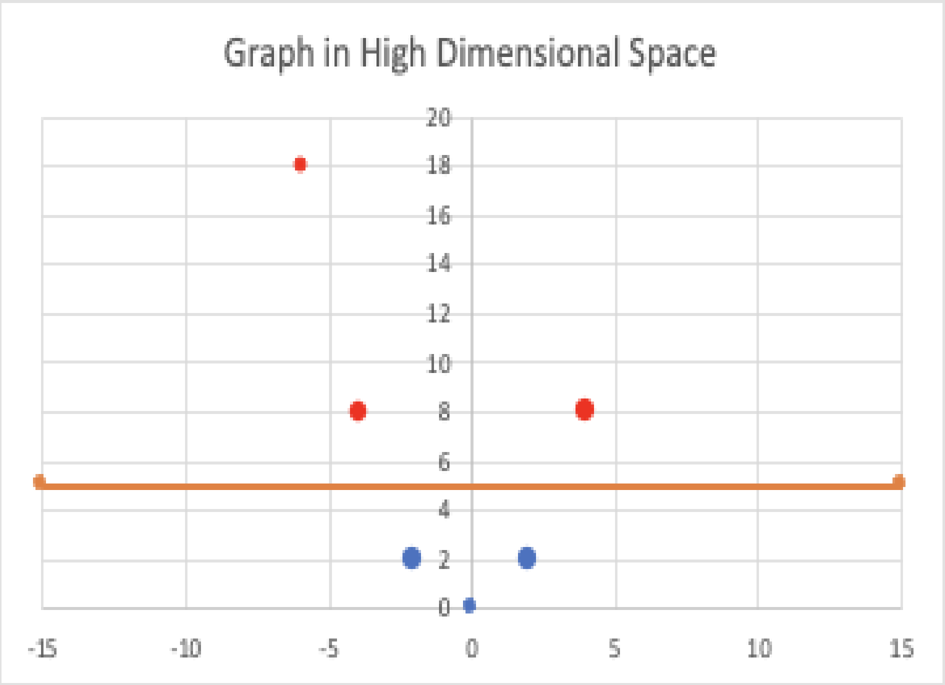
Representing this as a dot product would look as follows:
$[2X_i^2\ 2X_i\ 1\ ]^T[2X_j^2\ 2X_j\ 1]^T = phi(X_i).phi(X_j)$

$phi(X) = 2X^2 + 2X + 1$
As defined by the size of the above matrix, our transformed data has 3 dimensions.

(iv) Using the above phi(X) for all the provided data points. We end up with the following coordinates:

X	X value	Y	Point	2D points
X1	-3	-1	(1,-6,18)	(-6,18)
X2	-2	-1	(1,-4,8)	(-4,8)
X3	-1	1	(1,-2,2)	(-2,2)
X4	0	1	(1,0,0)	(0,0)
X5	1	1	(1,2,2)	(2,2)
X6	2	-1	(1,4,8)	(4,8)

Even Though the data is in 3D space, we see that all the coordinates have the same value of 1 for the first coordinate.This does not give us any information, therefore can be removed so that we can represent the data in 2D and modified as shown in the last column.



(v) Yes it is possible to linearly separate the data since the line y=5 clearly separates the data points into two different classes. The decision boundary is depicted by the orange line in the above graph. All the support vectors are enlarged points in the graph. We have four support vectors.

Orange data points → Class -1
Blue data points → Class 1

(vi) We have the Lagrange multipliers for X2, X3, X5 and X6 points, which are the support vectors. We know that Lagrange multipliers for instances which are not the support vectors have values of zero. Therefore, alpha1 and alpha4 = 0

(vii) Equation to find the class of new variable Z is as follows:

f(z) = sign(sum(alpha$_i$ * y$_i$ * (phi(X$_i$) * phi(Z)) + w0)
We know that the kernel function K(X$_i$, Z) = phi(X$_i$) * phi(Z)

Substituting this in the above equation we get,

f(z) = sign(sum(alpha$_i$ * y$_i$ * K(X$_i$, Z)) + w0)

Since, alpha = 0 for data points which are not support vectors. Summation includes only X2, X3, X5 and X6.

W0 = -⅔
K(X2, Z) = 25
K(X3, Z) = 4
K(X5, Z) = 16
K(X6, Z) = 49

Using the above values we find the value of f(z) as follows:

f(z) = -6.0667

Here sign = -1

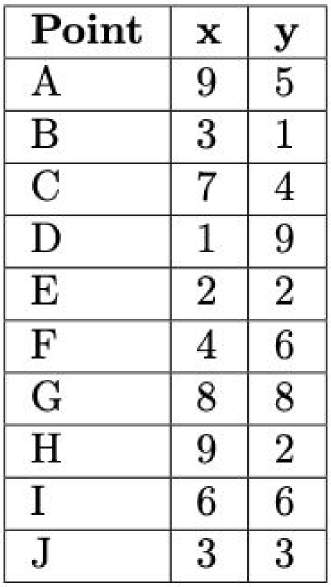
Therefore the data point Z is classified as class -1. I.e Y$_z$ = -1

Q2 K-Means Clustering (14 points) [Ge Gao]
Use the K-means clustering algorithm with Euclidean Dis- tance to cluster the 10 data points in Figure 3 into 3 clusters. Suppose that the initial seed centroids are at points: C, I and H. The data are also given in tabular format in Table 2.

(a) After each iteration of k-means, report the coordinates of the new centroids and which cluster each data point belongs to. Stop when the algorithm converges and clearly label on the graph where the algorithm converges. To report your work, either:

i. Draw the result clusters and the new centroid at the end of each round (including the first round). You can use the image hw4q2 start.jpg, included with your homework, to mark clusters and centroids. Additionally, indicate the coordinates (x,y) alongside corresponding centroids.

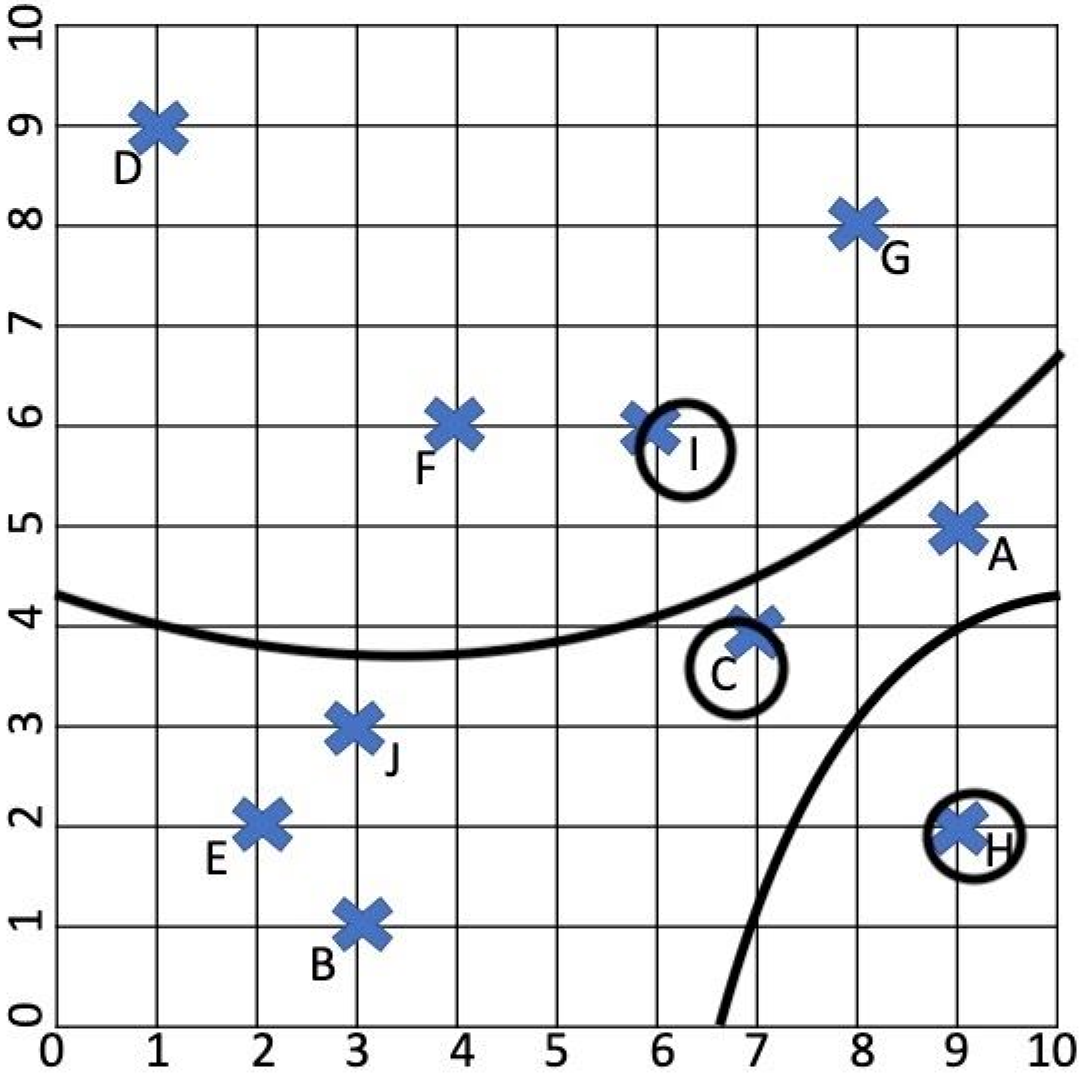
ii. Give your answer in tabular format with the following attributes: Round (e.g. Round 1, 2, etc), Points (e.g. {A, B, C}), and Cluster ID. Also report the centroids for each cluster after each round.



Point	x	y
A	9	5
B	3	1
C	7	4
D	1	9
E	2	2
F	4	6
G	8	8
H	9	2
I	6	6
J	3	3

Euclidean Distance Table:

	A	B	C	D	E	F	G	H	I	J
A	0	sqrt(52)	sqrt(5)	sqrt(80)	sqrt(58)	sqrt(26)	sqrt(10)	sqrt(9)	sqrt(10)	sqrt(40)
B	sqrt(52)	0	sqrt(25)	sqrt(68)	sqrt(2)	sqrt(26)	sqrt(74)	sqrt(37)	sqrt(34)	sqrt(4)
C	sqrt(5)	sqrt(25)	0	sqrt(61)	sqrt(29)	sqrt(13)	sqrt(17)	sqrt(8)	sqrt(5)	sqrt(17)
D	sqrt(80)	sqrt(68)	sqrt(61)	0	sqrt(50)	sqrt(18)	sqrt(50)	sqrt(113)	sqrt(34)	sqrt(40)
E	sqrt(58)	sqrt(2)	sqrt(29)	sqrt(50)	0	sqrt(20)	sqrt(72)	sqrt(49)	sqrt(32)	sqrt(2)
F	sqrt(26)	sqrt(26)	sqrt(13)	sqrt(18)	sqrt(20)	0	sqrt(20)	sqrt(41)	sqrt(4)	sqrt(10)
G	sqrt(10)	sqrt(74)	sqrt(17)	sqrt(50)	sqrt(72)	sqrt(20)	0	sqrt(37)	sqrt(8)	sqrt(50)
H	sqrt(9)	sqrt(37)	sqrt(8)	sqrt(113)	sqrt(49)	sqrt(41)	sqrt(37)	0	sqrt(25)	sqrt(37)
I	sqrt(10)	sqrt(34)	sqrt(5)	sqrt(34)	sqrt(32)	sqrt(4)	sqrt(8)	sqrt(25)	0	sqrt(18)
J	sqrt(40)	sqrt(4)	sqrt(17)	sqrt(40)	sqrt(2)	sqrt(10)	sqrt(50	sqrt(37)	sqrt(18)	0



The clusters after the first round are {D,F,I,G}
{A,C,E,J,B}
{H}

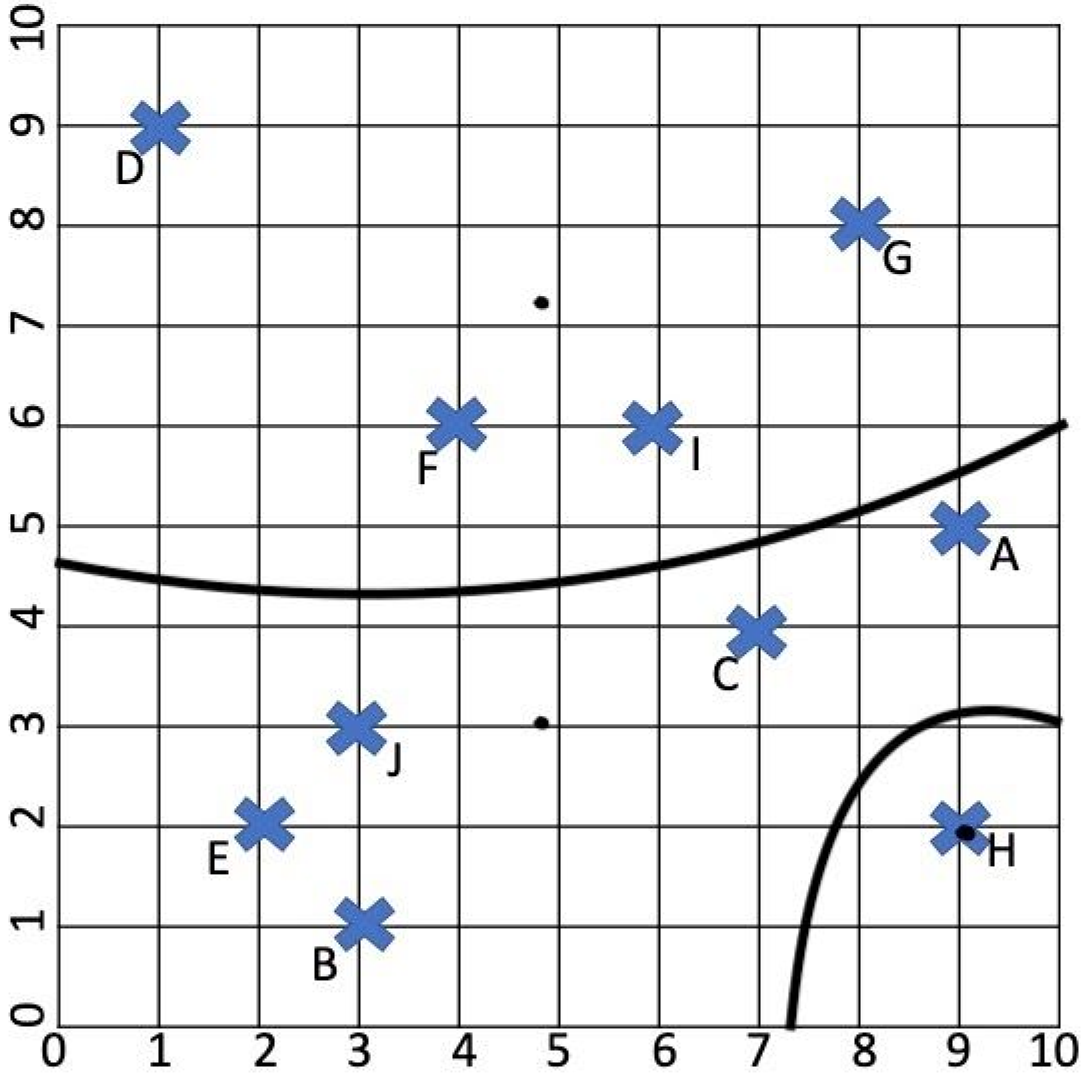
Finding the new centroids in the above clusters

(D+F+I+G)/4
(1+4+6+8)/4 , (9+6+6+8)/4
C1 - (4.75,7.25)

(A+C+E+J+B)/5
C2 - (4.8, 3)

H
C3 - (9,2)

We can see the new centroids and the new clusters



Now since we have found the new centroids we will try to cluster again:

C1
D, F, I G

Distance of A with C1 and C2 and C3
sqrt(23.125)
sqrt(21.64)
sqrt(9)

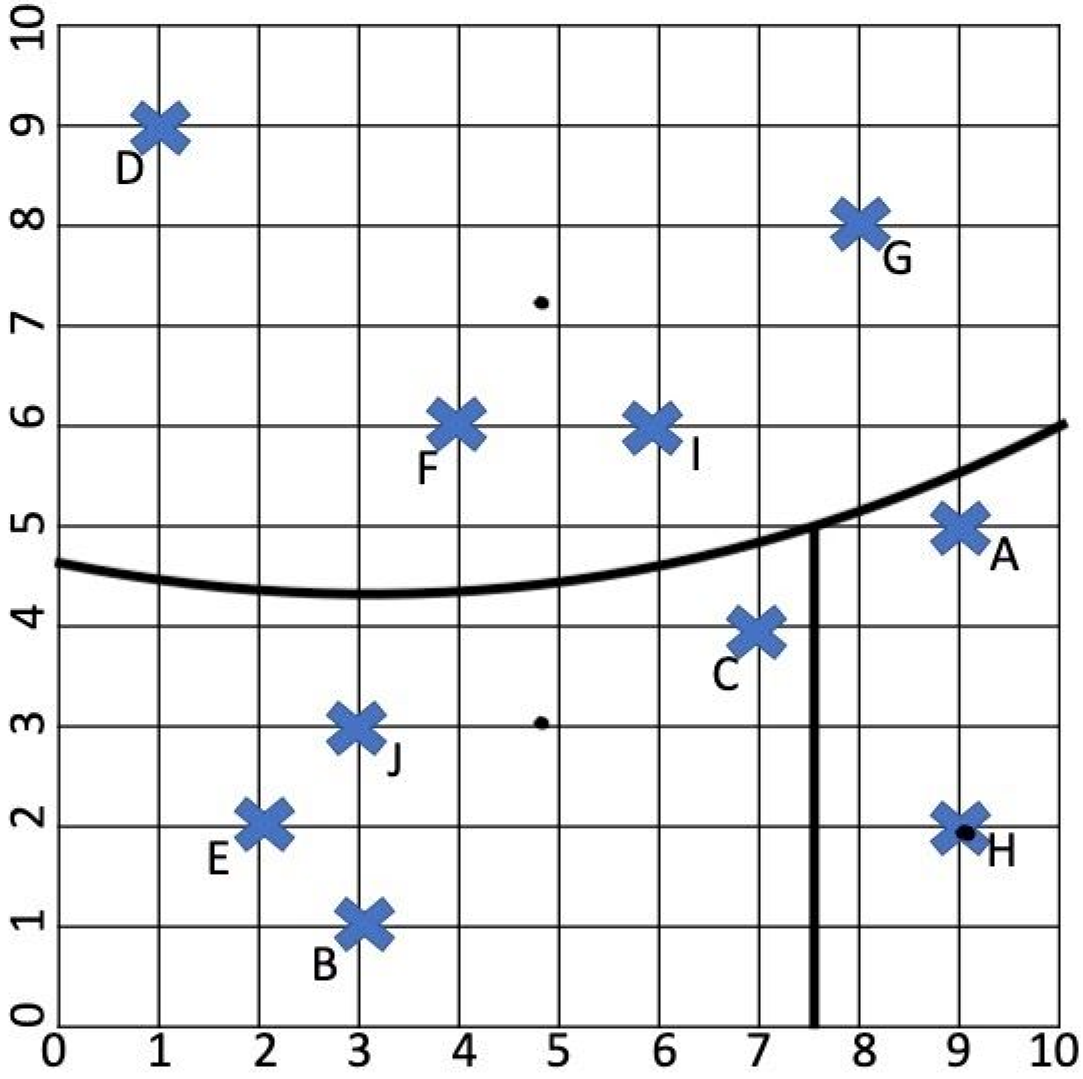
A will be with C3

Distance of C will be with C1, C2, C3
sqrt(15.625)
sqrt(5.84)
sqrt(8)

So C will be with C2

Further, we can clearly see that J,E,B will be with C2

So, the new Clusters will be:



Clusters after second round:
C1-D,F,I,G
C2-A,H
C3-C,J,E,B

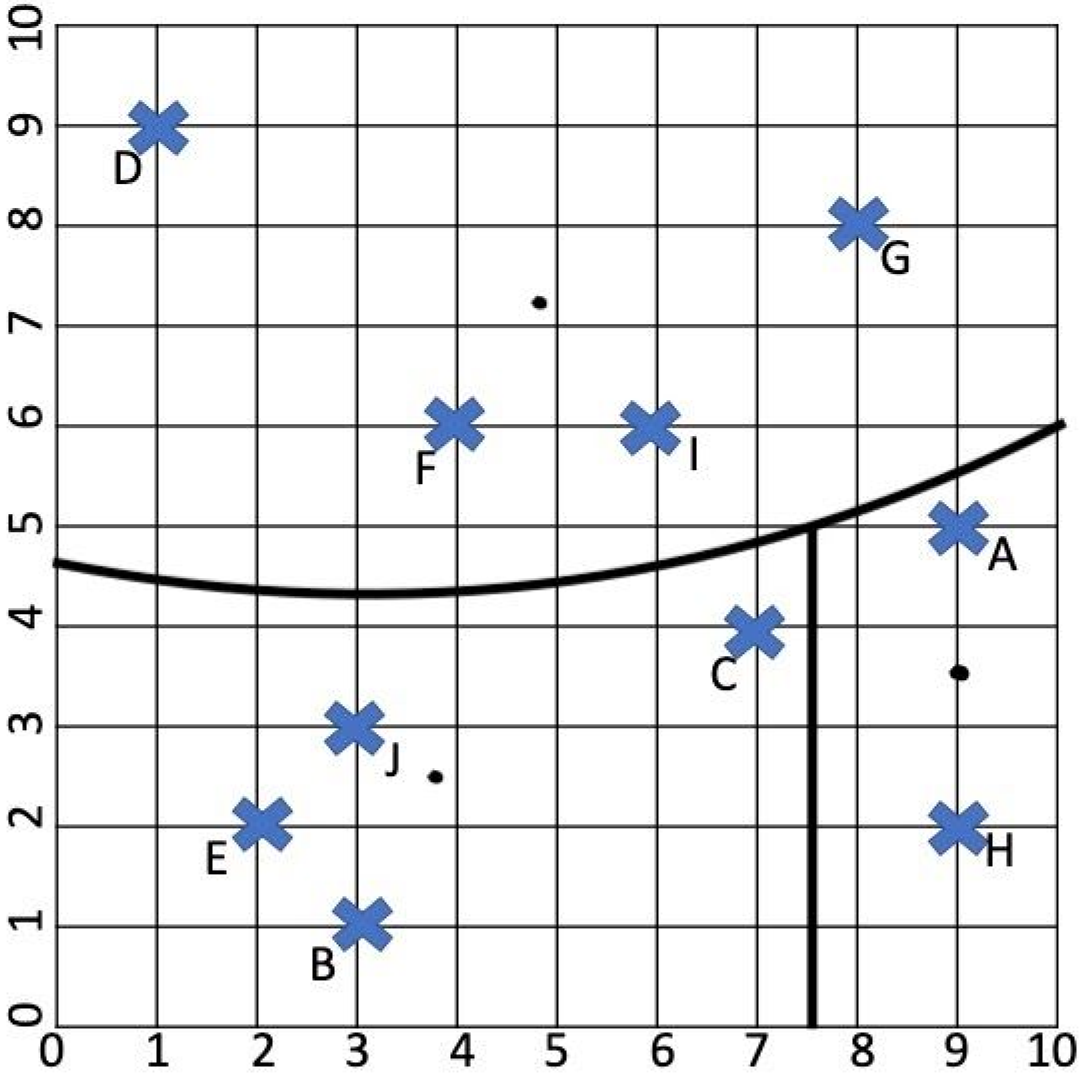
Now finding new centroids based on the changes in clusters:
(D+F+I+G)/4

C1 - (4.75,7.25)

C2-(C+E+J+B)/4
3.75, 2.5

C3-(A+H)/2
(9,3.5)

These centroids can be seen in the below diagram:



Now we again try to determine which datapoint would be in which cluster from the above diagram:

We can clearly see that B,E,J will be near C2

D and F are definitely near C1

Now we start to check for the point C

Is C near C1, C2, C3??
Distance with C1-sqrt(15.625)
Distance with C2-sqrt(12.81)
Distance with C3-sqrt(4.25)

So we see that now C will belong to the Cluster C3

Now we will check that G will be be in C1 or C3 as it is definitely far from C2
Distance with C1 - sqrt(11.125)
Distance with C2 - sqrt(48.3)
Distance with C3 - sqrt (21.25)

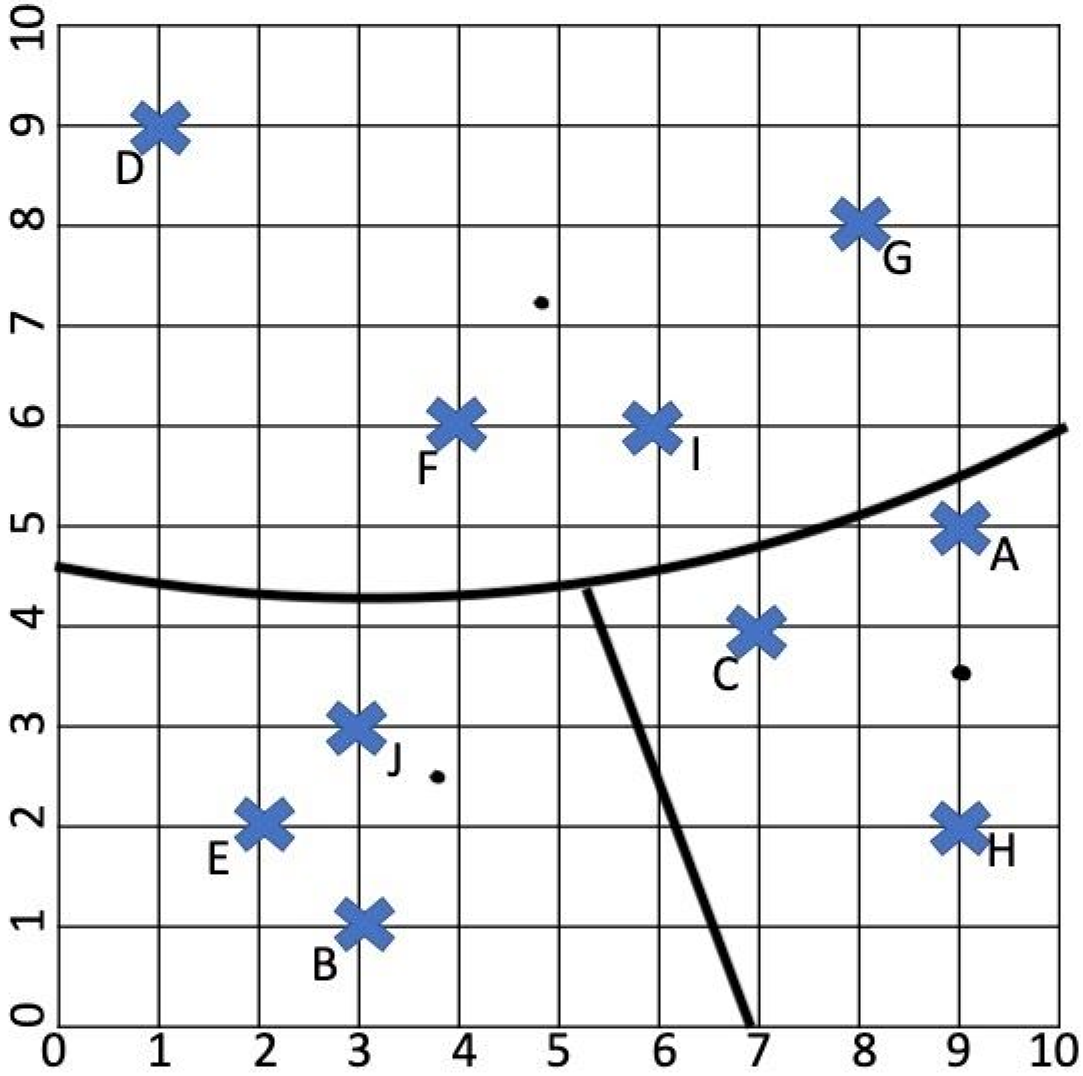
Hence G will be in C1

We will check if I is in C1, C2, C3
Distance with C1 - sqrt(3.125)
Distance with C2 - sqrt(17.3)
Distance with C3 - sqrt (15.25)

So I will be in C1

Now the Clusters after round 3 will look like:
C1-{F,I,D,G}
C2-{B,E,J}
C3-{A,H,C}

The clusters would look like something below:



Now we will calculate the new centroids on the above clusters

C1

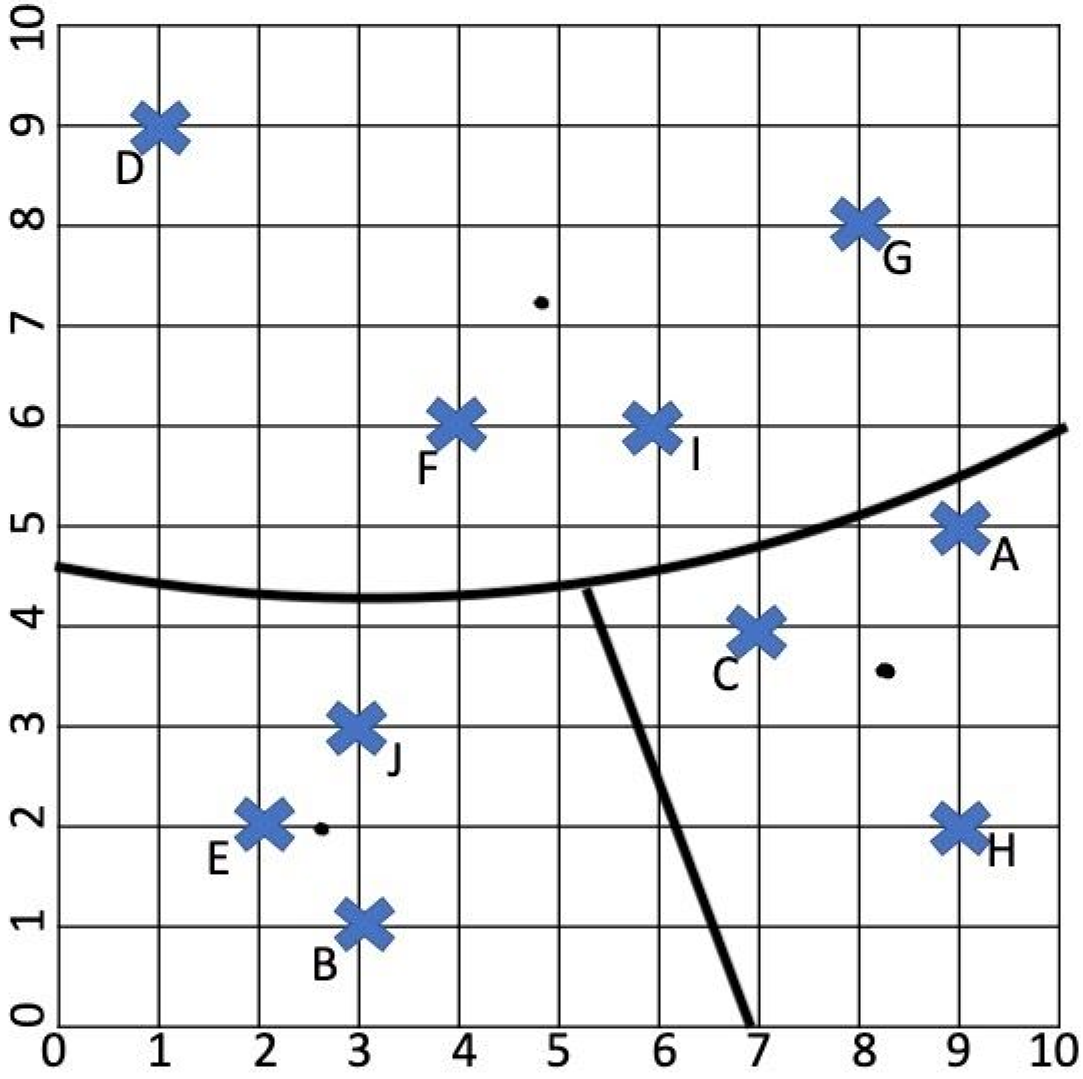
(D+F+I+G)/4=4.75,7.25

C2

(B+E+J)/3=2.66,2

C3

(A+C+H)/3=8.33,3.66



Now let us check if the Clusters would further change:
We can see that none of the clusters would change further.

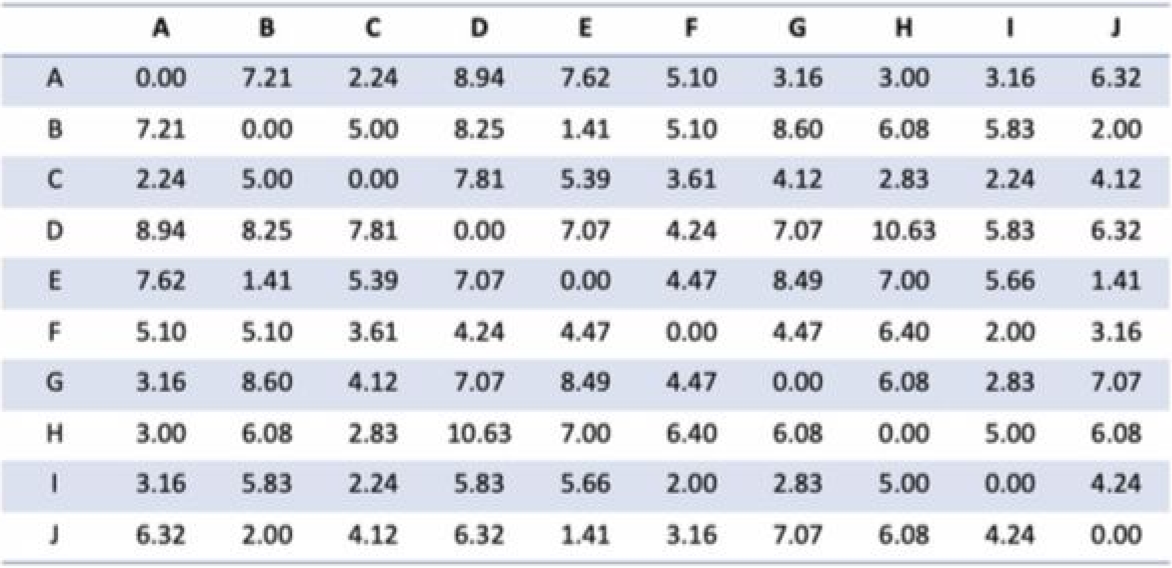
Hence, we can see that after this round the Clusters did not change at all so the final Clusters are:
C1-{F,I,D,G} Centroid-4.75,7.25

C2-{B,E,J} Centroid-2.66,2

C3-{A,H,C} Centroid-8.33,3.66

(b) How many rounds are needed for the K-means clustering algorithm to converge?
3 Rounds. Because at round 4 same clusters were formed as at round 3.

Q3) Hierarchical Clustering (18 points) [Ge Gao] We will use the same dataset as in Question 2 (shown in Figure 3) for Hierarchical Clustering. The Euclidean Distance matrix between each pair of the data points is given in Figure 4 below:



	A	B	C	D	E	F	G	H	I	J
A	0.00	7.21	2.24	8.94	7.62	5.10	3.16	3.00	3.16	6.32
B	7.21	0.00	5.00	8.25	1.41	5.10	8.60	6.08	5.83	2.00
C	2.24	5.00	0.00	7.81	5.39	3.61	4.12	2.83	2.24	4.12
D	8.94	8.25	7.81	0.00	7.07	4.24	7.07	10.63	5.83	6.32
E	7.62	1.41	5.39	7.07	0.00	4.47	8.49	7.00	5.66	1.41
F	5.10	5.10	3.61	4.24	4.47	0.00	4.47	6.40	2.00	3.16
G	3.16	8.60	4.12	7.07	8.49	4.47	0.00	6.08	2.83	7.07
H	3.00	6.08	2.83	10.63	7.00	6.40	6.08	0.00	5.00	6.08
I	3.16	5.83	2.24	5.83	5.66	2.00	2.83	5.00	0.00	4.24
J	6.32	2.00	4.12	6.32	1.41	3.16	7.07	6.08	4.24	0.00

a) Perform single link hierarchical clustering. Show your work at each iteration by giving the inter- cluster distances. Report your results by drawing a corresponding dendrogram. The dendrogram should clearly show the order and the height in which the clusters are merged. If possible, use a program to construct your dendrogram (e.g. PowerPoint, LucidChart2, or VisualParadigm3). Scanned hand drawings will also be accepted if they are very clear.

The distance is minimum between B, E that us 1.41 so they will form a cluster:

	A	BUE	C	D	F	G	H	I	J
A	0	7.21	2.24	8.94	5.10	3.16	3	3.16	6.32
BUE	7.21	0	5	7.07	4.47	8.49	6.08	5.66	1.41
C	2.24	5	0	7.81	3.61	4.12	2.83	2.24	4.12
D	8.94	7.07	7.81	0	4.24	7.07	10.63	5.83	6.32
F	5.10	4.47	3.61	4.24	0	4.47	6.40	2.00	3.16
G	3.16	8.49	4.12	7.07	4.47	0	6.08	2.83	7.07
H	3.00	6.08	2.83	10.63	6.40	6.08	0	5	6.08
I	3.16	5.66	2.24	5.83	2.00	2.83	5	0	4.24
J	6.32	1.41	4.12	6.32	3.16	7.07	6.08	4.24	0

Now the distance matrix has a minimum of 1.41 between J and B,E so we will make it a cluster:

	A	BUEUJ	C	D	F	G	H	I
A	0	6.32	2.24	8.94	5.10	3.16	3	3.16
BUEUJ	6.32	0	4.12	6.32	3.16	7.07	6.08	4.24
C	2.24	4.12	0	7.81	3.61	4.12	2.83	2.24
D	8.94	6.32	7.81	0	4.24	7.07	10.63	5.83
F	5.10	3.16	3.61	4.24	0	4.47	6.40	2.00
G	3.16	7.07	4.12	7.07	4.47	0	6.08	2.83
H	3.00	6.08	2.83	10.63	6.40	6.08	0	5
I	3.16	4.24	2.24	5.83	2.00	2.83	5	0

Now the least, distance in the above matrix is 2.00 between F,I

	A	BUEUJ	C	D	FUI	G	H
A	0	6.32	2.24	8.94	3.16	3.16	3.00
BUEUJ	6.32	0	4.12	6.32	3.16	7.07	6.08
C	2.24	4.12	0	7.81	2.24	4.12	2.83
D	8.94	6.32	7.81	0	4.24	7.07	10.63
FUI	3.16	3.16	2.24	4.24	0	2.83	5
G	3.16	7.07	4.12	7.07	2.83	0	6.08
H	3.00	6.08	2.83	10.63	5	6.08	0

So we will make a cluster of C,A as the distance between them is 2.24

	AUC	BUEUJ	D	FUI	G	H
AUC	0	4.12	7.81	2.24	3.16	2.83
BUEUJ	4.12	0	6.32	3.16	7.07	6.08
D	7.81	6.32	0	4.24	7.07	10.63
FUI	2.24	3.16	4.24	0	2.83	5

G	3.16	7.07	7.07	2.83	0	6.08
H	2.83	6.08	10.63	5	6.08	0

Now the smallest number in the above matrix is 2.24 that is between A,C and F,I

	AUCUFUI	BUEUJ	D	G	H
AUCUFUI	0	3.16	4.24	2.83	2.83
BUEUJ	3.16	0	6.32	7.07	6.08
D	4.24	6.32	0	7.07	10.63
G	2.83	7.07	7.07	0	6.08
H	2.83	6.08	10.63	6.08	0

Now the Lowest number in the above matrix is 2.83 so we will merge A,C,F,I and G

	AUCUF,UIUG	BUEUJ	D	H
AUCUFUIUG	0	3.16	4.24	2.83
BUEUJ	3.16	0	6.32	6.08
D	4.24	6.32	0	10.63
H	2.83	6.08	10.63	0

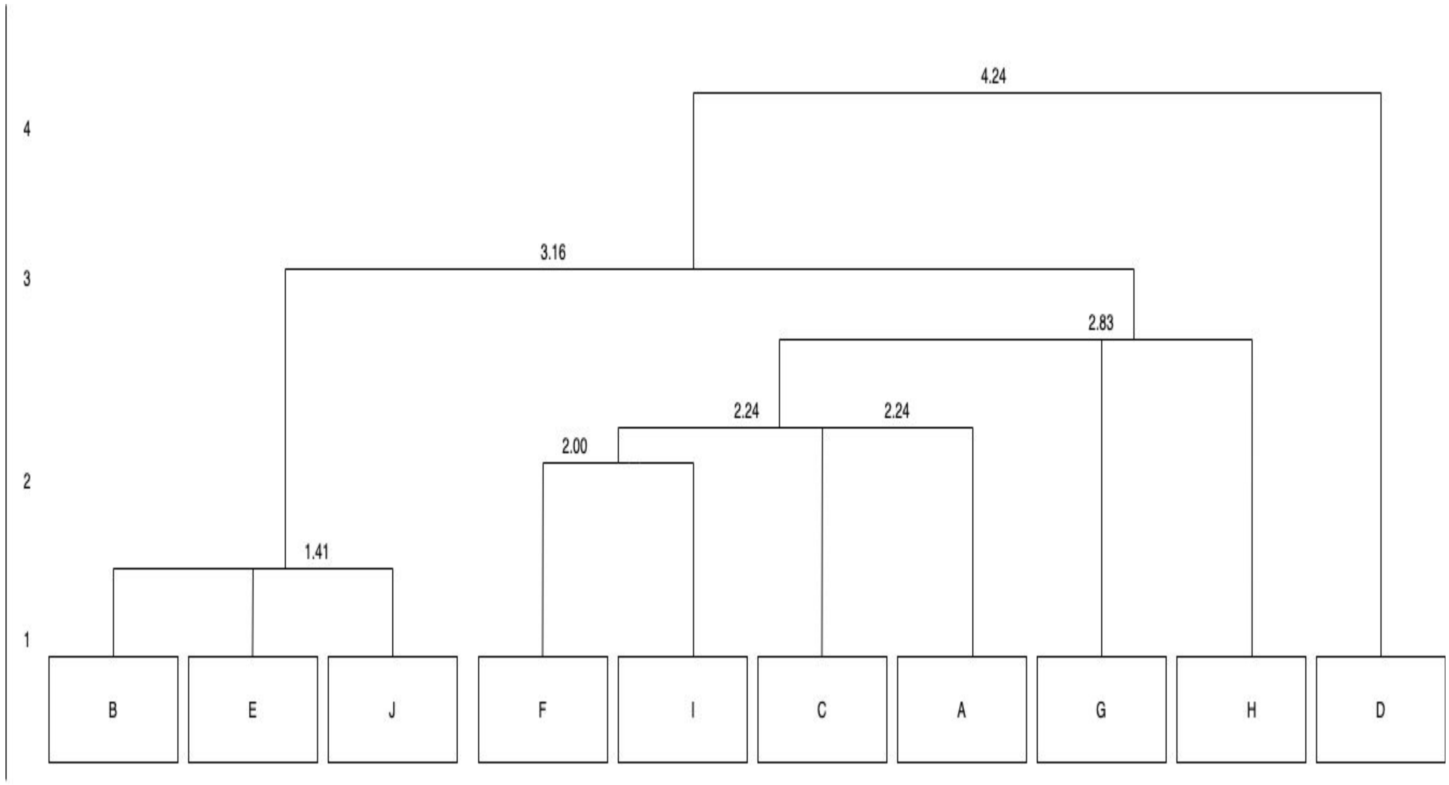
Now the lowest value in the 2.83 between A,C,F,I,G and H

	AUCUFUIUGU H	BUEUJ	D
AUCUFUIUGU H	0	3.16	4.24
BUEUJ	3.16	0	6.32
D	4.24	6.32	0

Now merge A,C,F,I,G, H and B,E,J as value is 3.16

	A,C,F,I,G, H, B, E, J	D
A,C,F,I,G, H, B, E, J	0	4.24
D	4.24	0

The dendrogram will look like:



(iii)
Perform complete link hierarchical clustering on the dataset. As above, show your calculations and report the corresponding dendrogram.

The distance is minimum between B, E that us 1.41 so they will form a cluster:

	A	BUE	C	D	F	G	H	I	J
A	0	7.62	2.24	8.94	5.10	3.16	3	3.16	6.32
BUE	7.62	0	5.39	8.25	5.10	8.60	7.00	5.83	2.00
C	2.24	5.39	0	7.81	3.61	4.12	2.83	2.24	4.12
D	8.94	8.25	7.81	0	4.24	7.07	10.63	5.83	6.32
F	5.10	5.10	3.61	4.24	0	4.47	6.40	2.00	3.16
G	3.16	8.60	4.12	7.07	4.47	0	6.08	2.83	7.07

H	3.00	7.00	2.83	10.63	6.40	6.08	0	5	6.08
I	3.16	5.83	2.24	5.83	2.00	2.83	5	0	4.24
J	6.32	2.00	4.12	6.32	3.16	7.07	6.08	4.24	0

Now since the minimum in the Cluster is 2 between B,E and J we will cluster them:

	A	BUEUJ	C	D	F	G	H	I
A	0	7.62	2.24	8.94	5.10	3.16	3	3.16
BUEUJ	7.62	0	5.39	8.25	5.10	8.60	7.00	5.83
C	2.24	5.39	0	7.81	3.61	4.12	2.83	2.24
D	8.94	8.25	7.81	0	4.24	7.07	10.63	5.83
F	5.10	5.10	3.61	4.24	0	4.47	6.40	2.00
G	3.16	8.60	4.12	7.07	4.47	0	6.08	2.83
H	3.00	7.00	2.83	10.63	6.40	6.08	0	5
I	3.16	5.83	2.24	5.83	2.00	2.83	5	0

Now the minimum distance in the above matrix is 2 between F and I so we will cluster:

	A	BUEUJ	C	D	FUI	G	H
A	0	7.62	2.24	8.94	5.10	3.16	3
BUEUJ	7.62	0	5.39	8.25	5.83	8.60	7.00
C	2.24	5.39	0	7.81	3.61	4.12	2.83
D	8.94	8.25	7.81	0	5.83	7.07	10.63
FUI	5.10	5.83	3.61	5.83	0	4.47	6.40
G	3.16	8.60	4.12	7.07	4.47	0	6.08
H	3.00	7.00	2.83	10.63	6.40	6.08	0

Now the minimum value in the above matrix is 2.24 between A,C so we will cluster them:

	AUC	BUEUJ	D	FUI	G	H

AUC	0	7.62	8.94	5.10	4.12	3
BUEUJ	7.62	0	8.25	5.83	8.60	7.00
D	8.94	8.25	0	5.83	7.07	10.63
FUI	5.10	5.83	5.83	0	4.47	6.40
G	4.12	8.60	7.07	4.47	0	6.08
H	3	7.00	10.63	6.40	6.08	0

Now the next smallest element in the above matrix is 3 that is between AC and H so merge

	AUCUH	BUEUJ	D	FUI	G
AUCUH	0	7.62	10.63	6.40	6.08
BUEUJ	7.62	0	8.25	5.83	8.60
D	10.63	8.25	0	5.83	7.07
FUI	6.40	5.83	5.83	0	4.47
G	6.08	8.60	7.07	4.47	0

Next smallest element is 4.47 between F,I and G so we will cluster them:

	AUCUH	BUEUJ	D	FUIUG
AUCUH	0	7.62	10.63	6.40
BUEUJ	7.62	0	8.25	8.60
D	10.63	8.25	0	7.07
FUIUG	6.40	8.60	7.07	0

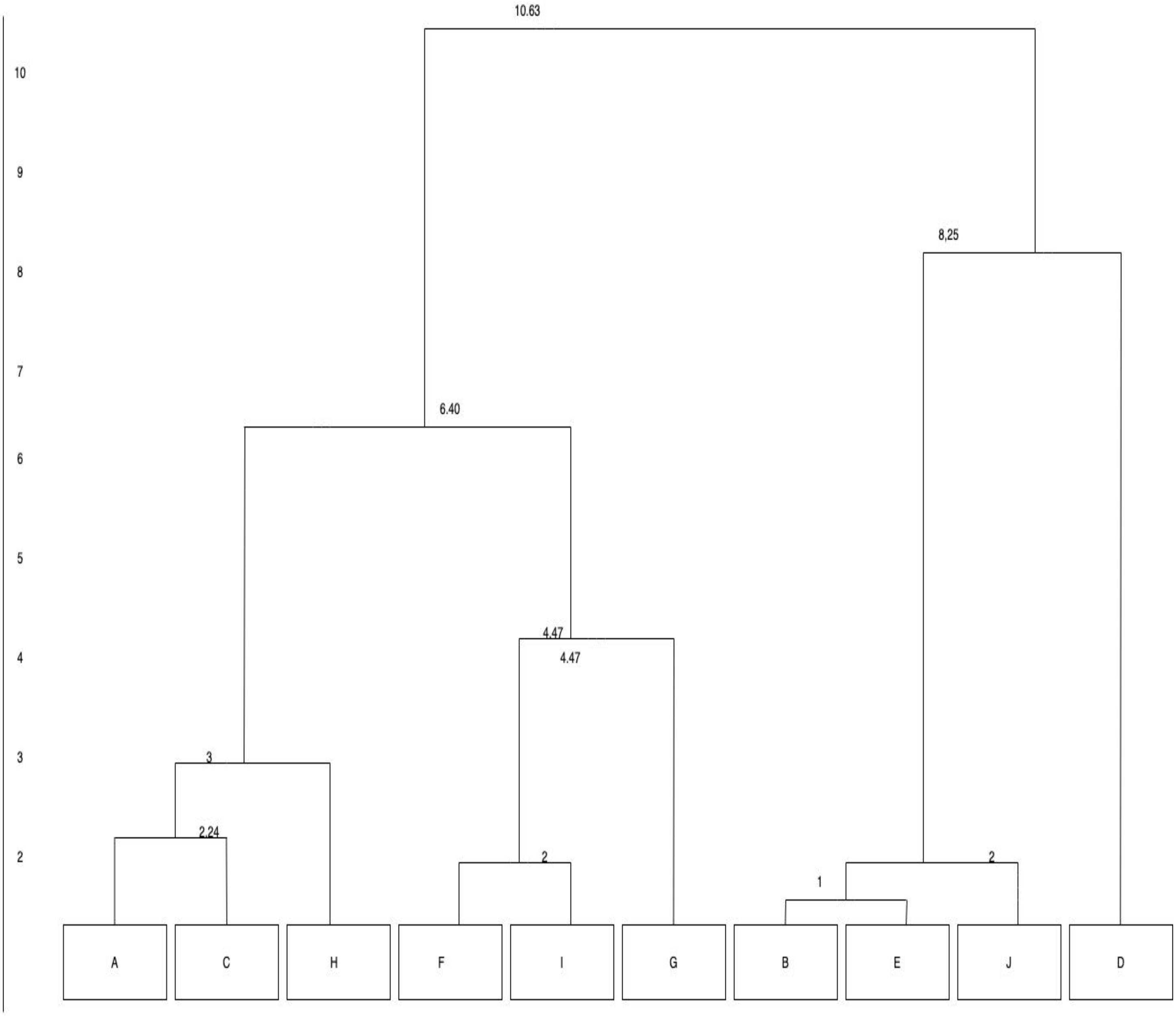
Now the smallest element in the matrix is 6.40 so we merge A,C,H and F,I,G:

	AUCUH U FUIUG	BUEUJ	D
AUCUHUFUIUG	0	8.60	10.63
BUEUJ	8.60	0	8.25
D	10.63	8.25	0

Now we will merge B,E,J and D (8.25)

	AUCUHUFUIUG	BUEUJUD
AUCUHUFUIUG	0	10.63
BUEUJUD	10.63	0

So the dendrogram would look something like below:



(c) If we assume there are two clusters, will the single or complete link approach give a better clustering? Justify your answer.

A single link clustering algorithm is decent at not elliptical shapes. However, single link clustering is a bit sensitive to noise and outliers because as soon as noise comes up the clusters that were very distinct before get closer and become less distinct.

Complete link is less susceptible to outliers. Little bit of noise would not let one cluster to spread over the other because it compares the farthest away points while trying to merge them. However, it does not do very well with datasets of different sizes and densities.

After studying about all the above properties of Single Link and Complete Link Hierarchical clustering, I had the following observations:

When we see both Single Link and Complete Link at K=3 in both the cases there were 3 Clusters:

{D}, {B,E,J} and {A,C,H,F,I,G}

Now we can see how the difference came in the 2 clustering algorithms. When we would see at K=2

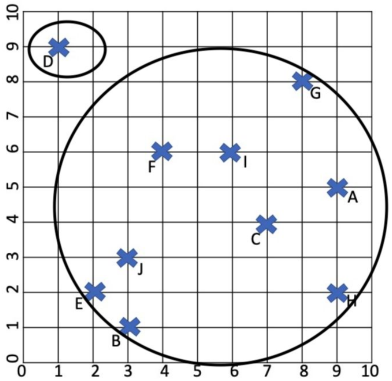
Single Link: {D}, {B,E,J,A,C,H,F,I,G}

Complete Link: {D,B,E,J} and {A,C,H,F,I,G}

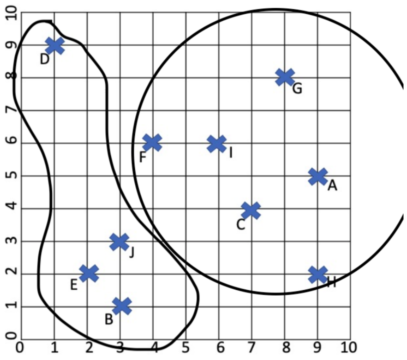
Although we say that Complete Link Clustering is much prone to noise and outliers but here due to the way it works it created a cluster for {D,B,E,J}. Seeing this I feel Single link Clustering is better because it made clusters for the points that were closely placed.

Below in the diagram we can see how the clusters would have looked like at K=2.

For Single Link Clustering:



For Complete Link:



In the above we can see that the Single Link Clustering seemingly performed better. It looks like D should not be clustered with B,E,J which it did in case of Complete Link happened. Also, we see that in case of Single Link Clustering D forms one cluster and B,E,J,A,C,H,F,I,G form one cluster which looks correct because D is far from B,E,J. So, Single Link Cluster performs better.

(d) Consider the single-link hierarchical clustering with 3 clusters. Compare the quality of this single link clustering with your final K-means clustering in Question 1. To evaluate the quality of each clustering, calculate its corresponding Sum of Squared Error (SSE). Based on this measure, which clustering (k-means, single link), is best? Do you agree with this assessment? Explain why in 1-2 sentences. Note: you may want to write some code to help speed up these calculations, which you can include in lieu of showing your work.

Single Link Clustering at K=3

{D} Centroid-1, 9

{B,E,J} Centroid - 2.667, 2

{A,C,H,F,I,G} Centroid - 7.167, 5.167

Final K means clustering Clusters from q2:

{F,I,D,G} Centroid-4.75,7.25

{B,E,J} Centroid-2.66,2

{A,H,C} Centroid-8.33,3.66

The formulae of SSE is :

$$\sum_{i=1}^{k}\sum_{x_j\in S_i}\left\|x_j-u_i\right\|^2$$

The above formulae means the sum of squared distances from cluster points to the centroids of each cluster.

Now we can calculate the SSE for both the Clusters:

Clusters	Point	Squared Errors
{D}	D	0
{B,E,J}	B	1.11
	E	0.44
	J	1.11
{A,C,H,F,I,G}	A	3.41
	C	1.37
	H	10.69
	F	8.77
	I	2.05
	G	13.37

Total SSE for Single Link Hierarchical clustering is:

42.32

Now calculating SSE for K means Clustering:

Clusters	Point	Squared Errors
{F,I,D,G}	F	2.125
	I	3.1250
	D	17.124
	G	11.124

{B,E,J}	B E J	1.11 0.44 1.11
{A,H,C}	A H C	2.24 3.21 1.88

The total SSE is 43.488

We can see that the SSE is more for K means clustering and Single Link Clustering performed better.

But when we look at the clusters I feel that K means clustering was better as the clusters that were formed were more defined and in better patterns. Further, I also feel that using SSE to compare a Hierarchical and K means clustering is not the best way to compare the 2 clustering algorithms. K means Clustering is designed to work in a way that the data points would try to go near the centroids, hence reducing the SSE. Also Single Link algorithm is better on non globular shapes. But here the result came otherwise and the SSE for K means was more. Hence, we cannot say that SSE was the right measure to determine which algorithm was better. Apart from that the data points are so less in number that proper clusters could not be formed.

Q4) Association Rule Mining (6 points) [Ge Gao]

(a) What is the maximum number of unique itemsets that can be extracted from this data set (including itemsets that have 0 support)? Briefly explain your answer in 1-2 sentences.

Maximum number of unique itemsets is given by 2^d where d is the number of items. Therefore, in this case we have d = 6. The maximum number of unique itemsets are $2^6 = 64$ itemsets.

(b) What is the maximum number of association rules that can be extracted from this data set (including rules that have zero support)? Briefly explain your answer in 2-3 sentences.

Maximum number of association rules is given by $R = \text{sum}_{(k=1 \text{ to } d -1)}(C(d, k) * (\text{sum}_{(j = 1 \text{ to } d-k)}(C(d-k, j))))$
Which can be represented as $= 3^d - 2^{d+1} + 1$

In this case this would be $= 3^6 - 2^7 + 1 = 602$

(c) Compute the support of the itemset: {Eggs , Cola} ?

Support = sigma(X U Y) / | T |

Support(eggs, cola) = 2/10 = ⅕ = 0.2

(d) Compute the support and confidence of association rule: {Bread} -> {Butter} ?

Using the above formula we get support(bread, butter) = 3/10 = 0.3

Confidence(X = bread, Y = butter) = sigma(X U Y) / sigma (X) = 3/6 = 0.5

(e) Given min support = 0.3 and min confidence = 0.6, identify all valid association rules of the form {A,B} -> {C} .

In order to have minimum support of 0.3, we need at least 3 items to be 1 in a transaction and at least 3 transactions as such since our total number of transactions are 10.

The following transactions have 3 1s → 2, 3, 5, 6, 7, 8, 9, 10

Transaction Id	Item1	item 2	Item 3	Item 4
2	Bread	Milk	Cola	
3	Bread	Butter	Beer	
4	Butter	Beer	Cola	
6	Bread	Milk	Butter	Cola
7	Bread	Milk	Eggs	
8	Milk	Butter	Eggs	Cola
9	Bread	Butter	Eggs	Beer
10	Bread	Milk	Cola	

Of these transactions, in order to have minimum support 0.3, 3 items should co-occur 3 times.

There is only one such set of items which satisfy this constraint and that is {Bread, Milk, Cola}

Note: Support for a set of (X U Y) remains the same no matter which items occur on the left hand and right hand side of the association rule.

Now, we try all possible combinations of these 3 items and find the confidence and it is tabulated as follows:

Association Rule	Confidence

{Bread, Milk} → {Cola}	3/4 = 0.75
{Bread, Cola} → {Milk}	3/3 = 1
{Milk, Cola} → {Bread}	3/3 = 1

Since all the above combinations have confidence > 0.6, they are all valid association rules.

(f) In a different dataset, the support of the rule {a} → {b} is 0.46, and the support of the rule {a,c} → {b,d} is 0.23. What can we say for sure about the support of the rule {a} → {b,d}. Explain in 1-2 sentences.

According to the anti-monotone property of support, we know that the support of an itemset never exceeds the support of its subsets. Using this we can say that support of {a} → {b,d} , here X U Y will be {a,b,d} will always be less than that of it's subsets which is {a,b} for the association rule {a} → {b}. Therefore,

we can say that support of {a} → {b,d} < 0.46

Again, using the anti-monotone property, we can say that support of {a,c,b,d} will never exceed support of {a,c,d},

We can say that support of {a} → {b,d} > 0.23

0.23 < support of {a} → {b,d} < 0.46

Q5) Apriori algorithm (12 points) [Yang Shi]

(a)

Transaction ID	Items
t1	A,B,C,D
t2	A,B,C,E
t3	A,B
t4	A,C,E
t5	A,C,D,E
t6	B,C,D
t7	B,C,E
t8	C,D,E

C1:

Itemset	Support Count
A	5
B	5
C	7
D	4
E	5

Since support of all the itemsets are > minimum support count i.e 3 we do not prune and we join as follows:

C2:

Itemset	Support Count
A,B	3
A,C	4
A,D	2
A,E	3
B,C	4
B,D	2
B,E	2
C,D	4
C,E	5
D,E	2

All the highlighted itemsets do not reach the minimum support count and hence get pruned.

We end up with the following:

L2:

Itemset	Support Count
A,B	3
A,C	4
A,E	3

B,C	4
C,D	4
C,E	5

Now, we join to get the following:

C3:

Itemset	**Support Count**
A,B,C	2
A,C,E	3

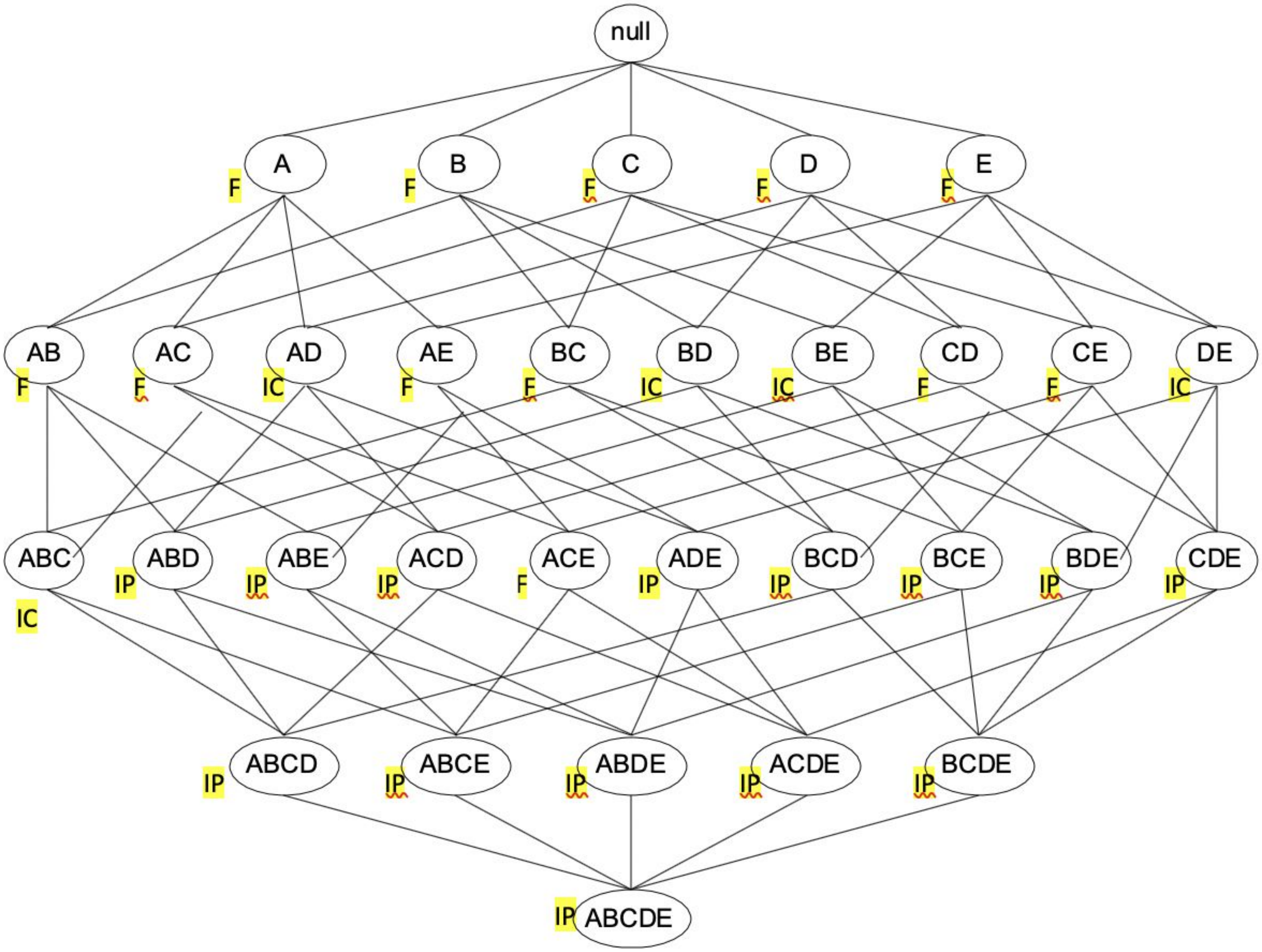
Highlighted itemset does not meet the minimum support count. Therefore gets pruned.

L3:

Itemset	**Support Count**
A,C,E	3

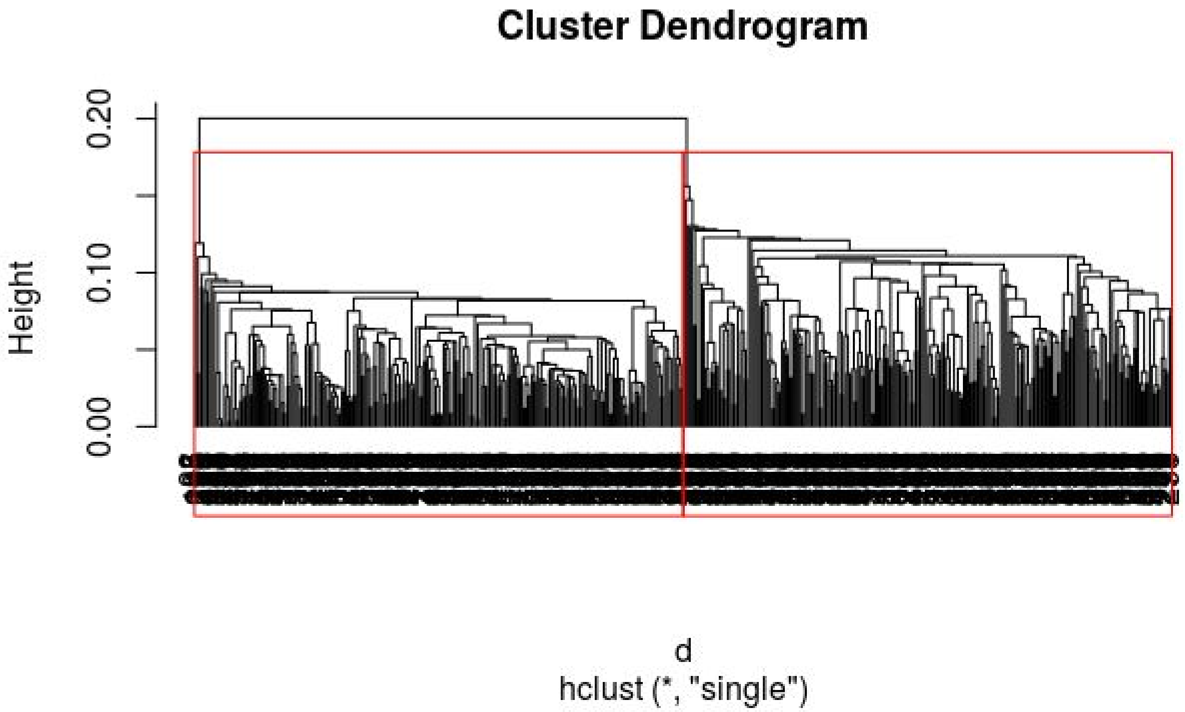
Outcome: A,C,E

(b)

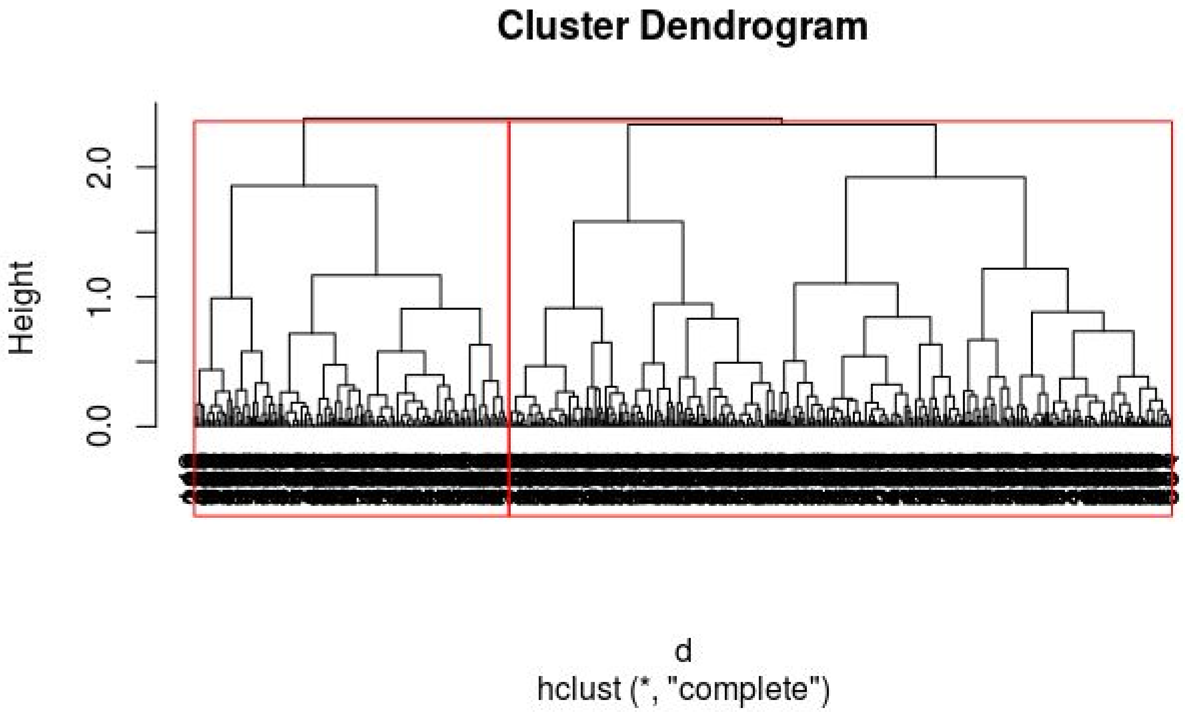


Q6)
Part 1:
Clustering

Dendogram for Single Link Clustering:

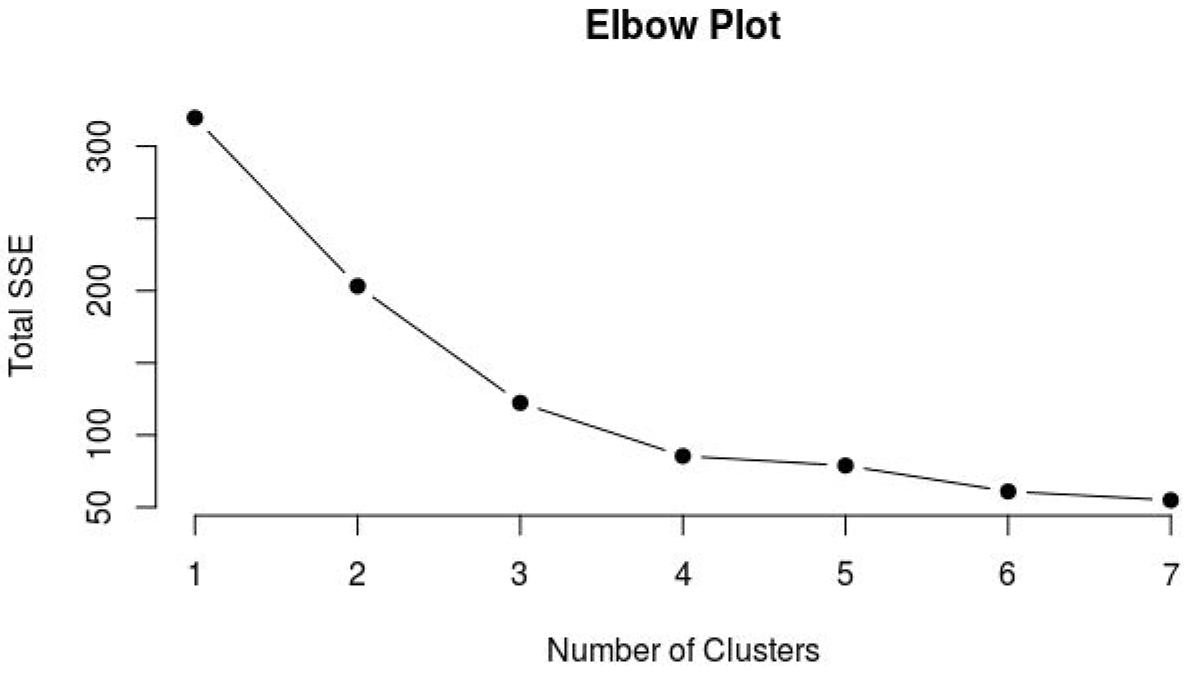


Dendogram for Complete Link Clustering:



Analysis 1:

The best value of K on the basis of the above plot is K=4. I feel this is the correct value because after that for further values of K like 5,6,7 the SSE curve flattens and there is less reduction in the values of SSE. Also there is a sharp drop in values of SSE till K=4 after which the plot flattens.



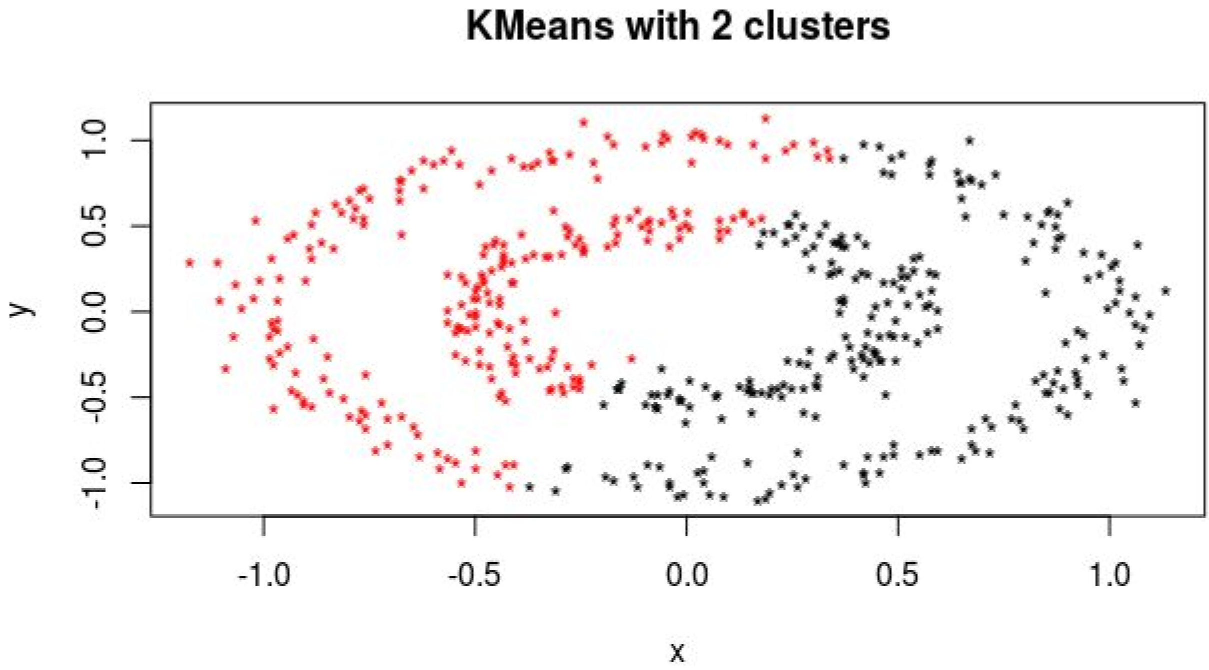
Analysis 2:
The values of SSE for the 3 type of Clustering techniques are :
"K Means SSE for given params = 203.042448463885"

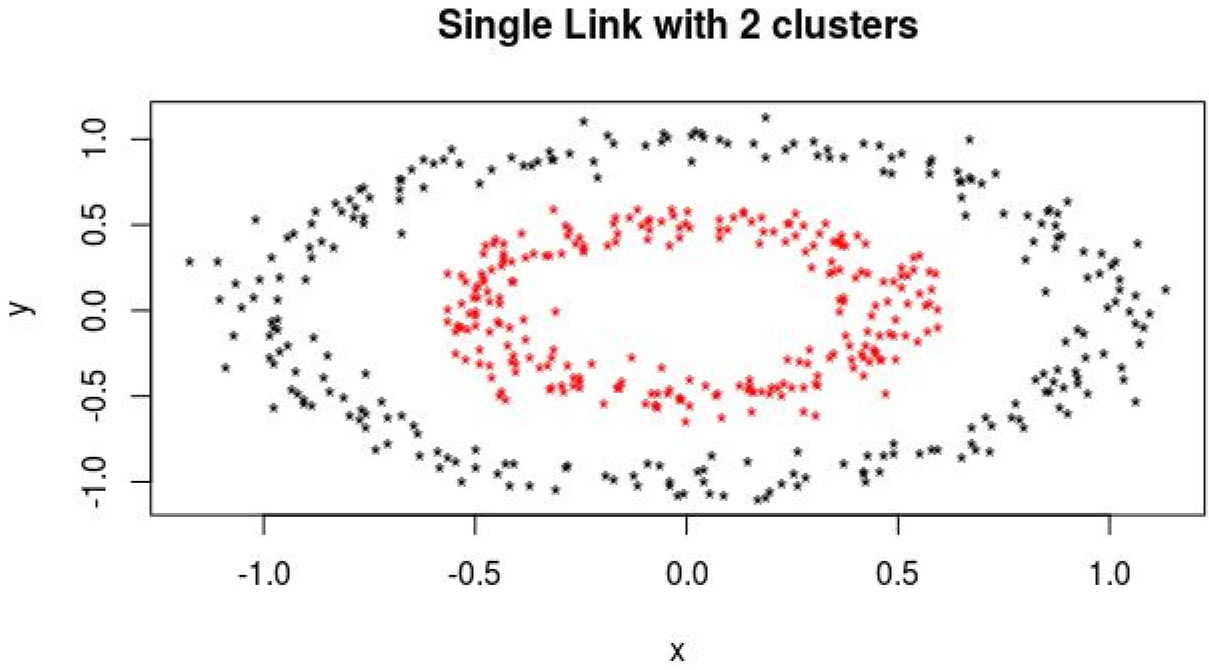
"Single link SSE for given params = 319.693456992432"

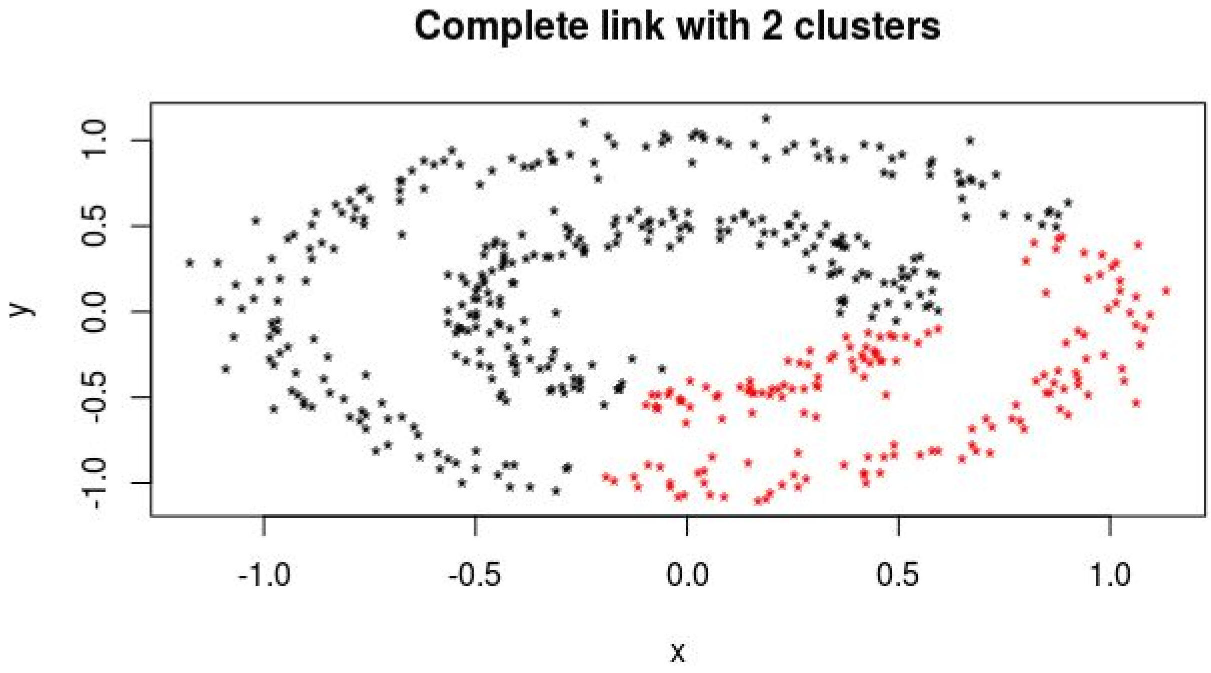
"Complete link SSE for given params = 221.04197449858"

Based on the values of SSE the Kmeans is the best Clustering algorithm

Analysis 3:


KMeans with 2 clusters
y
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
x


Single Link with 2 clusters
y
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
x



As we can see from the above clustering techniques, Single Link Clustering has performed the best because it was able to identify the specific 2 shapes in the data.

Analysis 4:

My answer for visual comparison through plots and SSE values is different. In order to identify the best clustering method it is important to analyse the plots to see how the data is spread and how it is clustered. Therefore numeric measures are not always reliable.

B - Classification

(iii)

B - Compare the kernels:

On comparing the visualization of the kernels and the accuracy values I would say that radial and polynomial kernels did better than the other models with equal accuracy values of 93.103. In the plot you can see that each of the models classified 2/29 values wrong.

When it comes to radial kernels, the two points closest to the second class were classified wrong whereas in polynomial kernels, one point closest to the second class and one instance farthest from the second class/ first class was classified wrong.

Either way, they both gave high accuracies of 93.103. Therefore, these models worked better than the other two.