TDT4300 - Assignment 3

Andreas B. Berg

09.04.2021

1 k-Means Clustering

> This is a programming part of the assignment. Your task is to implement the k-means clustering algorithm and assess the quality of the outputs by calculating Silhouette Coefficient.

See `k_means_clustering.ipynb` or `k_means_clustering.html` for the implementation.

The results were, if I dare say so myself, pretty decent, repeatedly achieving silhouette coefficients of around 0.73

2 Hierarchical Agglomerative Clustering (HAC)

a

Explain the Hierarchical Agglomerative Clustering (HAC) and the difference between MIN-link and MAX-link.

Hierarchical clustering works in one of two ways. Divisive hierarchical clustering starts by including every point in one large cluster, and then repeatedly splitting a cluster, until there are K clusters (or each point is its own cluster). Hierarchical agglomerative clustering works the opposite way - it starts out with each point as its own cluster, and then it repeatedly combines two clusters into one, until there are K (or only one) cluster(s).

MIN-link and MAX-link indicates how to calculate the distance between clusters. When using MIN-link, one takes the distance between the points of each cluster closest to each other (in other words, use the MINIMUM distance between two points, one in each cluster). MIN-link has the advantage that it can handle non-elliptical ("weird") shapes, while it is suffers when it comes to noise and outliers.

Using MAX-link, one takes the distance between the points in each cluster furthest from each other (the MAXIMUM distance between two points, one in each cluster). MAX-link is less impacted by noise and outliers, but it falls short when it comes to large or non-globular clusters.

b

You are given a two-dimensional dataset shown in Table 1. Perform HAC (for both MIN-link and MAX-link) and present the results in the form of dendrogram. Use the Euclidean distance. **Describe thoroughly the process and the outcome of each step.**

Table 1: Dataset for HAC

ID	x	y
A	4	3
B	5	8
C	5	7
D	9	2
E	11	6
F	14	8

HAC with MIN-link

The initial situation has each point as its own cluster, and a proximity matrix (with the distance between points). Using Euclidean distance, we get the following proximity matrix:

	A	B	C	D	E	F
A	-	5.099	4.123	5.099	7.616	11.180
B	5.099	-	1	7.211	6.324	9
C	4.123	1	-	6.403	6.083	9.055
D	5.099	7.211	6.403	-	4.472	7.810
E	7.616	6.324	6.083	4.472	-	3.606
F	11.180	9	9.055	7.810	3.606	-

While there are more than K (in this case, 1) cluster(s) - combine the two nearest clusters, and update the proximity matrix. As we use MIN-link, we keep the lowest of the two distances, for each step.

Combine B and C (distance = 1).

	A	B ∪ C	D	E	F
A	-	4.123	5.099	7.616	11.180
B ∪ C	4.123	-	6.403	6.083	9
D	5.099	6.403	-	4.472	7.810
E	7.616	6.083	4.472	-	3.606
F	11.180	9	7.810	3.606	-

Combine E and F (distance = 3.606).

	A	B ∪ C	D	E ∪ F
A	-	4.123	5.099	7.616
B ∪ C	4.123	-	6.403	6.083
D	5.099	6.403	-	4.472
E ∪ F	7.616	6.083	4.472	-

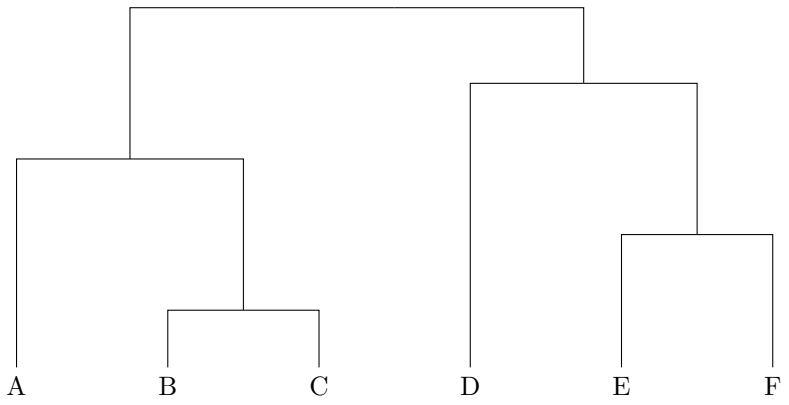
Combine A and $B \cup C$ (distance = 4.123).

	A ∪ B ∪ C	D	E ∪ F
A ∪ B ∪ C	-	5.099	6.083
D	5.099	-	4.472
E ∪ F	6.083	4.472	-

Combine D and $E \cup F$ (distance = 4.472).

	A ∪ B ∪ C	D ∪ E ∪ F
A ∪ B ∪ C	-	5.099
D ∪ E ∪ F	5.099	-

Combine the last clusters into one all-enveloping cluster. This gives the following dendrogram:



HAC with MAX-link

The initial situation has each point as its own cluster, and a proximity matrix (with the distance between points). Using Euclidean distance, we get the following proximity matrix:

	A	B	C	D	E	F
A	-	5.099	4.123	5.099	7.616	11.180
B	5.099	-	1	7.211	6.324	9
C	4.123	1	-	6.403	6.083	9.055
D	5.099	7.211	6.403	-	4.472	7.810
E	7.616	6.324	6.083	4.472	-	3.606
F	11.180	9	9.055	7.810	3.606	-

While there are more than K (in this case, 1) cluster(s) - combine the two nearest clusters, and update the proximity matrix. As we use MAX-link, we keep the highest of the two distances, for each step.

Combine B and C (distance = 1).

	A	B ∪ C	D	E	F
A	-	5.099	5.099	7.616	11.180
B ∪ C	5.099	-	7.211	6.324	9.055
D	5.099	7.211	-	4.472	7.810
E	7.616	6.324	4.472	-	3.606
F	11.180	9.055	7.810	3.606	-

Combine E and F (distance = 3.606).

	A	B ∪ C	D	E ∪ F
A	-	5.099	5.099	11.180
B ∪ C	5.099	-	7.211	9.055
D	5.099	7.211	-	7.810
E ∪ F	11.180	9.055	7.810	-

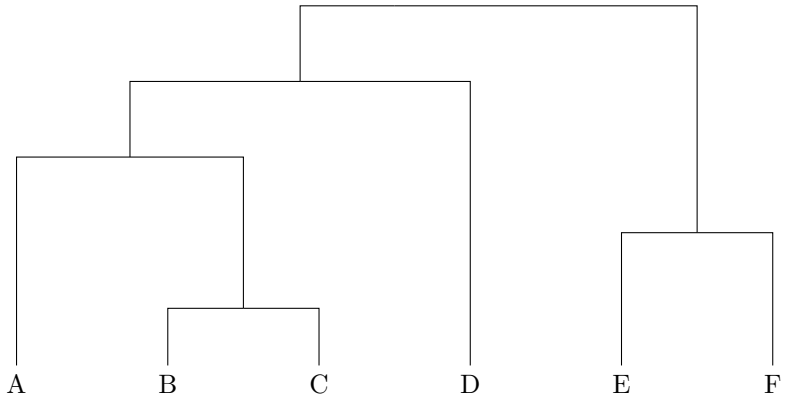
Combine A and $B \cup C$ (distance = 5.099). Note that there are two combinations of clusters that have the exact same distance - A and $B \cup C$ and A and D. According to my sources, one would then choose a pair at random. While not completely random, I chose this pairing first, as they are first alphabetically.

	A ∪ B ∪ C	D	E ∪ F
A ∪ B ∪ C	-	7.211	11.180
D	7.211	-	7.810
E ∪ F	11.180	7.810	-

Combine D and $A \cup B \cup C$ (distance = 5.099).

	A ∪ B ∪ C ∪ D	E ∪ F
A ∪ B ∪ C	-	11.180
E ∪ F	11.180	-

Combine the last clusters into one all-enveloping cluster. This gives the following dendrogram:

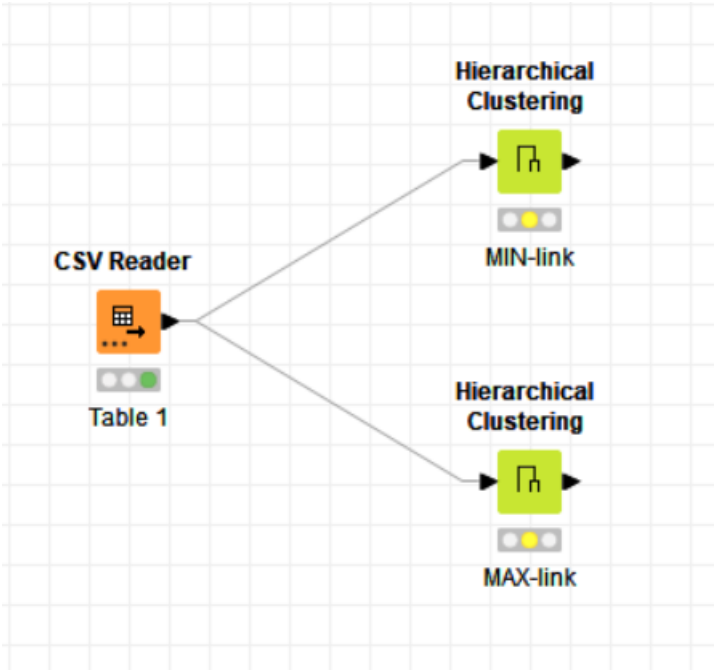


It is clear that we get slightly different results when using MIN-link, compared to MAX-link.

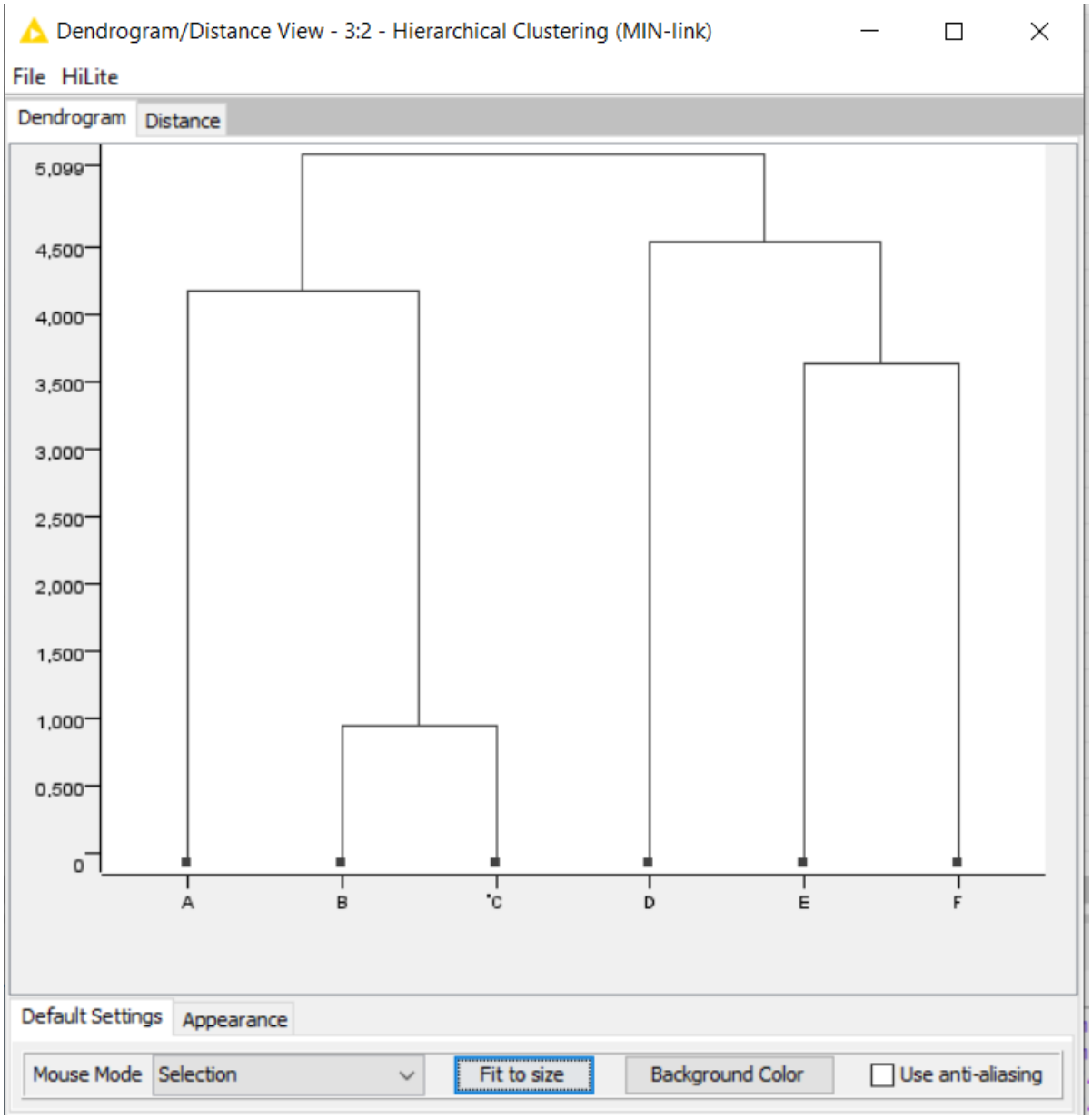
c

Verify your results using the KNIME data analytics platform. For clarification, MIN-link and MAX-link is in KNIME referred as SINGLE and COMPLETE linkage methods. **Present a picture of your workflow and the dendrograms.**

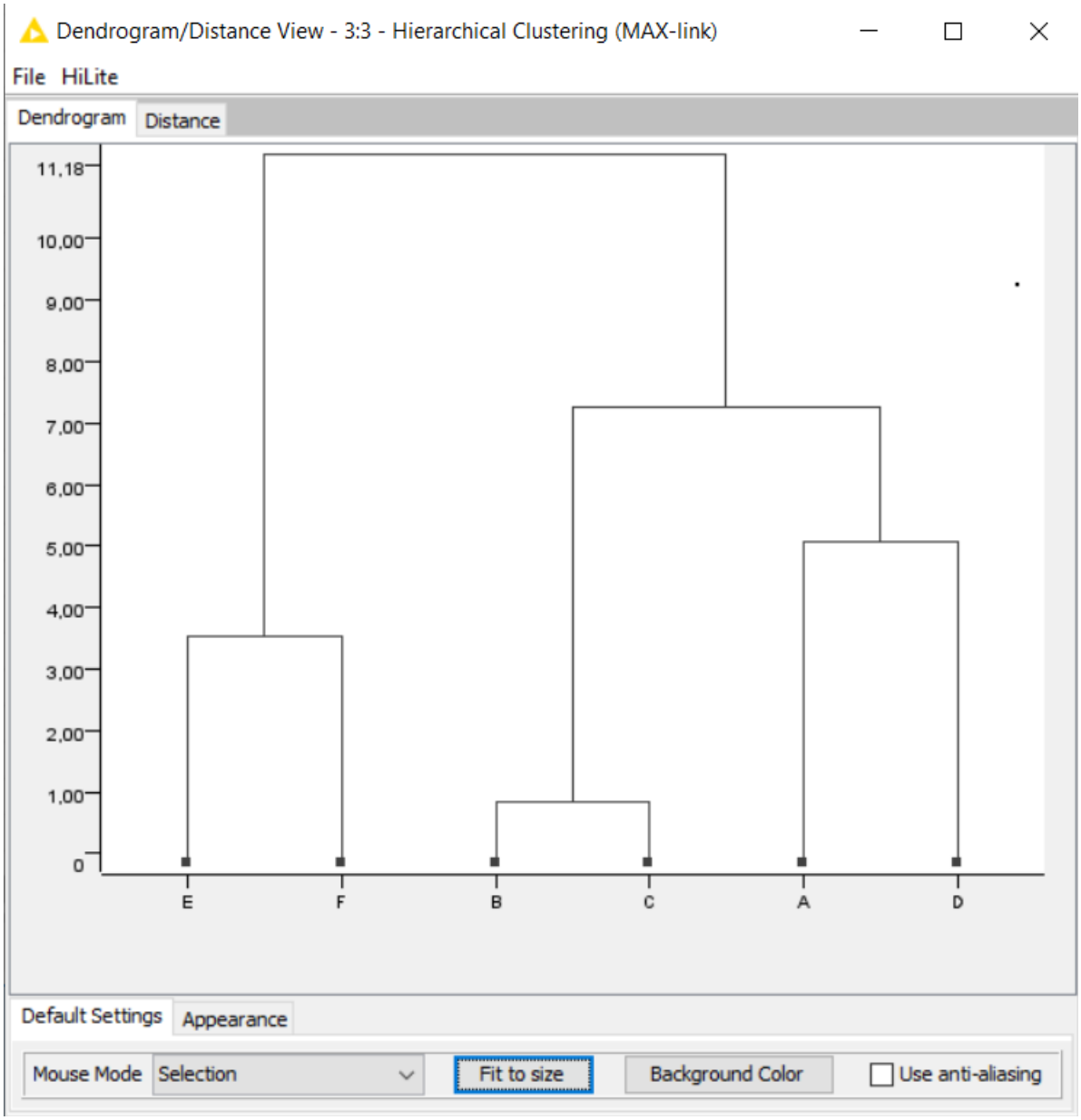
Knime workflow



MIN-link dendrogram



MAX-link dendrogram



The only difference between my dendrograms and that KNIME chose to combine A and D instead of A and $B \cup C$ in step 3 in using MAX-link. As one selects the pair to combine randomly, neither is preferable to the other.

3 DBSCAN Clustering

You are given following points:

$$
\begin{aligned}
P_1 &= (1,1) \\
P_2 &= (3,3) \\
P_3 &= (3,4) \\
P_4 &= (2,4) \\
P_5 &= (6,5) \\
P_6 &= (7,6) \\
P_7 &= (7,8) \\
P_8 &= (6,10) \\
P_9 &= (12,4) \\
P_{10} &= (5,11) \\
P_{11} &= (6,11) \\
P_{12} &= (5,10) \\
P_{13} &= (16,8) \\
P_{14} &= (11,9) \\
P_{15} &= (13,8) \\
P_{16} &= (10,7) \\
P_{17} &= (12,8) \\
P_{18} &= (15,3)
\end{aligned}
$$

a

Your task is to perform DBSCAN clustering given the parameters $Eps = 2$ (Euclidean metric) and $MinPts = 3$ (including the analyzed point). Identify core, border and noise points. Identify clusters. **Describe thoroughly the process and the outcome of each step.**

Euclidean distance calculations give us the distance matrix found on the following page. Note that all pairs with distance less than or equal to $Eps = 2$ are marked green. This, in turn, gives us the following table:

Point	Points within border ($Eps = 2$)	Number of points within border	Point classification
P_1	P_1	1	Noise
P_2	P_2, P_3, P_4	3	Core
P_3	P_2, P_3, P_4	3	Core
P_4	P_2, P_3, P_4	3	Core
P_5	P_5, P_6	2	Border
P_6	P_5, P_6, P_7	3	Core
P_7	P_6, P_7	2	Border
P_8	$P_8, P_{10}, P_{11}, P_{12}$	4	Core
P_9	P_9	1	Noise
P_{10}	$P_8, P_{10}, P_{11}, P_{12}$	4	Core
P_{11}	$P_8, P_{10}, P_{11}, P_{12}$	4	Core
P_{12}	$P_8, P_{10}, P_{11}, P_{12}$	4	Core
P_{13}	P_{13}	1	Noise
P_{14}	P_{14}, P_{17}	2	Border
P_{15}	P_{15}, P_{17}	2	Border
P_{16}	P_{16}	1	Noise
P_{17}	P_{14}, P_{15}, P_{17}	3	Core
P_{18}	P_{18}	1	Noise

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}
P_1	0,00	2,83	3,61	3,16	6,40	7,81	9,22	10,30	11,40	10,77	11,18	9,85	16,55	12,81	13,89	10,82	13,04	14,14
P_2	2,83	0,00	1,00	1,41	3,61	5,00	6,40	7,62	9,06	8,25	8,54	7,28	13,93	10,00	11,18	8,06	10,30	12,00
P_3	3,61	1,00	0,00	1,00	3,16	4,47	5,66	6,71	9,00	7,28	7,62	6,32	13,60	9,43	10,77	7,62	9,85	12,04
P_4	3,16	1,41	1,00	0,00	4,12	5,39	6,40	7,21	10,00	7,62	8,06	6,71	14,56	10,30	11,70	8,54	10,77	13,04
P_5	6,40	3,61	3,16	4,12	0,00	1,41	3,16	5,00	6,08	6,08	6,00	5,10	10,44	6,40	7,62	4,47	6,71	9,22
P_6	7,81	5,00	4,47	5,39	1,41	0,00	2,00	4,12	5,39	5,39	5,10	4,47	9,22	5,00	6,32	3,16	5,39	8,54
P_7	9,22	6,40	5,66	6,40	3,16	2,00	0,00	2,24	6,40	3,61	3,16	2,83	9,00	4,12	6,00	3,16	5,00	9,43
P_8	10,30	7,62	6,71	7,21	5,00	4,12	2,24	0,00	8,49	1,41	1,00	1,00	10,20	5,10	7,28	5,00	6,32	11,40
P_9	11,40	9,06	9,00	10,00	6,08	5,39	6,40	8,49	0,00	9,90	9,22	9,22	5,66	5,10	4,12	3,61	4,00	3,16
P_{10}	10,77	8,25	7,28	7,62	6,08	5,39	3,61	1,41	9,90	0,00	1,00	1,00	11,40	6,32	8,54	6,40	7,62	12,81
P_{11}	11,18	8,54	7,62	8,06	6,00	5,10	3,16	1,00	9,22	1,00	0,00	1,41	10,44	5,39	7,62	5,66	6,71	12,04
P_{12}	9,85	7,28	6,32	6,71	5,10	4,47	2,83	1,00	9,22	1,00	1,41	0,00	11,18	6,08	8,25	5,83	7,28	12,21
P_{13}	16,55	13,93	13,60	14,56	10,44	9,22	9,00	10,20	5,66	11,40	10,44	11,18	0,00	5,10	3,00	6,08	4,00	5,10
P_{14}	12,81	10,00	9,43	10,30	6,40	5,00	4,12	5,10	5,10	6,32	5,39	6,08	5,10	0,00	2,24	2,24	1,41	7,21
P_{15}	13,89	11,18	10,77	11,70	7,62	6,32	6,00	7,28	4,12	8,54	7,62	8,25	3,00	2,24	0,00	3,16	1,00	5,39
P_{16}	10,82	8,06	7,62	8,54	4,47	3,16	3,16	5,00	3,61	6,40	5,66	5,83	6,08	2,24	3,16	0,00	2,24	6,40
P_{17}	13,04	10,30	9,85	10,77	6,71	5,39	5,00	6,32	4,00	7,62	6,71	7,28	4,00	1,41	1,00	2,24	0,00	5,83
P_{18}	14,14	12,00	12,04	13,04	9,22	8,54	9,43	11,40	3,16	12,81	12,04	12,21	5,10	7,21	5,39	6,40	5,83	0,00

All core points have at least $MinPts = 3$ points within its border, itself included. Observe that we have 4 border points. These are points with fewer than $MinPts = 3$ points within its border, but they are connected to a core point - P_5 and P_7 are connected to P_6, while P_{14} and P_{15} are connected to P_{17}. The remaining points are considered noise.

To identify clusters, we traverse down the list of core points. All connected core points (meaning they are within each other's border) will be clustered together. As all our border points are connected to only one core point, these will be clustered with their connected core point. All noise points will be eliminated. This gives us the following clusters:

$$P_2, P_3, P_4$$

$$P_5, P_6, P_7$$

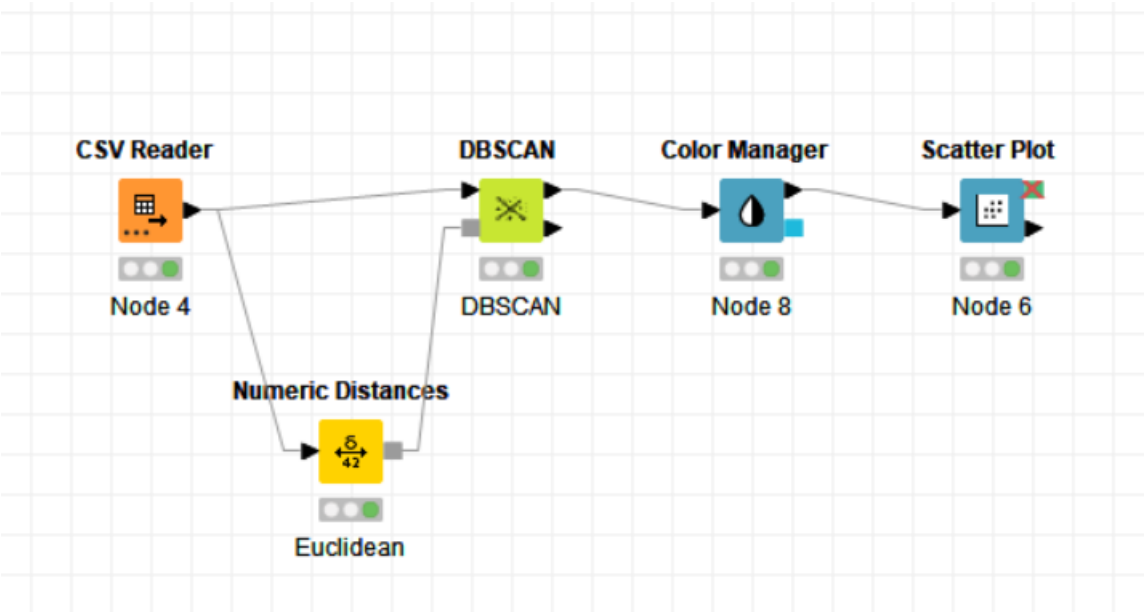
$$P_8, P_{10}, P_{11}, P_{12}$$

$$P_{14}, P_{15}, P_{17}$$

b

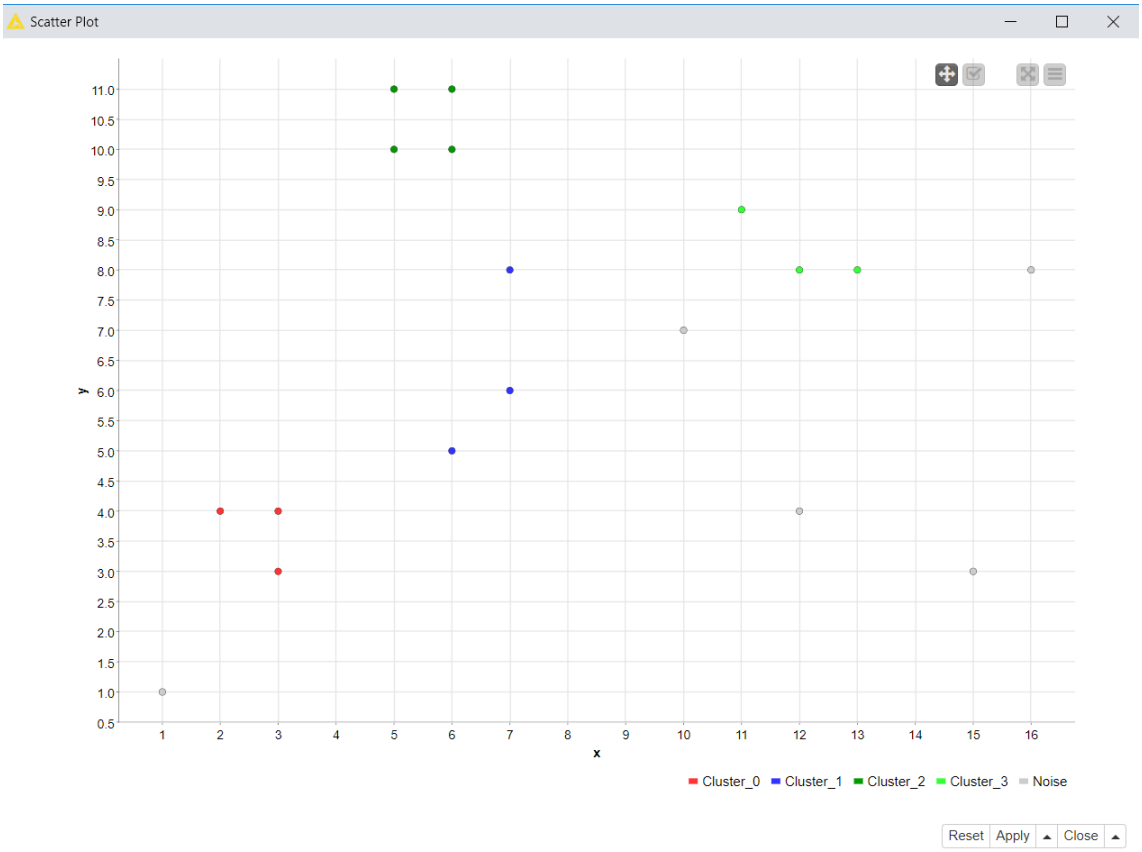
Verify your results using the KNIME data analytics platform. **Present a picture of your workflow and the scatter plot with marked clusters and outliers.**

KNIME workflow



The Euclidean Distance-node is used to calculate distances between points. The Color Manager is used to color the scatter plot, which is gotten from the Scatter Plot-node.

Scatter plot



It's a little hard to see, at least to my colorblind eye, but the noise points are marked gray, while the clusters are colored in their own colors. The clusters coincide with my results.