TDT4300 Datawarehouse and datamining

Clustering

Assignment 3

Group: 99

Name: Erik Turøy Midtun

Submission date: March 3, 2021

Group 99
Erik Turøy Midtun

Task 1: KMeans

See attached file: *k_means_clustering.ipynb*

Task 2: Hierarchical Agglomerative Clustering (HAC)

a) Hierarchical Agglomerative Clustering is a bottom-up approach for hierarchial clustering. It works by starting with all points in their own clusters and then merge the clusters that are close to eachother.

MIN-link uses the minimum distance between a member of two clusters as the new distance
MAX-link uses the maximum distance between a member of two clusters as the new distance

b) Calculate the Euclidean distance between all data points, the distance is the same from A to B as from B to A:

	A	B	C	D	E
A		4.123106	4.123106	1	7.211103
B			7.071068	3.162278	7
C				4.472136	5.385165
D					6.708204
E					

Start with the MIN-link method: The smallest distance is from A to D, so we merge them.

	B	C	A, D	E
A, D	3.162278	4.123106	0	6.708204
B		7.071068	3.162278	7
C			4.123106	5.385165
E				

	A, D, B	C	E
A, D, B		4.123106	6.708204
C			7
E			

	A, D, B, C	E
A, D, B, C		6.708204
E		

Thus we have clustered all points.

For MAX-link we still choose the smallest value when clustering, but use the greatest distance when merging

	A	B	C	D	E
A		4.123106	4.123106	1	7.211103
B			7.071068	3.162278	7
C				4.472136	5.385165

D					6.708204
E					

We merge A and D

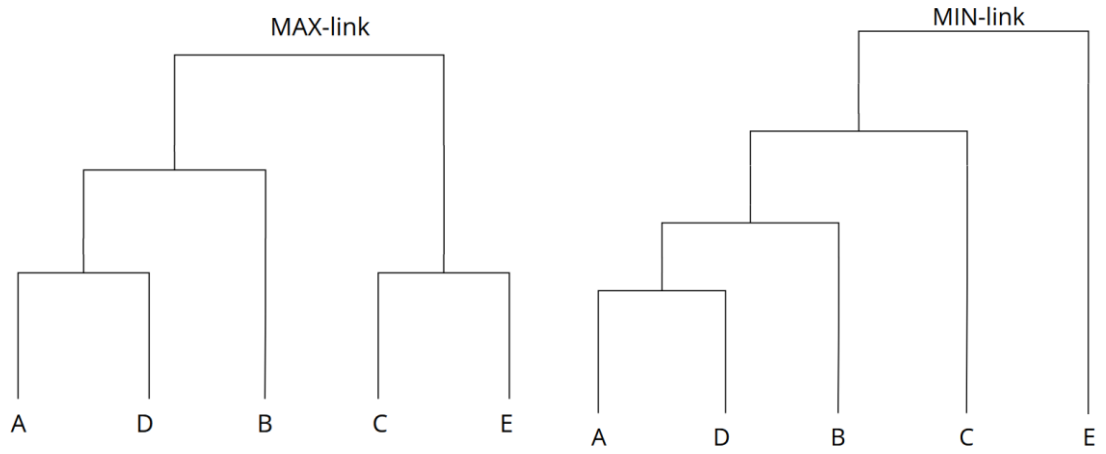
	A,D	B	C	E
A, D		4.123106	4.472136	7.211103
B			7.071068	7
C				5.385165
E				

We merge A, D with B

	A,D,B	C	E
A, D, B		7.071068	7.211103
C			5.385165
E			

The smallest is C to E and we merge those to:

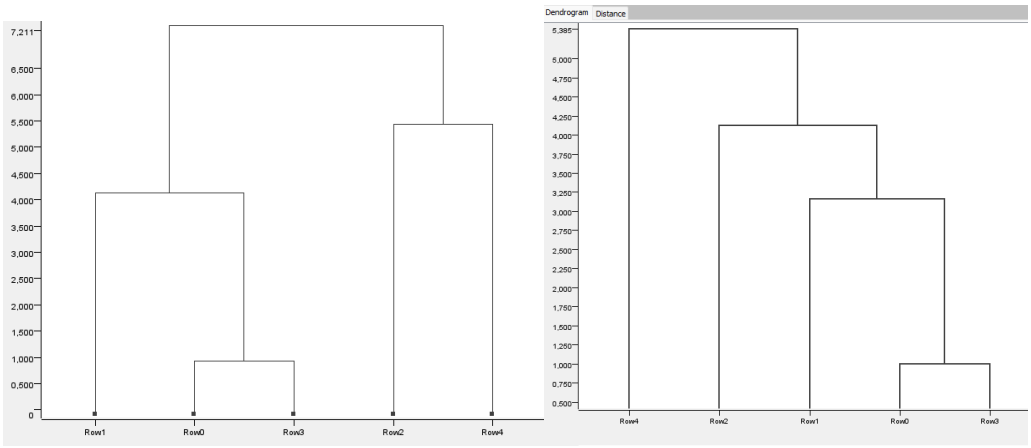
	A,D,B	C, E
A, D, B		7.211103
C, E		



Dendograms for MIN and MAX-link HAC

KNIME-workflow:

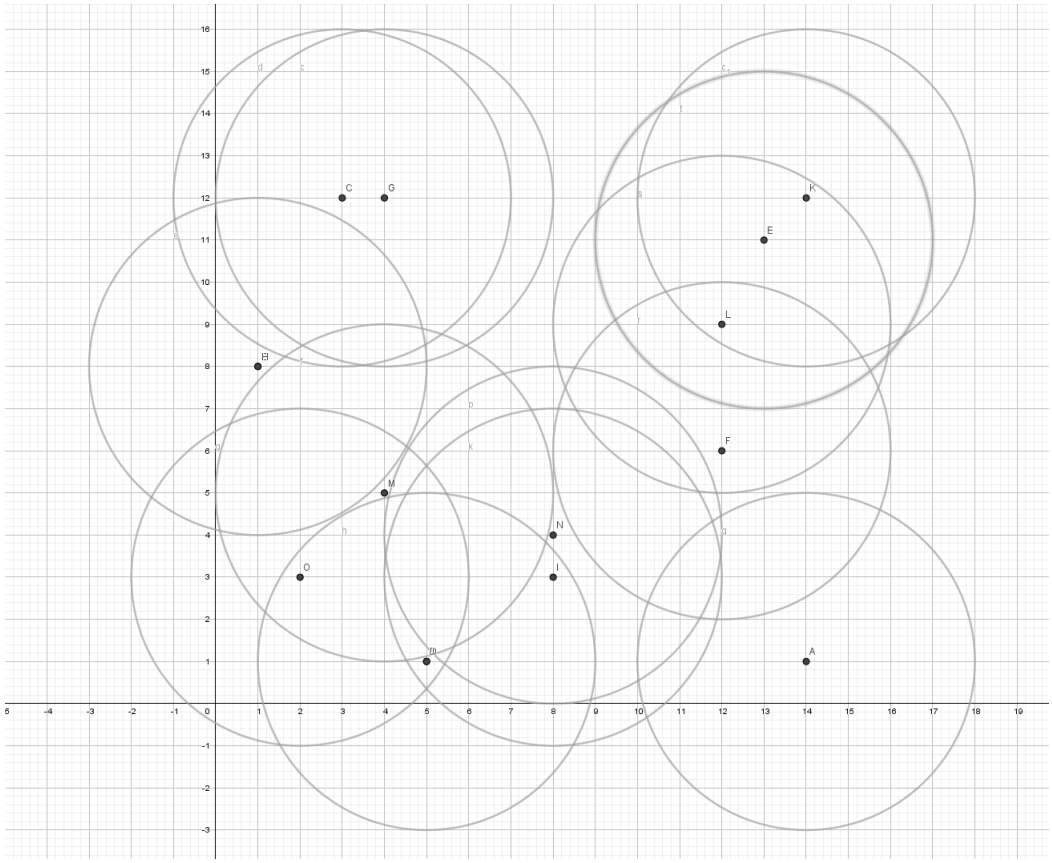
KNIME-SINGLE-linkage (MIN-linkage) and COMPLETE-linkage (MAX-linkage) output:



If we substitute the Row<number> with the input data, we see that these results correspond with we found by hand above.

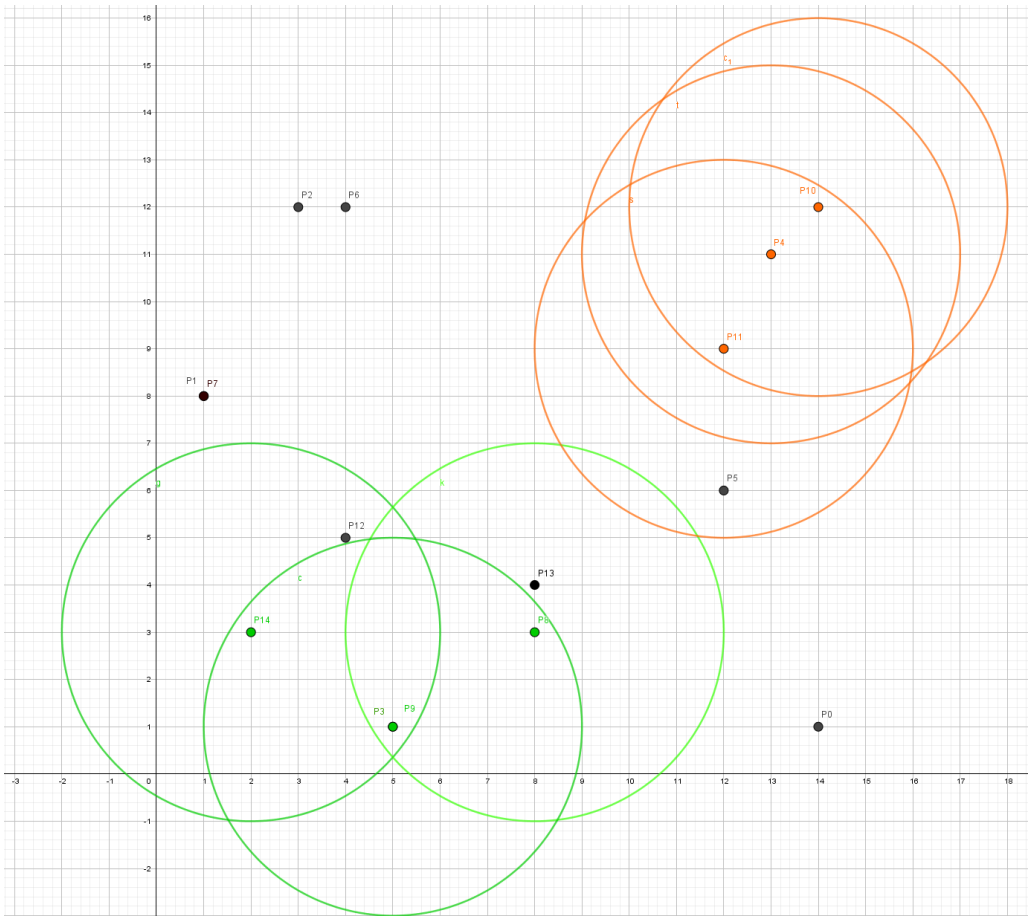
Task 3: DBSCAN Clustering

I did the task in a plotting tool called GeoGebra because it was easy to visualize, and the task did not specify how to do the clustering. I plotted all the points and drew a circle with radius equal to the Eps=4. This resulted in the following plot:

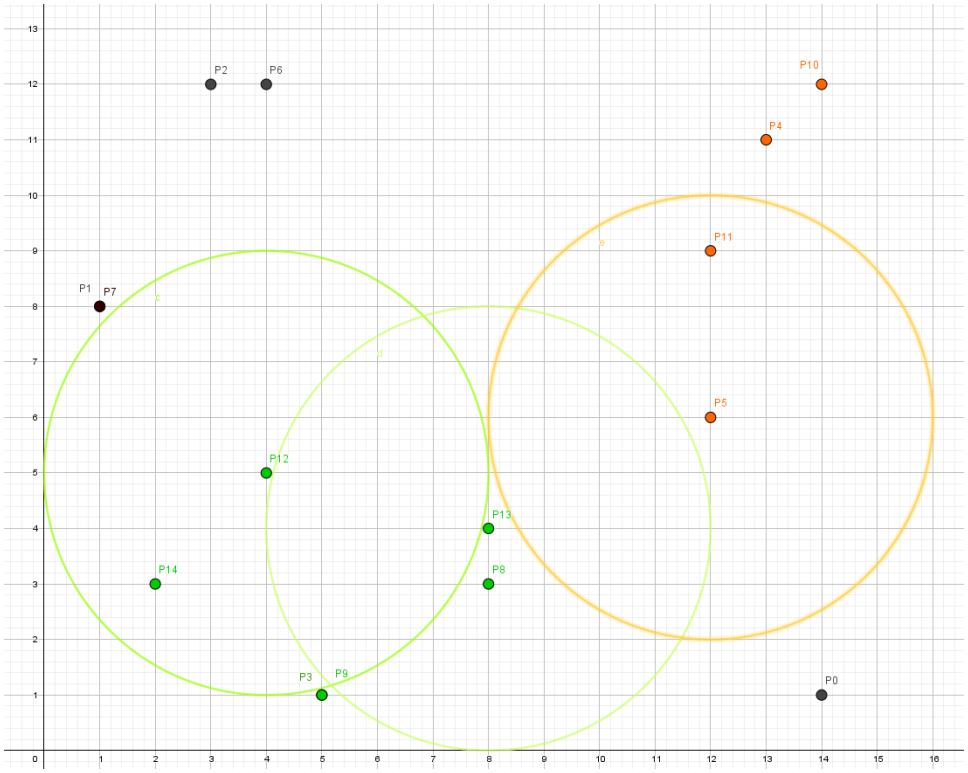


For each point I then counted the number of points inside the point's circle. If the circle contains 3 or more points, I put a color on it. Else I remove the circle and the point remains black. This resulted in the following plot

Group 99
Erik Turøy Midtun



The points colored green, and orange is core points of two different clusters. We then must mark the border points:



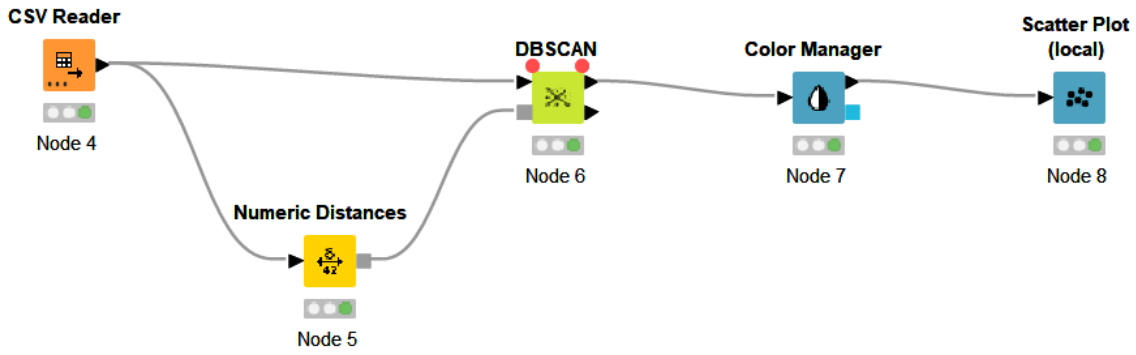
The border points are the points within a core points eps, but it does not satisfy minPts = 3.

The points colored black are thus noise points.

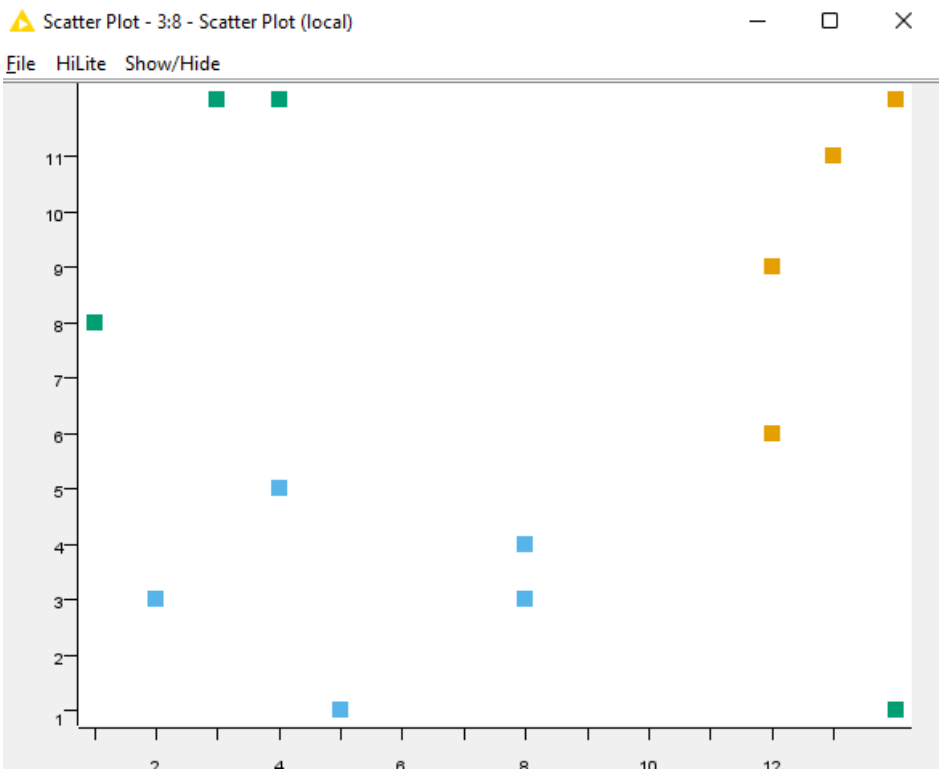
Now we have final two clusters, one marked with green, and the other with orange.

b)

KNIME workflow:



Output:



This verifies our results in 3a.