

## ΠΡΟΒΛΗΜΑ 1

### 1.

$a(0) - b(3) - c(5) - d(8) - e(10) - f(12) - g(16) - h(17) - i(19) - j(20)$	(Βήμα 1)
$a(0) - b(3) - c(5) - d(8) - e(10) - f(12) - g(16) - h(17) - ij(19,20)$	(Βήμα 2)
$a(0) - b(3) - c(5) - d(8) - e(10) - f(12) - gh(16,17) - ij(19,20)$	(Βήμα 3)
$a(0) - b(3) - c(5) - d(8) - e(10) - f(12) - ghij(16,17,19,20)$	(Βήμα 4)
$a(0) - b(3) - c(5) - d(8) - ef(10,12) - ghij(16,17,19,20)$	(Βήμα 5)
$a(0) - b(3) - c(5) - def(8,10,12) - ghij(16,17,19,20)$	(Βήμα 6)
$a(0) - bc(3,5) - def(8,10,12) - ghij(16,17,19,20)$	(Βήμα 7)
$a(0) - bcdef(3,5,8,10,12) - ghij(16,17,19,20)$	(Βήμα 8)
$abcdef(0,3,5,8,10,12) - ghij(16,17,19,20)$	(Βήμα 9)
$abcdefghij(0,3,5,8,10,12,16,17,19,20)$	(Βήμα 10)

### 2.

$a(0) - b(3) - c(5) - d(8) - e(10) - f(12) - g(16) - h(17) - i(19) - j(20)$	(Βήμα 1)
$a(0) - b(3) - c(5) - d(8) - e(10) - f(12) - g(16) - h(17) - ij(19,20)$	(Βήμα 2)
$a(0) - b(3) - c(5) - d(8) - e(10) - f(12) - gh(16,17) - ij(19,20)$	(Βήμα 3)
$a(0) - b(3) - c(5) - d(8) - ef(10,12) - gh(16,17) - ij(19,20)$	(Βήμα 4)
$a(0) - bc(3,5) - d(8) - ef(10,12) - gh(16,17) - ij(19,20)$	(Βήμα 5)
$a(0) - bc(3,5) - d(8) - ef(10,12) - ghij(16,17,19,20)$	(Βήμα 6)
$a(0) - bc(3,5) - def(8,10,12) - ghij(16,17,19,20)$	(Βήμα 7)
$abc(0,3,5) - def(8,10,12) - ghij(16,17,19,20)$	(Βήμα 8)
$abc(0,3,5) - defghij(8,10,12,16,17,19,20)$	(Βήμα 9)
$abcdefghij(0,3,5,8,10,12,16,17,19,20)$	(Βήμα 10)

### 3.

Βήμα 1: cluster1 centroid=0 cluster2 centroid=3 cluster3 centroid=8

ανάθεση σημείων: C1(a) C2(b,c) C3(d,e,f,g,h,i,j)

Βήμα 2: cluster1 centroid=0 cluster2 centroid=4 cluster3 centroid=14.6

ανάθεση σημείων: C1(a) C2(b,c,d) C3(e,f,g,h,i,j)

Βήμα 3: cluster1 centroid=0 cluster2 centroid=5.33 cluster3 centroid=15.67

ανάθεση σημείων: C1(a) C2(b,c,d,e) C3(f,g,h,i,j)

Βήμα 4: cluster1 centroid=0 cluster2 centroid=6.5 cluster3 centroid=16.8

ανάθεση σημείων: C1(a,b) C2(c,d,e) C3(f,g,h,i,j)

Βήμα 5: cluster1 centroid=1.5 cluster2 centroid=7.67 cluster3 centroid=16.8

ανάθεση σημείων: C1(a,b) C2(c,d,e,f) C3(g,h,i,j)

Βήμα 6: cluster1 centroid=1.5 cluster2 centroid=8.75 cluster3 centroid=18

ανάθεση σημείων: C1(a,b,c) C2(d,e,f) C3(g,h,i,j)

Βήμα 7: cluster1 centroid=2.67 cluster2 centroid=10 cluster3 centroid=18

ανάθεση σημείων: C1(a,b,c) C2(d,e,f) C3(g,h,i,j)

## ΠΡΟΒΛΗΜΑ 2

**K-means:**Υπάρχουν τα αντίστοιχα αποτελέσματα στο φάκελο k-means.

### 1)seed=10

Σε αυτή τη περίπτωση κατηγοριοποιεί 8 από τα 63 ως λάθος και έχει ποσοστό λάθους  $8/63 * 100 = 12.7\%$ .

### 2)seed=4

Σε αυτή τη περίπτωση κατηγοριοποιεί 15 από τα 63 ως λάθος και έχει ποσοστό λάθους  $15/63 * 100 = 23.8\%$ . Επίσης, ο cluster3 είναι απροσδιόριστος.

### 3)seed=2 (Βέλτιστο)

Σε αυτή τη περίπτωση ο αλγόριθμος k-mean δεν κατηγοριοποιεί κανένα ως λάθος.

### 4)seed=7

Στην περίπτωση αυτή κατηγοριοποιεί 13 από τα 63 ως λάθος και έχει ποσοστό λάθους  $13/63 * 100 = 20.63\%$ . Επίσης, ο cluster2 είναι απροσδιόριστος.

### 5)seed=15 (Βέλτιστο)

Σε αυτή τη περίπτωση ο αλγόριθμος k-mean δεν κατηγοριοποιεί κανένα ως λάθος.

**Hierarchical: Υπάρχουν τα αντίστοιχα αποτελέσματα στο φάκελο Hierarchical.**

**1)LinkType:Single**

Σε αυτή τη περίπτωση κατηγοριοποιεί 13 ως λάθος και έχει ποσοστό λάθους 20.63%.Επίσης cluster3 είναι απροσδιόριστος.

**2)LinkType:Complete**

Σε αυτή τη περίπτωση κατηγοριοποιεί 9 ως λάθος και έχει ποσοστό λάθους 14.28%.

**3)LinkType:Average(Βέλτιστο)**

Σε αυτή τη περίπτωση ο αλγόριθμος **Hierarchical** δεν κατηγοριοποιεί κανένα ως λάθος.

**4)LinkType:MEAN**

Σε αυτή τη περίπτωση κατηγοριοποιεί 2 ως λάθος και έχει ποσοστό λάθους 3.17%.

**5)LinkType:CENTROID**

Σε αυτή τη περίπτωση κατηγοριοποιεί 34 ως λάθος και έχει ποσοστό λάθους 53.96%.Επίσης έχει δύο clusters με απροσδιόριστες τιμές class.

**6)LinkType:WARD(Βέλτιστο)**

Σε αυτή τη περίπτωση ο αλγόριθμος **Hierarchical** δεν κατηγοριοποιεί κανένα ως λάθος.

**7)LinkType:ADJCOMPLETE**

Σε αυτή τη περίπτωση κατηγοριοποιεί 39 ως λάθος και έχει ποσοστό λάθους 61.9%. Επίσης έχει δύο clusters με απροσδιόριστες τιμές class.

**8)LinkType:NEIGHBOR\_JOINING**

Σε αυτή τη περίπτωση κατηγοριοποιεί 40 ως λάθος και έχει ποσοστό λάθους 63.49%.Επίσης κατηγοριοποιεί τα δεδομένα μόνο σε ένα cluster.

**ΣΥΜΠΕΡΑΣΜΑ:**

Και οι δύο αλγόριθμοι έχουν βέλτιστες κατηγοριοποιήσεις , παρόλα αυτά βλέποντας τα αποτελέσματα των δύο αυτών αλγορίθμων παρατηρούμε πως ο Hierarchical σημειώνει πολύ μεγαλύτερα ποσοστά λάθους στις περιπτώσεις που δεν είναι βέλτιστος και σε κάποιες από αυτές έχει clusters με απροσδιόριστη τιμή class ενώ από την άλλη ο K-means σημειώνει πολύ μικρότερα ποσοστά λάθους και σε ελάχιστες περιπτώσεις μπορεί να αφήσει απροσδιόριστη τη τιμή κάποιου cluster.Όλα αυτά οδηγούν στο συμπέρασμα πως ο k-means ερμηνεύει καλύτερα το συγκεκριμένο data set από τον Hierarchical.

### **ΠΡΟΒΛΗΜΑ 3**

**1.**

1: (1,12) -> NOISE

2: (2,5,9)-> NOISE

3: (3,8,9)-> BORDER

4: (4,6,7)-> NOISE

5: (2,5,9)-> NOISE

6: (4,5,6)-> NOISE

7: (4,7,11)->NOISE

8: (3,8,10,11)-> CORE

9: (2,3,9)-> NOISE

10: (8,10)-> BORDER

11: (7,8,11)-> BORDER

12: (1,12)-> NOISE

CLUSTER0={8,3,10,11}

**2.**

Το αντίστοιχο αποτέλεσμα του DBSCAN στο weka υπάρχει στο φάκελο DBSCAN.

Όπως θα δείτε ο αλγόριθμος DBSCAN στο weka έβγαλε τα ίδια αποτελέσματα με αυτά που έβγαλα χρησιμοποιώντας τον αλγόριθμο DBSCAN στο χέρι.