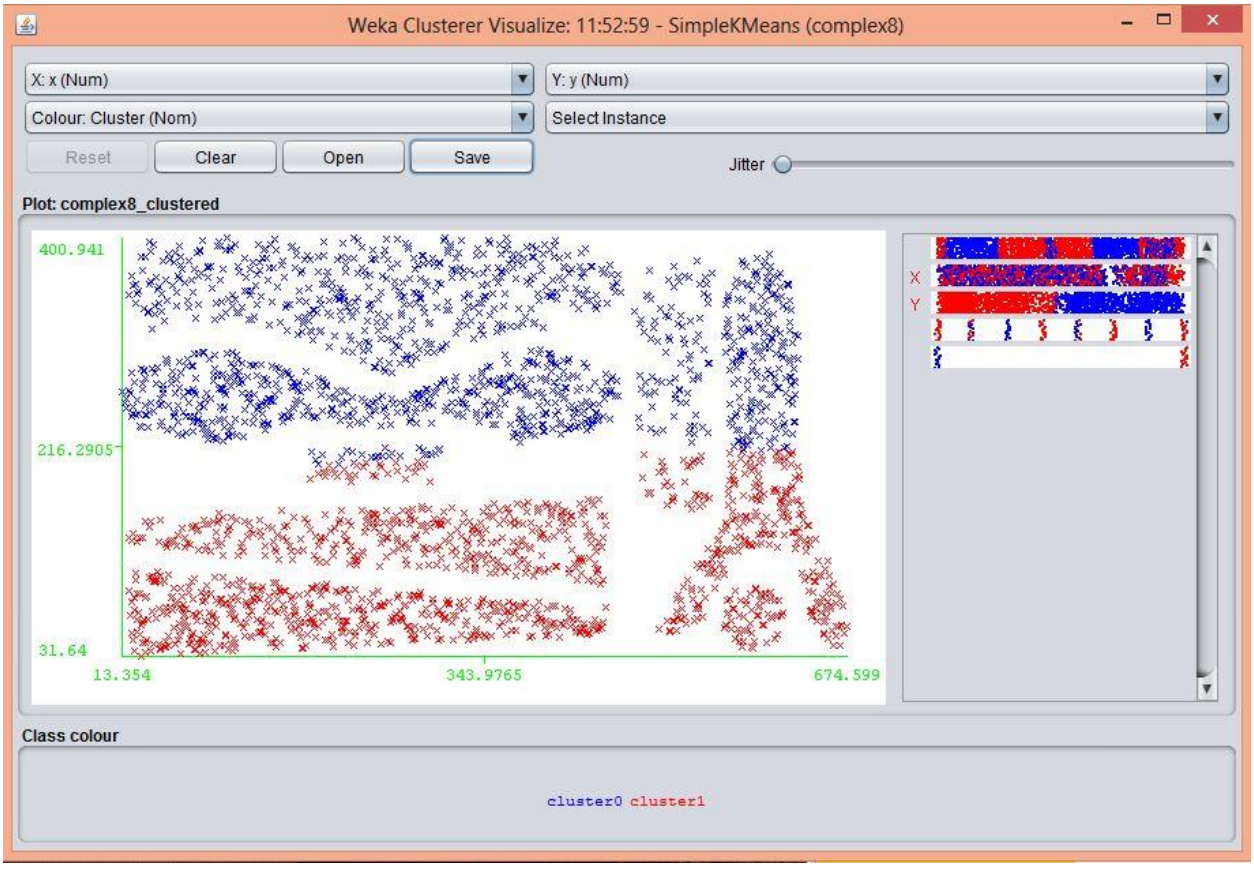
Εξόρυξη δεδομένων και αλγόριθμοι μάθησης
Εαρινό εξάμηνο 2017
Project Weka (part B)

ΑΣΚΗΣΗ 1 – Simple Clustering

1. SimpleKMeans με default παραμέτρους



K-Means cluster	Real Clusters
0	{1,2,4,6,7}
1	{0,1,3,4,5,7}

2. Αρχές λειτουργίας του k-means

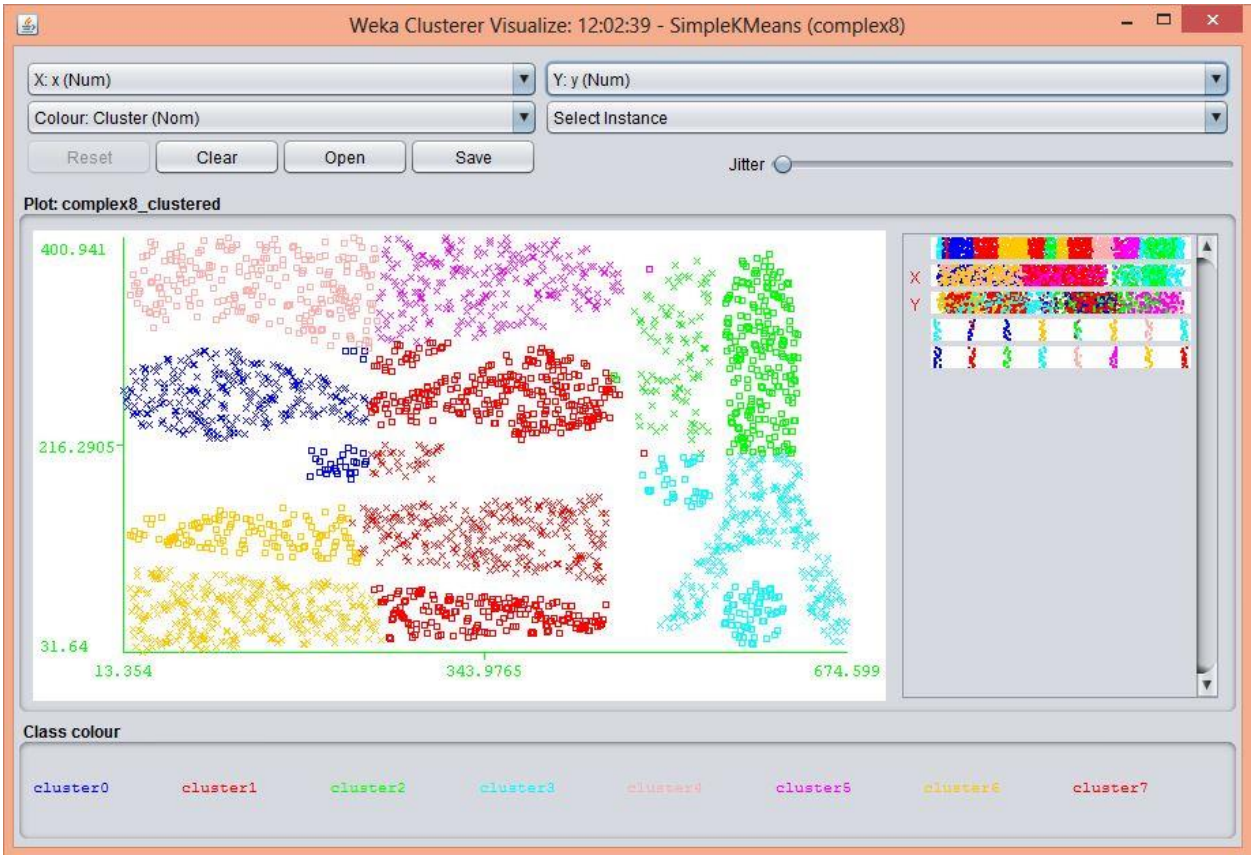
Βήμα 1: Αρχικά επιλέγεται ένα σύνολο από cluster centers τυχαία.

Βήμα 2: Κάθε στοιχείο ανατίθεται στο πιο κοντινό του cluster center.

Βήμα 3: Υπολογισμός των νέων κέντρων των συστάδων(αθροίζοντας τις τιμές των στοιχείων ενός cluster και διαιρώντας με το πλήθος των στοιχείων του cluster).

Βήμα 4 : Επανάληψη βημάτων 2,3 .

3. SimpleKMeans με numClusters 8



Από τον result buffer βλέπουμε ότι η απόδοση του δεν είναι τόσο καλή αφού έχει 1137(44.5708 %) στοιχεία σε λάθος cluster.

Ο k-means αλγόριθμος ελαχιστοποιεί την μέση τετραγωνική απόσταση των στοιχείων από τα πλησιέστερα cluster centers πράγμα που φαίνεται και στην παραπάνω απεικόνιση αφού οι συστάδες που έχει δημιουργήσει είναι πιο ‘συμπαγείς’. Έχει δημιουργήσει ,δηλαδή, clusters σύμφωνα με την απόσταση των στοιχείων από ένα κεντροειδές σχηματίζοντας έτσι clusters με περίπου το ίδιο μέγεθος και σχήμα.

Ωστόσο, το dataset που έχουμε βλέπουμε ότι έχει μορφές συστάδων με διαφορετική πυκνότητα στοιχείων και σχήματος, οδηγώντας έτσι τον simple k-means σε μια όχι τόσο καλή απόδοση.

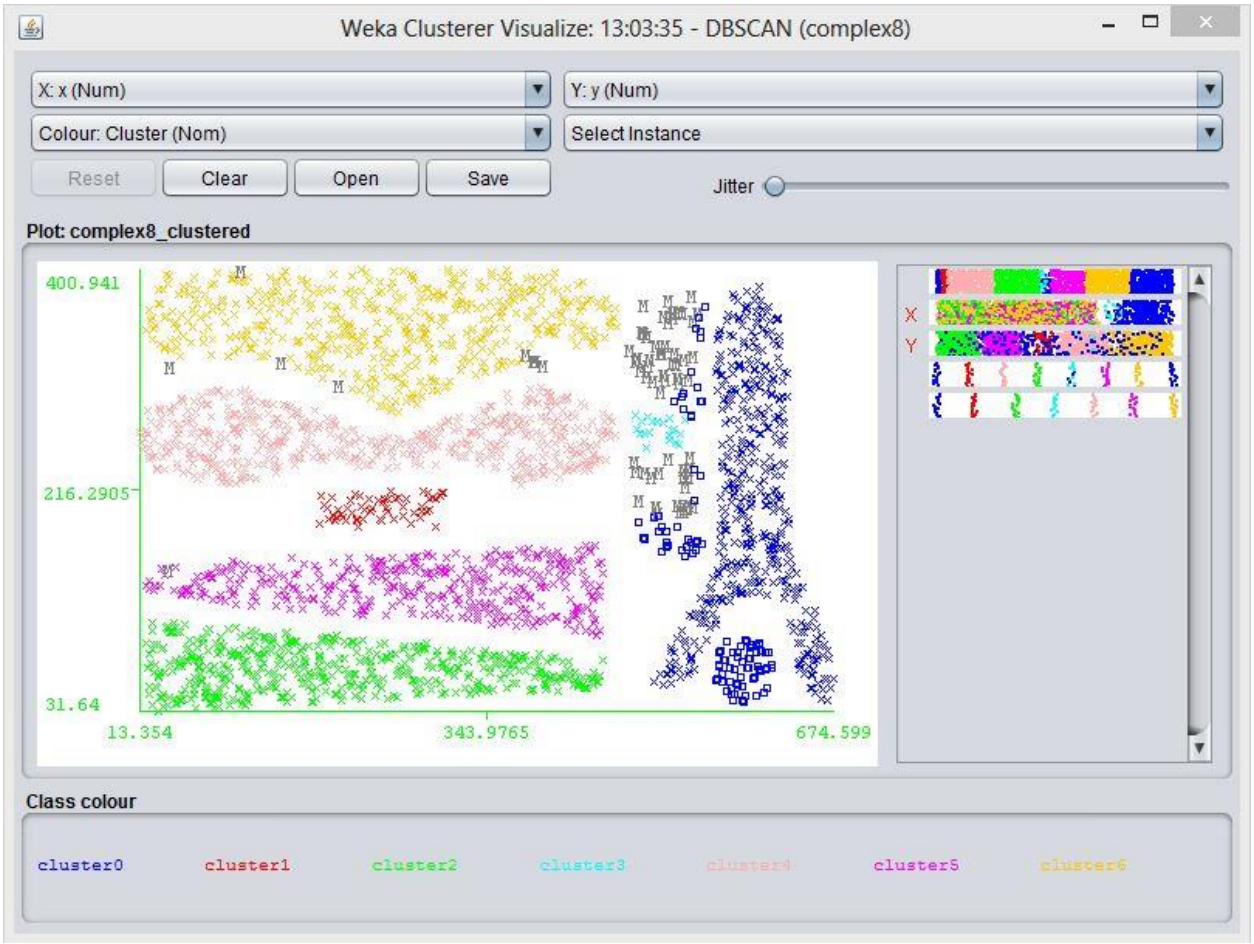
4. DBSCAN με προεπιλεγμένες παραμέτρους και αρχές λειτουργίας του.

   Ο DBSCAN είναι ένας αλγόριθμος συσταδοποίησης που βασίζεται στην πυκνότητα των σημείων.Δημιουργεί clusters σύμφωνα με τις 2 παραμέτρους e και minPoints. Η πρώτη καθορίζει τη μέγιστη απόσταση που μπορεί να έχουν δύο στοιχεία για να ανήκουν στο ίδιο cluster και η δεύτερη πόσους τουλάχιστον γείτονες χρειάζεται για να δημιουργηθεί ένα cluster.

   Εφαρμόζοντας τον με τις προεπιλεγμένες παραμέτρους δίνει 1 cluster με 2032 (79.655 %) incorrectly clustered instances.

5. DBSCAN optimal

   Δοκιμάζοντας τις παραμέτρους του αλγορίθμου επέλεξα σαν βέλτιστες τις e=0.04 και minpoints=13 με απόδοση 103(4.0376%) incorrectly clustered instances .



ΑΣΚΗΣΗ 2 – Apriori Associator

1. Python=1 Revolution Analytics (now part of Microsoft)=1 55 ==> Rlang=1 55 <conf:(1)> lift:(2.13) lev:(0.01) [29] conv:(29.22)

2. Python=1 Rattle=1 44 ==> Rlang=1 44 <conf:(1)> lift:(2.13) lev:(0.01) [23] conv:(23.38)

3. MLlib=1 Scala=1 38 ==> Spark=1 38 <conf:(1)> lift:(8.87) lev:(0.01) [33] conv:(33.72)

4. Hadoop=1 Revolution Analytics (now part of Microsoft)=1 37 ==> Rlang=1 37 <conf:(1)> lift:(2.13) lev:(0.01) [19] conv:(19.66)

5. Hive=1 Revolution Analytics (now part of Microsoft)=1 31 ==> Rlang=1 31 <conf:(1)> lift:(2.13) lev:(0.01) [16] conv:(16.47)

6. Python=1 Revolution Analytics (now part of Microsoft)=1 SQLang=1 31 ==> Rlang=1 31 <conf:(1)> lift:(2.13) lev:(0.01) [16] conv:(16.47)

7. Hive=1 Spark=1 SQL on Hadup tools=1 SQLang=1 31 ==> Hadoop=1 31 <conf:(1)> lift:(5.44) lev:(0.01) [25] conv:(25.3)

8. Pig=1 Rlang=1 Tableau=1 30 ==> Hadoop=1 30 <conf:(1)> lift:(5.44) lev:(0.01) [24] conv:(24.49)

9. Hadoop=1 Revolution Analytics (now part of Microsoft)=1 SQLang=1 30 ==> Rlang=1 30 <conf:(1)> lift:(2.13) lev:(0.01) [15] conv:(15.94)

10. MLlib=1 Rlang=1 Scala=1 30 ==> Spark=1 30 <conf:(1)> lift:(8.87) lev:(0.01) [26] conv:(26.62)