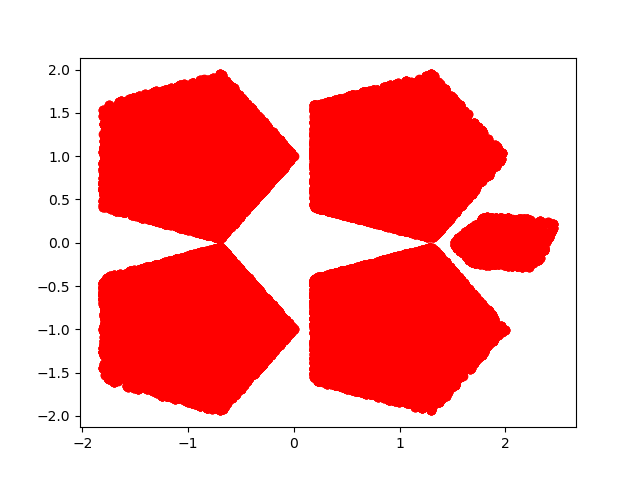
Εύρεση Συστάδων

Για να βρεθεί ο αριθμός των συστάδων που υπάρχουν στα δεδομένα, χρησιμοποιήθηκαν δύο τρόποι. Ο πρώτος αφορά την απεικόνιση ενός μεγάλου δείγματος των δεδομένων με χρήση python, όπου τα αποτελέσματα φαίνονται στο παρακάτω σχήμα. Όπως είναι αντιληπτό, τα δεδομένα εμφανισιακά μπορούν να χωριστούν σε 5 clusters. Στην δεύτερη τεχνική, χρησιμοποιήθηκε δείγμα από τα αρχικά δεδομένα και πάνω σε αυτά έτρεξε επαναληπτικά ένας αλγόριθμος kmeans. Σε κάθε επανάληψη ο αριθμός των clusters αυξάνονταν κατα ένα σε ένα εύρος από 2 έως 10 clusters και γινόταν μέτρηση της μετρικής sum of squared distances(SSE). Η τιμή του SSE φαίνεται να σταματά να μειώνετε σημαντικά στα 4 clusters. Παρόλα αυτά, στην υλοποίηση των αλγορίθμων χρησιμοποιήθηκαν 5 clusters.

Cure

Για τον αλγόριθμο cure αρχικά παίρνουμε ένα μικρό δείγμα από το σύνολο των δεδομένων. Με αυτα τα δεδομένα εκτελούμε έναν ιεραρχικό αλγόριθμο τοπικά με τον οποίο βρίσκουμε 5 clusters όσα υπάρχουν δηλαδή και στα δεδομένα μας. Το επόμενο βήμα είναι από το δείγμα να βγάλουμε τους αντιπροσώπους για κάθε cluster. Για ένα cluster ο πρώτος αντιπρόσωπος επιλέγεται στην τύχη έπειτα βρίσκουμε το σημείο που έχει την μεγαλύτερη απόσταση απο τον πρώτο αντιπρόσωπο και αυτό γίνεται ο δεύτερος αντιπρόσωπος. Αυτή η διαδικασία συνεχίζεται μέχρι να φτάσουμε τον επιθυμητό αριθμό αντιπροσώπων. Όταν έχουμε πάνω από έναν αντιπρόσωπο η απόσταση ενός σημείου από τους αντιπροσώπους ορίζεται ως η ελάχιστη απόσταση του σημείου και των αντιπροσώπων. Έπειτα θέλουμε να μετακινήσουμε τους αντιπροσώπους πιο κοντά στο κέντρο τους, οπότε για κάθε αντιπρόσωπο θέλουμε να βρούμε ένα σημείο πάνω στην ευθεία που ενώνει τον ίδιο και το κέντρο του και μειώνει την απόστασή τους κατά 20%. Έστω ότι τα δυο σημεία, του κέντρο(xk,yk) και του αντιπροσώπου(xa,xb), βρίσκονται στην ευθεία y = ax+b και έχουν απόσταση dist1. Εμείς θέλουμε να βρούμε τα σημεία για τα οποία ισχύει dist2^2 = (xk-x')^2 + (yk-y')^2 από την εξίσωση της ευθεία έχουμε yk-y'=a(xk-x') οπότε έχουμε από την εξίσωση ευθείας dist^2 = (a^2+1)(xk-x')^2 οπότε βγάζουμε το συμπέρασμα ότι τα δύο δυνατά σημεία έχουν x' = xk – (dist^2/(a^2+1))^1/2 ή x' = xk + (dist^2/(a^2+1))^1/2 από τα δύο αυτά σημεία διαλέγουμε αυτό που είναι ανάμεσα στο xa και xk. Βρίσκοντας για όλους τους αντιπροσώπους τα αντίστοιχα x',y' μπορούμε να ορίσουμε για όλα τα σημεία του dataset ένα cluster με ένα πέρασμα βρίσκοντας τον πιο κοντινό αντιπρόσωπο για κάθε σημείο και ορίζοντας το αντίστοιχο cluster.

Για την αξιολόγηση των αποτελεσμάτων χρησιμοποιήθηκε η μετρική SSE

Υβριδικός Αλγόριθμος

Ο υβριδικός αλγόριθμος που υλοποιήθηκε περιλαμβάνει έναν kmeans αλγόριθμο με είσοδο αρκετά μεγάλο αριθμό από clusters και στη συνέχεια με χρήση των κέντρων από αυτά τα clusters έτρεξε ένας ιεραρχικός agglomerative αλγόριθμος για να συγχωνεύσει τα clusters στον επιθυμητό τους αριθμό.

Ο kmeans που χρησιμοποιήθηκε ήταν από την βιβλιοθήκη mllib του spark σε γλώσσα scala. Ως είσοδος δόθηκαν οι συντεταγμένες των παραδειγμάτων όλων των δεδομένων, με μεγάλο αριθμό από clusters (5 φορές τον αριθμό των clusters που ψάχνουμε) και μεγάλο αριθμό από επαναλήψεις (1000), ενώ για τα αρχικά centroids που επιλέγει ο αλγόριθμος χρησιμοποιήθηκε η τυχαία επιλογή μεταξύ των παραδειγμάτων. Ο τύπος των δεδομένων εισόδου ήταν τύπου RDD[linalg.Vector] ενώ ως έξοδος υπολογίστηκαν τα κέντρα των clusters και η λίστα με την κατηγοριοποίηση κάθε στοιχείου (prediction).

Ο ιεραρχικός αλγόριθμος είναι ένας επαναληπτικός αλγόριθμος υλοποιημένος και αυτός σε scala. Σε κάθε επανάληψη υπολογίζει την Ευκλείδεια απόσταση όλων των κέντρων μεταξύ τους και επιλέγει να συγχωνεύσει τα δύο clusters με τα πιο κοντινά κέντρα. Στη συνέχεια, κρατάει ως επιθυμητό cluster αυτό με τον μικρότερο αριθμό (id) και ενημερώνει την λίστα με την κατηγοριοποίηση κάθε στοιχείου. Επιπλέον, ενημερώνει το κέντρο του καινούργιο πλέον cluster που προκύπτει από την συγχώνευση, το οποίο παίρνει τιμή ίση με την μέση απόσταση μεταξύ των δύο προηγούμενων κέντρων. Ο αλγόριθμος ολοκληρώνεται μόλις παραμείνουν τα επιθυμητά clusters που έχει ορίσει ο χρήστης.

Τέλος υπολογίζουμε την μετρική SSE για την αξιολόγηση των αποτελεσμάτων. Από την αξιολόγηση και των δύο αλγορίθμων ο υβριδικός φαίνεται να δημιουργεί καλύτερες συστάδες.

Το jar αρχείο περιέχει 3 κλάσεις:Cure,Hybrid,FindClusters τα ορίσματα που χρειάζονται είναι για τον Cure ο αριθμός των αντιπροσώπων, ο αριθμός των cluster, το ποσοστό του δείγματος, το path σε φάκελο με τα δεδομένα,το path στο python script .

Για τον Hybrid το path σε φάκελο με τα δεδομένα, ο αριθμός των cluster.

Για την FindCluster το path σε φάκελο με τα δεδομένα

Ακόμα στο python script πρέπει να αλλάξει η πρώτη γραμμή και να δείχνει σε ένα τοπικό περιβάλλον python που έχει εγκατεστημένη την βιβλιοθήκη sklearn.

Στον πηγαίο κώδικα οι μεταβλητές αυτές είναι hardcoded