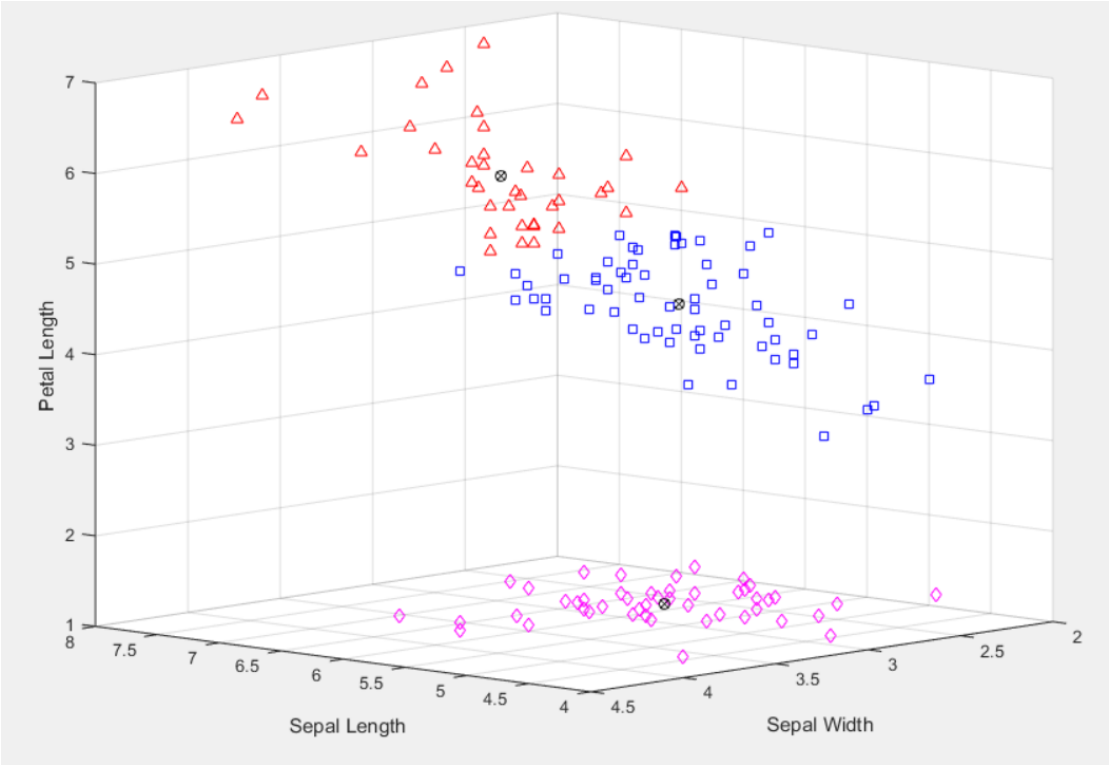
**Αναγνώριση προτύπων**
**Εργασία 8**
**Σάββας Λιάπης 57403**

## Άσκηση 8.1

Χρησιμοποιώντας τον αλγόριθμο K-means προσπαθήστε να ταξινομήσετε τα φυτά σε 3 ομάδες. Ποιο είναι το λάθος που βρίσκετε;

### Απάντηση 8.1:

Ο αλγόριθμος στην ουσία ορίζει στην τύχη τα αρχικά κέντρα και χρησιμοποιώντας την ευκλείδεια απόσταση υπολογίζει σε ποιο κέντρο είναι πιο κοντά στο κάθε κέντρο. Αφού δημιουργηθούν οι πρώτες κλάσεις τα σημεία της κάθε κλάσης επιλέγουν το καινούργιο κέντρο. Το μειονέκτημα αυτού του αλγορίθμου είναι ότι παρόλο που η αρχικοποίηση των κέντρων είναι τυχαία, επηρεάζει σημαντικά το αποτέλεσμα της αρχικοποίησης. Με τον αλγόριθμο που τρέξαμε πήραμε τα εξής αποτελέσματα :



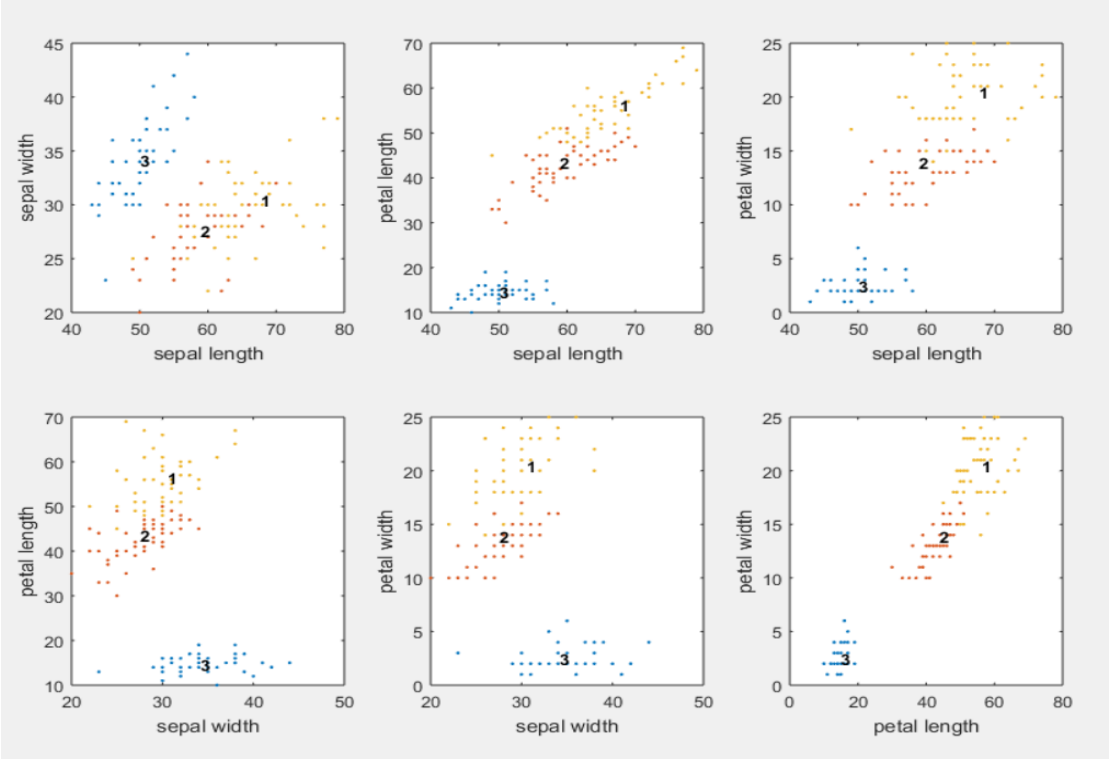

Η ταξινόμηση αυτή δίνει σφάλμα 10.67 %

## Άσκηση 8.2

Χρησιμοποιώντας τον αλγόριθμο fuzzy C-means προσπαθήστε να ταξινομήσετε τα φυτά σε 3 ομάδες. Ποιό είναι το λάθος που βρίσκετε;

### Απάντηση 8.2:

Ο αλγόριθμος fuzzy-C means λειτουργεί παρόμοια με τον K-means, με διαφορά ότι ο fuzzy c means δίνει για το κάθε σημείο, την πιθανότητες με τις οποίες ανήκει στην κάθε κλάση και χρησιμοποιεί μια membership function στην οποία συμμετέχουν όλα τα σημεία του dataset. Αυτό επιτρέπει σε πολύ κοντινά σημεία να «αυτοοργανωθουν» λαμβάνοντας υπόψιν και τα γύρω τους σημεία. Ο δικός μας αλγόριθμος έδωσε τα εξής αποτελέσματα .




```
Iteration count = 1, obj. fcn = 29529.990302
Iteration count = 2, obj. fcn = 22595.346981
Iteration count = 3, obj. fcn = 21269.893332
Iteration count = 4, obj. fcn = 13822.145632
Iteration count = 5, obj. fcn = 7563.345314
Iteration count = 6, obj. fcn = 6743.870874
Iteration count = 7, obj. fcn = 6203.492832
Iteration count = 8, obj. fcn = 6076.784404
Iteration count = 9, obj. fcn = 6060.947054
Iteration count = 10, obj. fcn = 6059.104804
Iteration count = 11, obj. fcn = 6058.813283
Iteration count = 12, obj. fcn = 6058.736682
Iteration count = 13, obj. fcn = 6058.708964
Iteration count = 14, obj. fcn = 6058.697826
Iteration count = 15, obj. fcn = 6058.693235
Iteration count = 16, obj. fcn = 6058.691332
Iteration count = 17, obj. fcn = 6058.690542
Iteration count = 18, obj. fcn = 6058.690214
Iteration count = 19, obj. fcn = 6058.690078
Iteration count = 20, obj. fcn = 6058.690022
Iteration count = 21, obj. fcn = 6058.689999
Iteration count = 22, obj. fcn = 6058.689989
Iteration count = 23, obj. fcn = 6058.689985
Iteration count = 24, obj. fcn = 6058.689983
Iteration count = 25, obj. fcn = 6058.689983

error =

    0.1067
```

Αριστερά παρατηρούμε πως εξελίσσεται η τιμή της objective function, καθώς αυξάνεται ο αριθμός των επαναλήψεων αυξάνεται και η τιμή του objective function.

Σαν σφάλμα παίρνουμε πάλι 10.67% . Αυτό το σφάλμα όπως και στην πρώτη άσκηση οφείλεται στο ότι τα δείγματα 2 ειδών ιρις είναι πολύ κοντά μεταξύ τους στα «σύνορά» τους και ο ταξινομιτής δεν ξέρει που να τα βάλει.

## Άσκηση 8.3

Χρησιμοποιώντας τον αλγόριθμο ISODATA προσπαθήστε να ταξινομήσετε τα φυτά σε ομάδες. Πόσες ομάδες βρίσκετε ; Ποιο είναι το λάθος που βρίσκετε ;

### Απάντηση 8.3:

Αυτός ο αλγόριθμος είναι πολύ παρόμοιος με τον k-means καταπολεμώντας ένα ακόμα μειονέκτημά του. Δεν απαιτεί να γνωρίζουμε τον αριθμό των clusters που θα χωρίσουμε το dataset. Με άλλα λόγια ορίζουμε έναν μέγιστο αριθμό κλάσεων (αρκεί να είναι μεγαλύτερος ή ίσος από τις κλάσεις που πιθανόν να υπάρχουν ) και κάποια κριτήρια σύμφωνα με τα οποία 2 κλάσεις μπορούν να συγχωνευθούν . Έτσι καθώς τρέχει ο αλγόριθμος καταλήγει (αν έχουμε ορίσει σωστά όλα τα κριτήρια ) σε ένα ικανοποιητικό αποτέλεσμα. Το Πρόβλημα είναι ότι αν δεν έχουμε μία γενική εικόνα για το πρόβλημα που πάμε να ταξινομήσουμε, τότε δεν θα ξέρουμε αν έχουμε ορίσει σωστά τα κριτήρια μας. Στην δική μας περίπτωση παίρνουμε τα εξής αποτελέσματα :

```
Number of Clusters: 2

error =

    0.3333
```

Αν ορίσουμε κ=3 (δηλαδή μέγιστος αριθμός κλάσεων) τότε ο αλγόριθμος επιστρέφει 2 clusters αντί για 3 και έχουμε και ένα απαγορευτικό σφάλμα όπως είναι προφανές

Αν ορίσουμε k=4 και παραπάνω το αποτέλεσμα που παίρνουμε είναι σωστός αριθμός κλάσεων με ικανοποιητικό σφάλμα 10.67 % . Ωστόσο δεν αρκεί μόνο αυτό .

```
Number of Clusters: 3

error =

    0.1067
```

```
Number of Clusters: 3

error =

    0.2067
```

Αν για παράδειγμα θεωρήσουμε ότι το minimum distance είναι 1 αντί για 10 (που είχαμε ορίσει αρχικά τότε το σφάλμα είναι μεγαλύτερο)

Οπότε είναι ένα καλός αλγόριθμος αν έχουμε γενική γνώση του θέματος που πάμε να ταξινομήσουμε για να ορίσουμε σωστά τα κριτήρια.