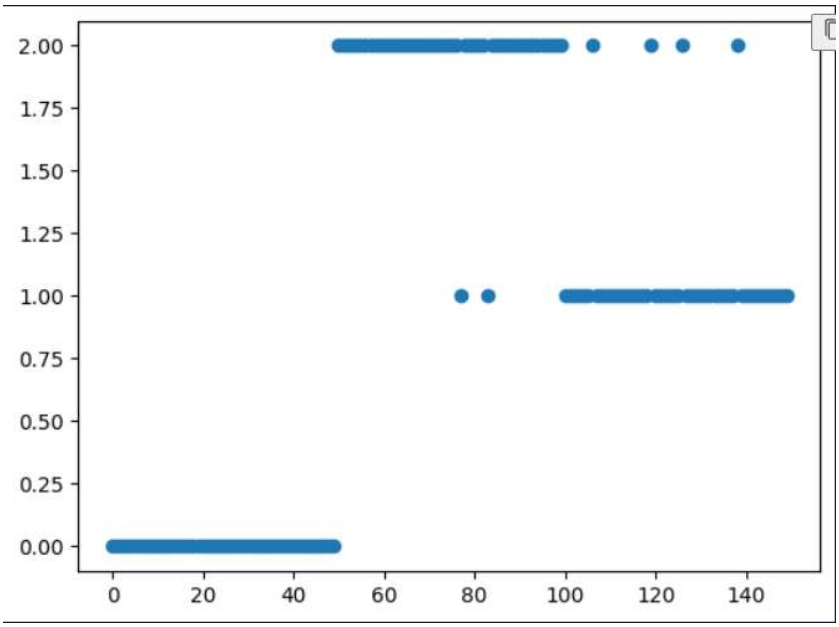
Εξόρυξη Δεδομένων
Χειμερινό εξάμηνο
Εργασία 2023-2024
Αριθμός Μητρώου: ice18390094
Ονοματεπώνυμο: Άγγελος Τζώρτζης

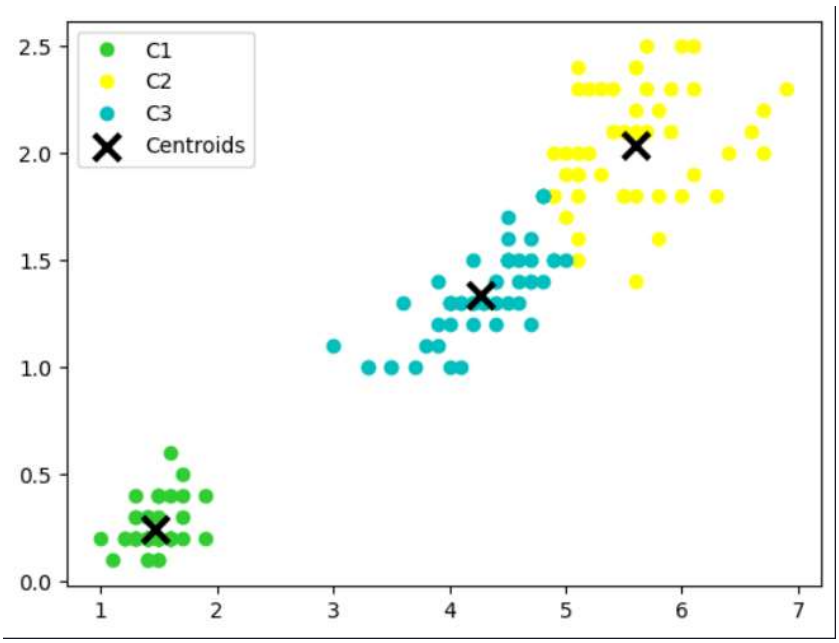
Μέρος 1ο: Διαμεριστική συσταδοποίηση με k-means

1.1 Εφαρμογή στο σύνολο δεδομένων iris: part_1_1.py

Βήμα 1, Βήμα 2: Εκτελούμε τις εντολές python από την εκφώνηση για την εφαρμογή του αλγορίθμου k-means στα δεδομένα του συνόλου iris.

Βήμα 3: Εκτελούμε τις εντολές python από την εκφώνηση για την παρουσίαση των δεδομένων σε γράφημα:





Παραπάνω φαίνονται οι 3 συστάδες που προέκυψαν μαζί με το κεντρο τους.

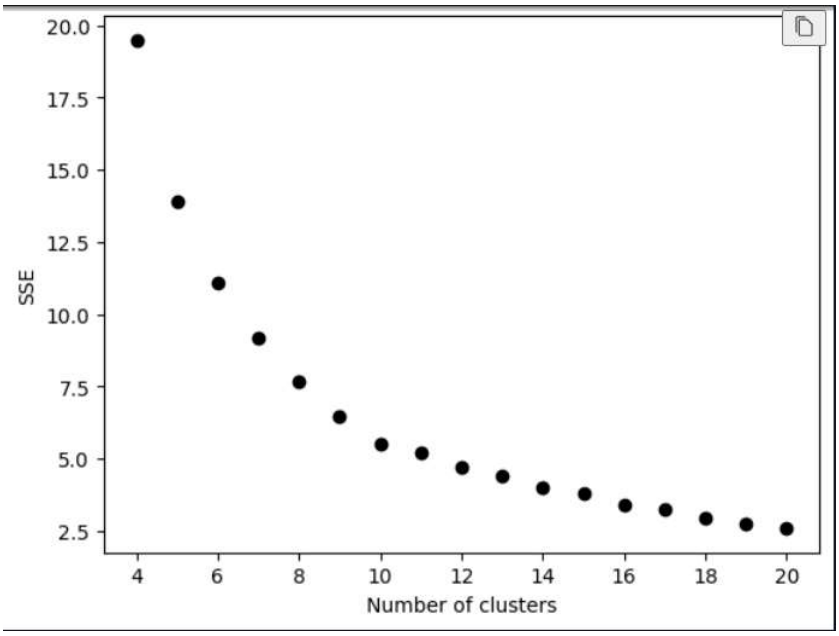
Βήμα 4:
Υπολογίζουμε τον συντελεστή περιγράμματος καθώς και το SSE για αριθμό συστάδων από το 4 εώς το 20.

Οι τιμές του συντελεστή περιγράμματος:

```
Silhouette coefficient for 4 clusters: 0.6127580794464402
Silhouette coefficient for 5 clusters: 0.5883732712110276
Silhouette coefficient for 6 clusters: 0.5753448148375986
Silhouette coefficient for 7 clusters: 0.5733268544686511
Silhouette coefficient for 8 clusters: 0.58962814137616
Silhouette coefficient for 9 clusters: 0.5866932137871326
Silhouette coefficient for 10 clusters: 0.43421900489031967
Silhouette coefficient for 11 clusters: 0.4210785623234128
Silhouette coefficient for 12 clusters: 0.41449212533731716
Silhouette coefficient for 13 clusters: 0.3854868307501737
Silhouette coefficient for 14 clusters: 0.38854892643642497
Silhouette coefficient for 15 clusters: 0.38686408718863535
Silhouette coefficient for 16 clusters: 0.3920704672328498
Silhouette coefficient for 17 clusters: 0.3683374504736415
Silhouette coefficient for 18 clusters: 0.3889680616066231
Silhouette coefficient for 19 clusters: 0.39753308069146487
Silhouette coefficient for 20 clusters: 0.4054168692175378
```

Παρατηρούμε ότι ο συντελεστής περιγράμματος είναι θετικός σε όλες τις περιπτώσεις άρα η συσταδοποίηση πιθανόν έχει γίνει σωστά με κάθε σημείο να ανήκει στην σωστή συστάδα.

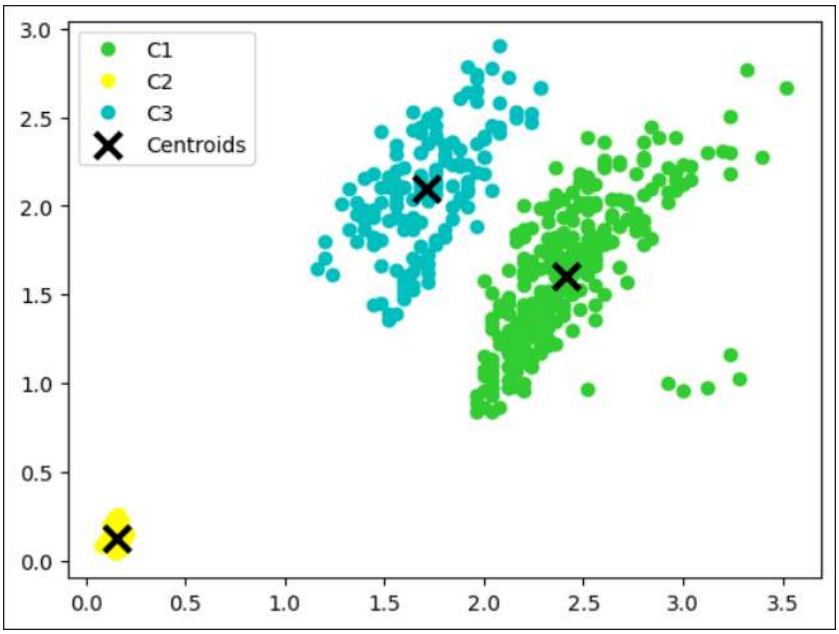
Σχέση τιμής SSE με τον αριθμό των συστάδων:



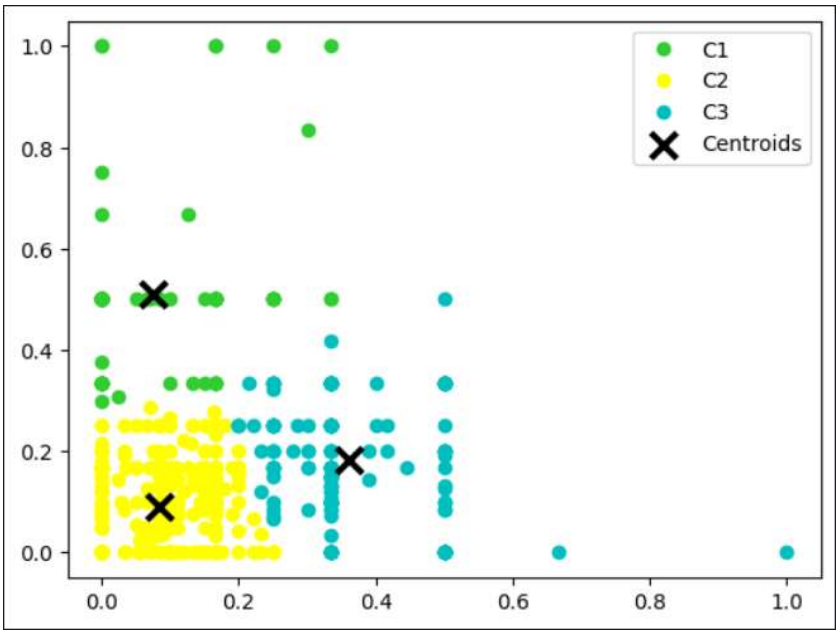
Παρατηρείται μείωση της τιμής του SSE με την αύξηση του αριθμού των συστάδων. Προκύπτει αυτό το αποτέλεσμα καθώς όσο αυξάνεται ο αριθμός των συστάδων,

μειώνεται η απόσταση του κάθε σημείου από την συστάδα του που οδηγεί σε μικρότερη τιμή του SSE.

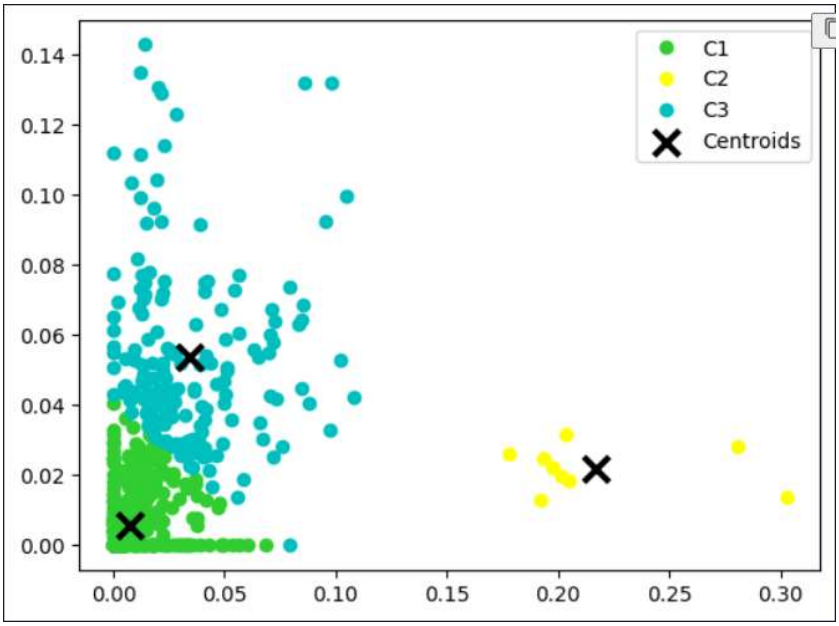
1.2 Εφαρμογή στο σύνολο δεδομένων xV.mat: part_1_2.py
Βήμα 1, Βήμα 2: Εκτελούμε τις εντολές python από την εκφώνηση για την εφαρμογή του αλγορίθμου k-means στα δεδομένα του συνόλου xV.mat, εμφάνιση των δεδομένων σε γράφημα και υπολογισμός της τιμής SSE:
Τιμή SSE:
`Sum of Squared Errors: 99.4493946370509`
Γράφημα:



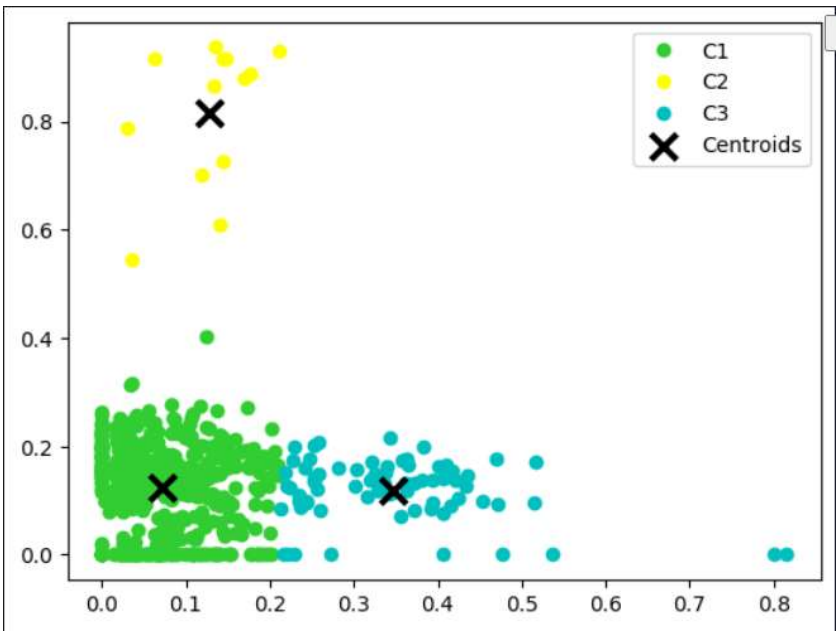
Βήμα 3:
Τιμή SSE
`Sum of Squared Errors: 11.401819120120038`
Γράφημα:



Βήμα 4:
Τιμή SSE
`Sum of Squared Errors: 0.3300200789718988`
Γράφημα:



Βήμα 5:
Τιμή SSE
`Sum of Squared Errors: 6.362905489884494`
Γράφημα:



Βήμα 6:
Συγκρίνοντας τις τιμές των SSE και βλέποντας τα γραφήματα των βημάτων 2, 4, και 5 παρατηρούμε μεγάλη απόκλιση στις τιμές των SSE. Καταλαβαίνουμε πώς προκύπτει αυτό το αποτέλεσμα αν παρατηρήσουμε τις αποστάσεις των σημείων των συστάδων απο τα κέντρα τους π.χ. στο βήμα 4 βγαίνει πολύ μικρή τιμή για το SSE καθώς είναι

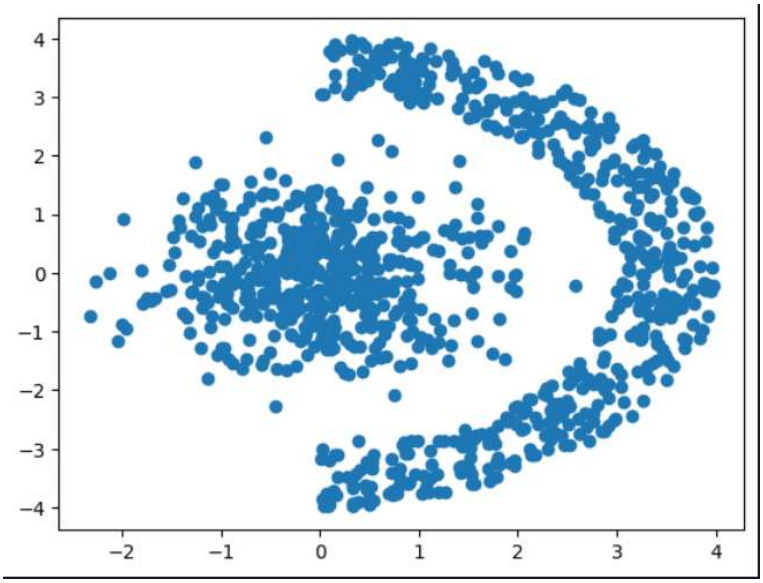
πολύ μικρές οι αποστάσεις από τα κέντρα. Αντιθέτως οι αποστάσεις είναι πολύ μεγάλες στο βήμα 2 για αυτό έχει προκύψει και μεγάλη τιμή SSE. Καταλήγουμε ότι τα δεδομένα που επιλέχθηκαν για το βήμα 4 εκφράζουν καλύτερα το σύνολο το δεδομένων απο τα τα δεδομένα των βημάτων 2, 5 και αντίστοιχα τα δεδομένα του βήματος 2 τα εκφράζουν χειρότερα. Άρα με σειρά ακρίβειας: 4 > 5 > 2.

Μέρος 2ο: Συσταδοποίηση βάση πυκνότητας με DBSCAN

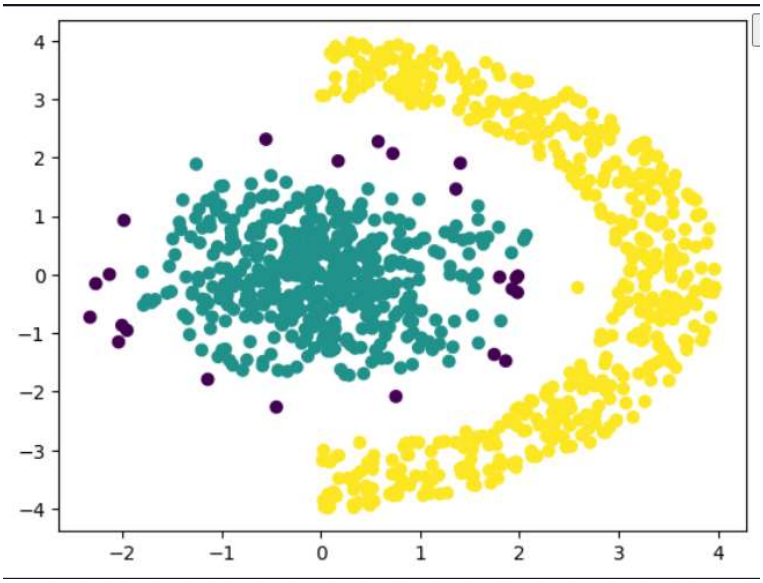
2.1 Εφαρμογή στο σύνολο δεδομένων mydata: part_2_1.py

Βήμα 1, Βήμα 2: Εκτελούμε τις εντολές python από την εκφώνηση για την εφαρμογή του αλγορίθμου DBSCAN στα δεδομένα του συνόλου mydata.

Βήμα 3: Δεδομένα από το αρχείο mydata.mat που θα αναλύσουμε:



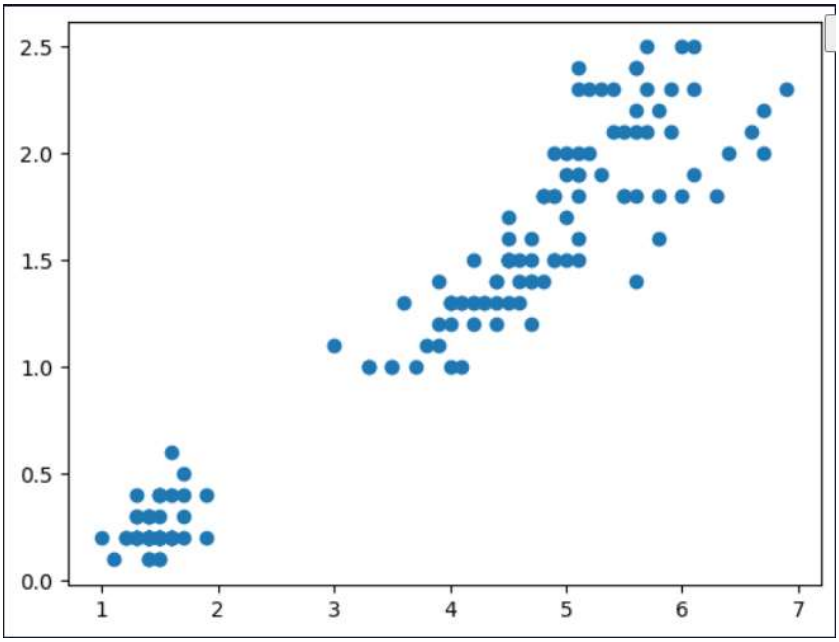
Βήμα 4: Ομαδοποιημένα δεδομένα με χρήση του αλγορίθμου DBSCAN με παραμέτρους ε = 0.5, MinPoints = 15:



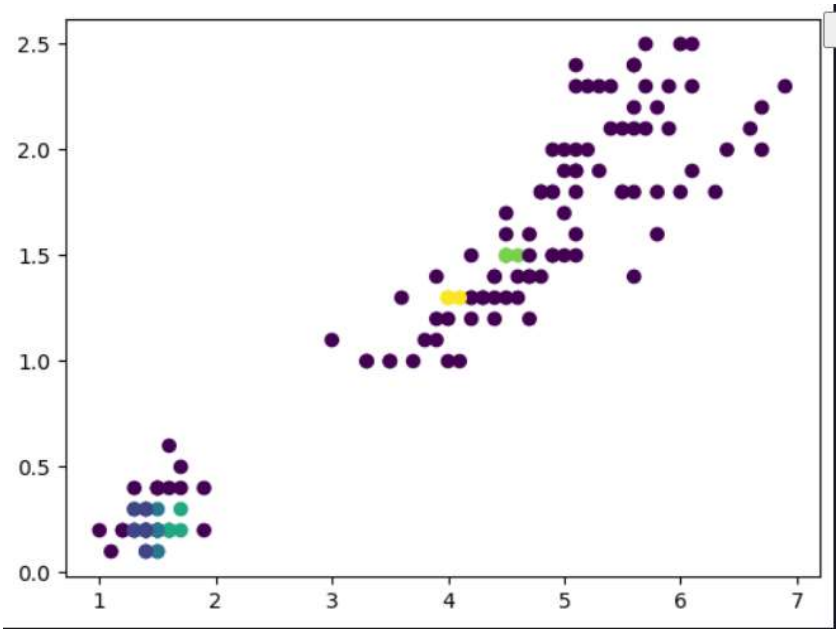
Παρατηρούμε ότι ο DBSCAN ομαδοποιεί τα δεδομένα με βάση την πυκνότητα των σημείων μεταξύ τους. Επίσης παρατηρείται θόρυβος εκεί που η πυκνότητα των σημείων είναι μικρή. Προέκυψε ικανοποιητικό αποτέλεσμα καθώς ορίστηκαν σωστά οι τιμές ε και MinPoints.

2.2 Εφαρμογή στο σύνολο δεδομένων iris: part_2_2.py
Βήμα 1, Βήμα 2: Εκτελούμε τις εντολές python από την εκφώνηση για την εφαρμογή του αλγορίθμου DBSCAN στα δεδομένα του συνόλου iris.

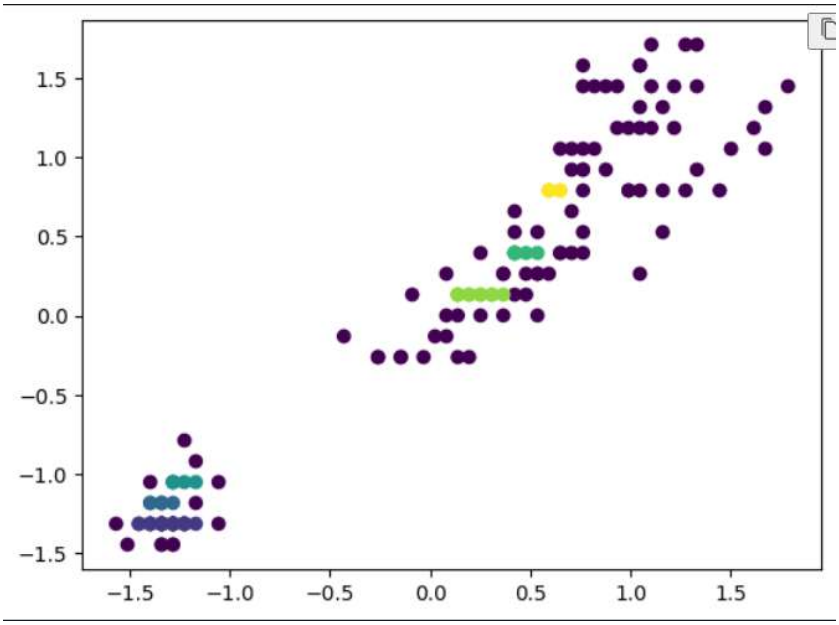
Βήμα 3: Δεδομένα από το σύνολο iris που θα αναλύσουμε:



Βήμα 4: Ομαδοποιημένα δεδομένα με χρήση του αλγορίθμου DBSCAN με παραμέτρους ε = 0.1, MinPoints = 5:



Βήμα 5: Ομαδοποίηση και με κανονικοποιημένα δεδομένα βάση το zscore:



Παρατηρούμε πώς τα δεδομένα δεν ομαδοποιήθηκαν όπως θα θέλαμε και εμφανίζεται μεγάλος αριθμός δεδομένων ώς θόρυβος. Προέκυψε αυτό το αποτέλεσμα καθώς οι τιμές που ορίσαμε για την ακτίνα και τον αριθμό σημείων ήταν μικρές για την πυκνότητα των δεδομένων μας. Επίσης είναι πρόβλημα πως οι ομάδες που θέλουμε να προκύψουν δέν είναι τόσο προφανές όπως ήταν για τα δεδομένα του mydata.mat.

Το τελικό συμπέρασμα που προκύπτει είναι πώς δέν μπορούμε να χρησιμοποιήσουμε οποιονδήποτε αλγόριθμό για οποιοδήποτε σύνολο δεδομένων αλλα θα πρέπει πρώτα να αναλύσουμε την διάταξη των δεδομένων με βάση τις γνώσεις μας και στην πορεία να εφαρμόσουμε τον κατάλληλο αλγόριθμό ώστε να εξάγουμε το σωστό αποτέλεσμα.