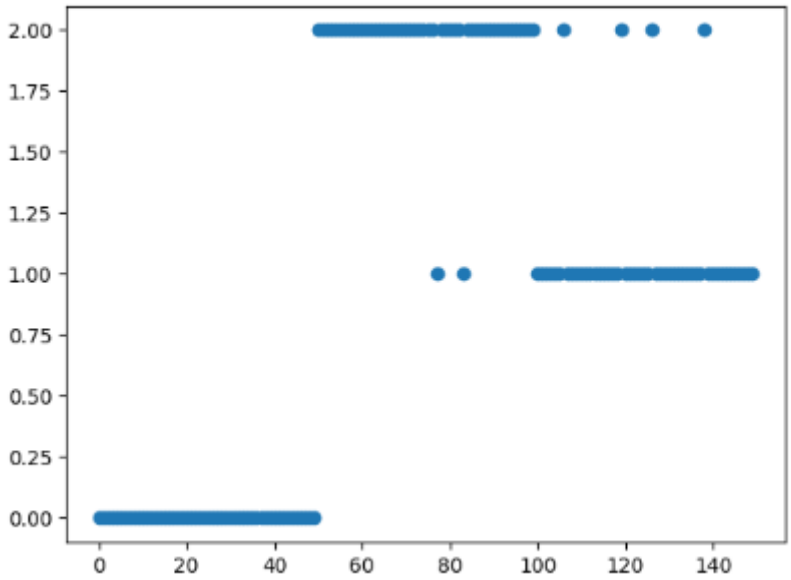
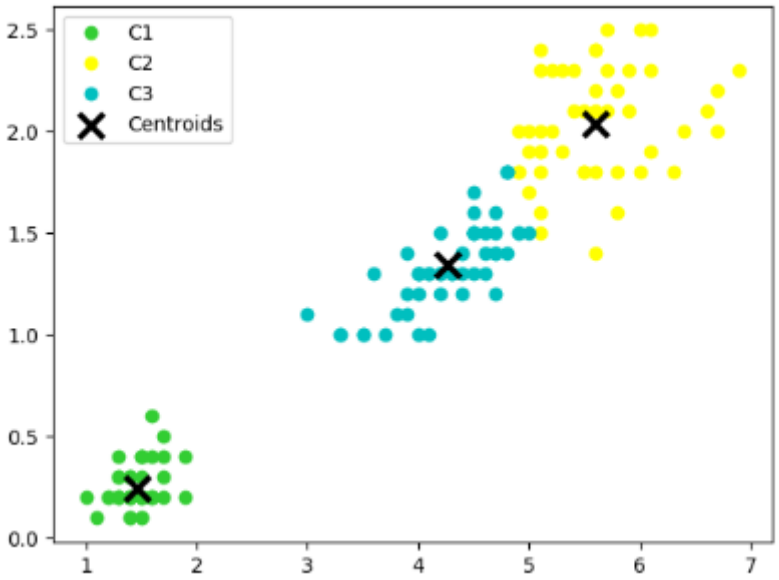
Μέρος 1 : Διαμεριστική συσταδοποίηση με k-means

Εφαρμογή στο Iris dataset

Αρχικά έγινε χρήση των δύο τελευταίων διαστάσεων του πίνακα δεδομένων X. Ορίστηκε ότι τα δεδομένα θα οργανωθούν σε 3 συστάδες και με όλες τις υπόλοιπες παραμέτρους του k-means στις default ρυθμίσεις τα αποτελέσματα είναι τα εξής:
Οργάνωση δεδομένων σε συστάδες 0, 1, 2:



Κεντροειδή για τις δύο τελευταίες διαστάσεις:

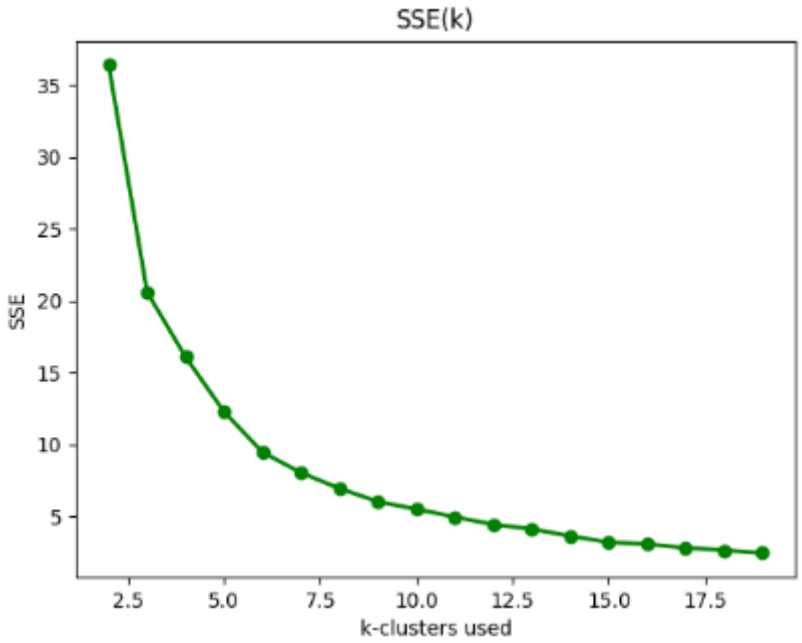


Παρατηρούμε ότι ο αλγόριθμος υπολογίζει κεντροειδή στο κέντρο των χαρακτηριστικών. Τα χαρακτηριστικά είναι εφικτό να απεικονιστούν μιας και είναι μόνο 2. Παρατηρείτε, επίσης, ότι οι ομάδες των χαρακτηριστικών είναι εύκολα διαχωρίσιμες. Ως μετρική για την απόδοση του

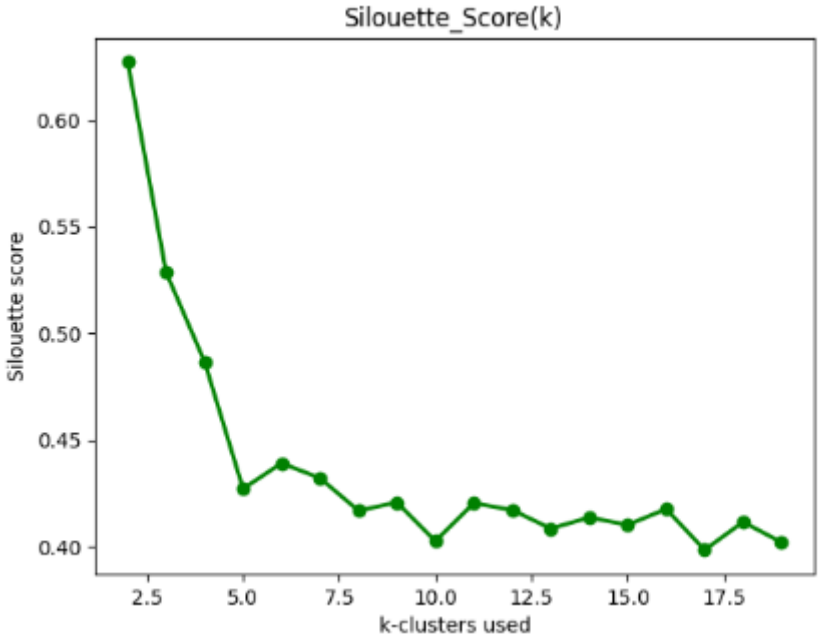
k-means γίνεται χρήση του SSE (Sum of Squared Errors), μέσω της μεταβλητής inertia_, και του Silhouette Coefficient ή Silhouette Score ή συντελεστή περιγράμματος. Η σταδιακή μείωση του SSE, κατά την εκτέλεση του k-means, χρησιμοποιείτε για να παρατηρήσουμε το converge του αλγορίθμου. Για την παραπάνω εκτέλεση έχουμε ότι SSE=31.4 και SilScore=0.66. Τα αποτελέσματα αυτά μας δείχνουν ότι η τεχνική δούλεψε σωστά, μιας και το SSE είναι σχετικά χαμηλό και το SilScore μας δείχνει ότι οι συστάδες απέχουν αρκετά για να διαφοροποιούνται.

Από τους πειραματισμούς με τις παραμέτρους του k-means αλγορίθμου με τις παραπάνω ρυθμίσεις, βρέθηκε ότι το μικρότερο SSE επιτυγχάνεται από τα χαρακτηριστικά 1 και 3 (δείκτες του πίνακα). Η αλλαγή του init από την μέθοδο k-means++, που επιλέγει “έξυπνα” τα αρχικά κέντρα, σε random, που τα επιλέγει τυχαία στοιχεία του συνόλου δεδομένων ως κέντρα, δεν έδειξε κάποια αλλαγή, θεωρητικά λόγω του υψηλού αριθμού επαναλήψεων του αλγορίθμου επιτυγχάνεται ούτως η άλλως σύγκλιση.

Στην συνέχεια, μέσω της εναλλαγής του k παρατηρούμε ότι το SSE μειώνεται όσο αυξάνει το k.

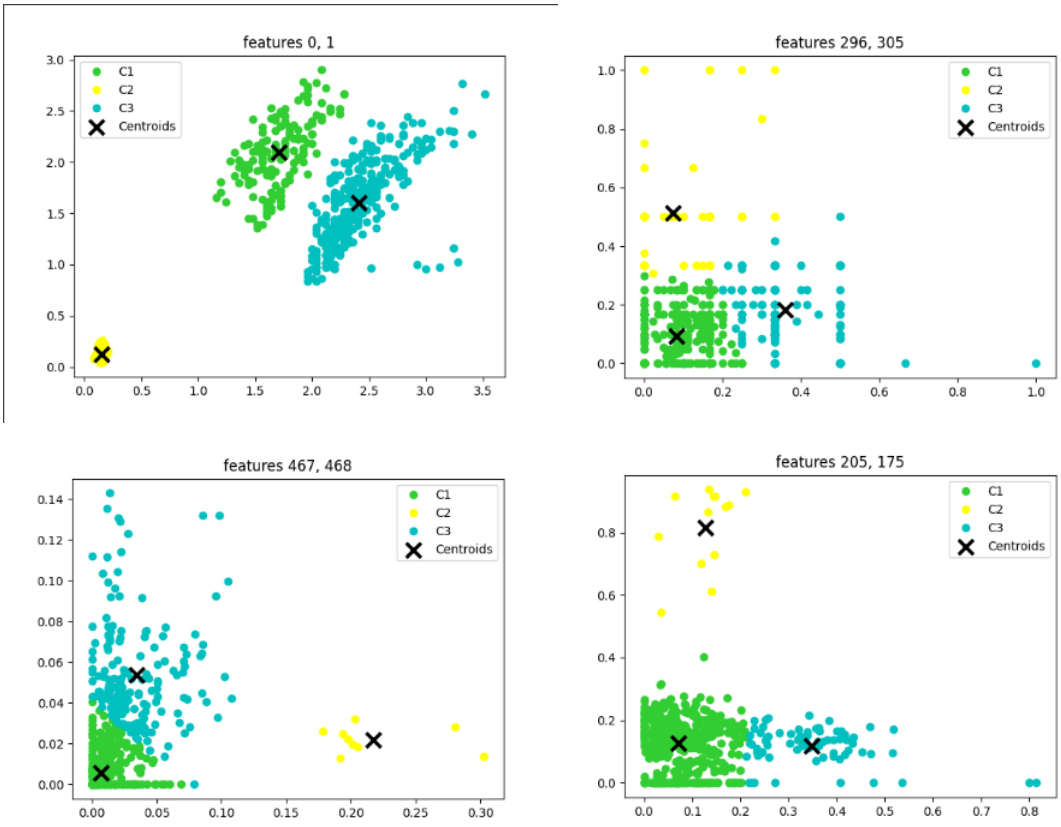


Αυτό συμβαίνει διότι όσο μεγαλώνει ο αριθμός των συστάδων, τόσο μικραίνει και η απόσταση του εκάστοτε δεδομένου από το κοντινότερο κεντροειδές, μιας και εκ των πραγμάτων υπάρχουν περισσότερα κεντροειδή. Στο διάγραμμα μπορούμε να παρατηρήσουμε και το elbow point (το σημείο που η μείωση του SSE δεν είναι πλέον σημαντική) στις 3 ή και 4 συστάδες και να επιβεβαιώσουμε ότι όντως τα δεδομένα μας είναι χωρισμένα σε 3 κλάσεις στην πραγματικότητα. Τέλος, για την περαιτέρω κατανόηση κατασκευάστηκε και το ίδιο διάγραμμα για το SilScore:



Εφαρμογή στο xV.mat

Στην συγκεκριμένη εφαρμογή εκτελέστηκε ο αλγόριθμος k-means για να συσταδοποίηση τα δεδομένα xV.mat, σε 3 συσταδες. Για την εκτέλεση του αλγορίθμου έγινε χρήση ορισμένων από τα 469 features και υπολογίστηκε το SSE. Τα αποτελέσματα παρουσιάζονται παρακάτω, αρχικά ως scatter plot των δεδομένων και των κεντροειδών και ύστερα του SSE.



Features	SSE

[0, 1]	99.5
[296, 305]	11.4
[467, 468]	0.3
[205, 175]	6.4

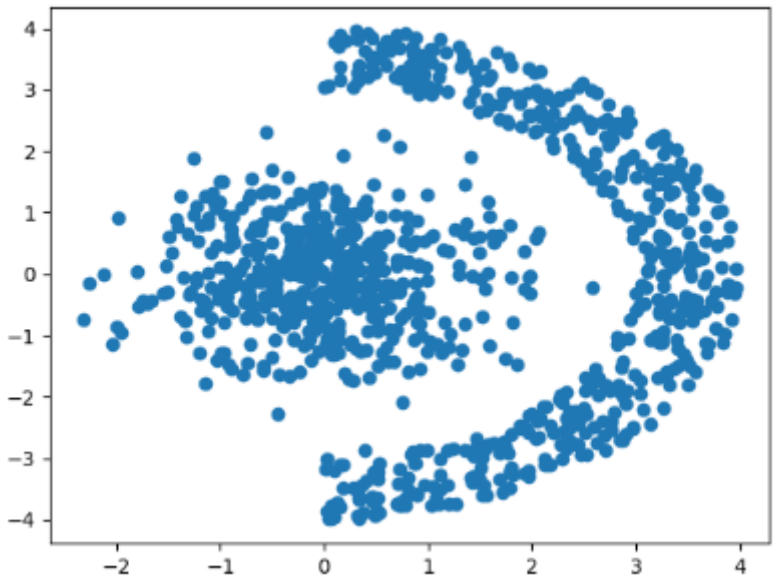
Παρατηρούμε ότι το μέγιστο SSE είναι στα features 0,1, ενώ το ελάχιστο στα δύο τελευταία features. Για να το παρατηρήσουμε αυτό πρέπει να παρατηρήσουμε ότι οι άξονες στα διαγράμματα έχουν διαφορετικα όρια. Στο διάγραμμα των 467,468 features τα όρια είναι 0-0.14 ενώ στα 0,1 τα όρια είναι 0-3.

Μέρος 2 : Συσταδοποίηση βάση πυκνότητας με DBSCAN

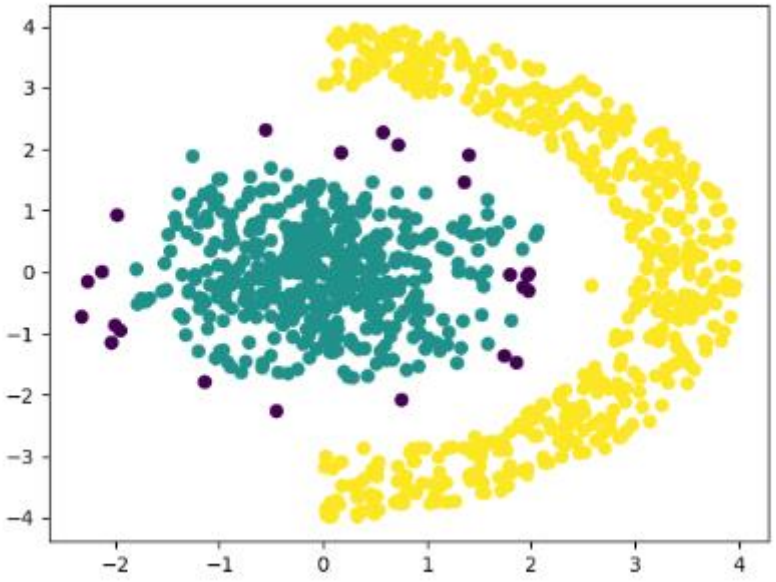
Ο αλγόριθμος συσταδοποίησης DBSCAN προσπαθεί να συσταδοποίηση τα δεδομένα σύμφωνα με τον αριθμό των γειτόνων (πάνω ή κάτω από MinPoints) που υπάρχουν σε ακτίνα epsilon κάθε δεδομένου.

Εφαρμογή στο mydata

Αφού καθορίσαμε τις παραμέτρους της μεθόδου DBSCAN ως εξής: ε=0.5 και minPoints=15 , εκτελέστηκε η μέθοδος DBSCAN. Τα δεδομένα εισόδου είναι τα εξής:



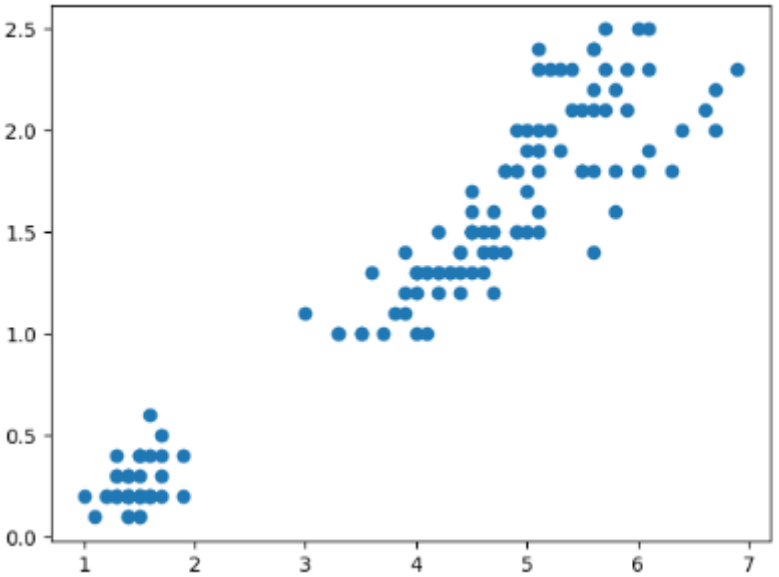
Παρατηρούμε “με το μάτι” ότι οι συστάδες του προβλήματος είναι 2. Τα αποτελέσματα του DBSCAN είναι τα εξής:



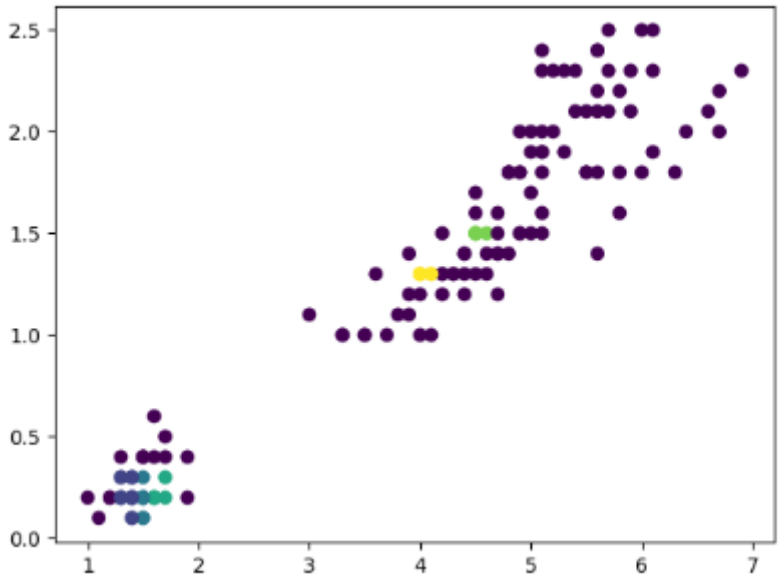
Παρατηρούμε ότι ο DBSCAN, όντως χώρισε τα δεδομένα σε 2 συστάδες αφήνοντας λίγα noise points.

Εφαρμογή στο Iris

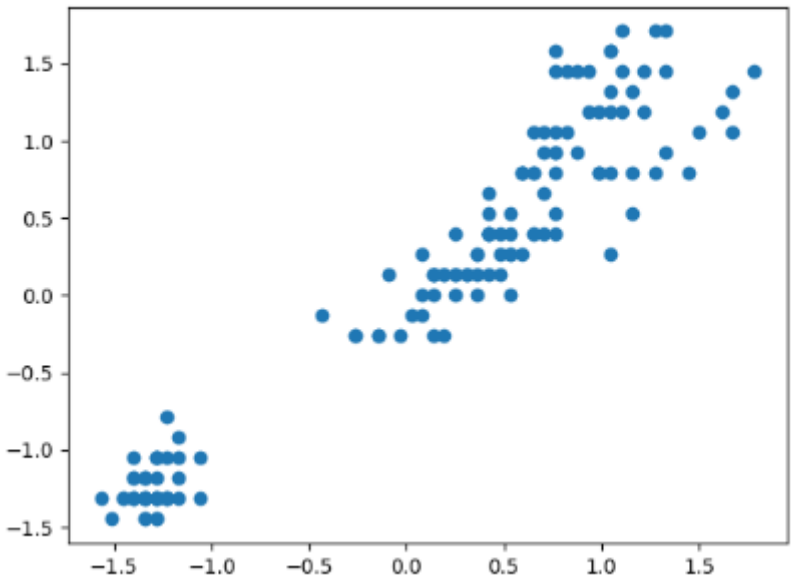
Για την συσταδοποίηση του Iris Dataset όπως και παραπάνω. Στην αρχή κάναμε clustering σε non-normalized (μη κανονικοποιημένα) δεδομένα των δύο τελευταίων features με παραμέτρους ε=0.1 και MinPoints=5. Παρακάτω παρατηρούμε τα δεδομένα:



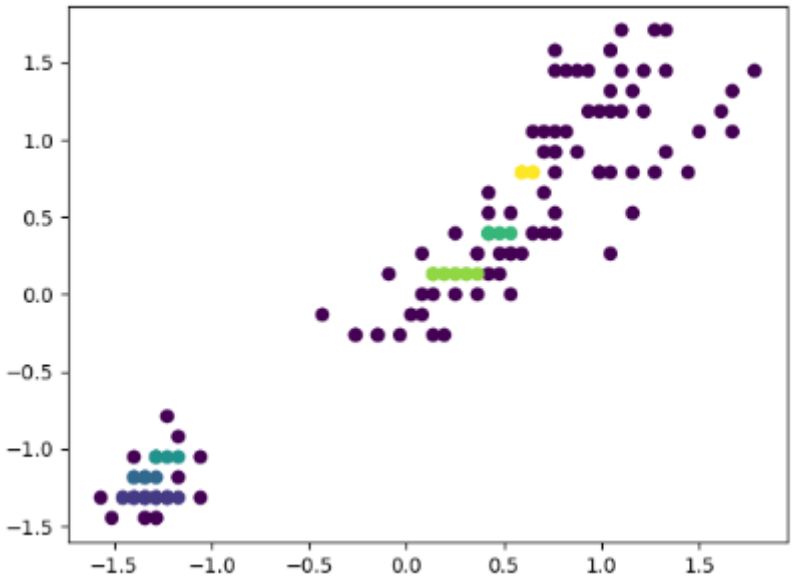
Η διαδικασία του clustering με τον DBSCAN έδωσε τα παρακάτω αποτελέσματα:



Παρατηρούμε ότι πολύ λίγα δεδομένα έχουν γίνει core και border points, ενώ τα περισσότερα είναι noise. Αυτό σημαίνει ότι DBSCAN δεν μπορεί να αποφασίσει με σιγουριά το cluster του κάθε δεδομένου. Επίσης αξίζει να παρατηρήσουμε ότι ο DBSCAN θεώρησε ότι υπάρχουν 4 clusters, ενώ είναι 2 “με το μάτι”. Στην συνέχεια έγινε κανονικοποίηση των δεδομένων μέσω της zscore και να νέα δεδομένα πλέον εισήχθησαν ξανά στον αλγόριθμο. Τα κανονικοποιημένα δεδομένα φαίνονται παρακάτω:



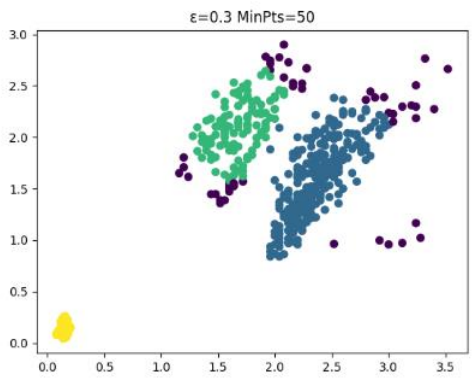
Παρατηρούμε ότι πλέον όλα τα features των δεδομένων βρίσκονται ανάμεσα στις τιμές -1.5 και 1.5, αυτό δείχνει ότι η κανονικοποίηση συνέβη. Ο DBSCAN έδωσε τα παρακάτω αποτελέσματα:

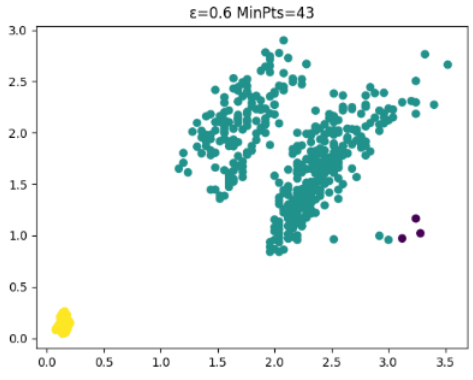


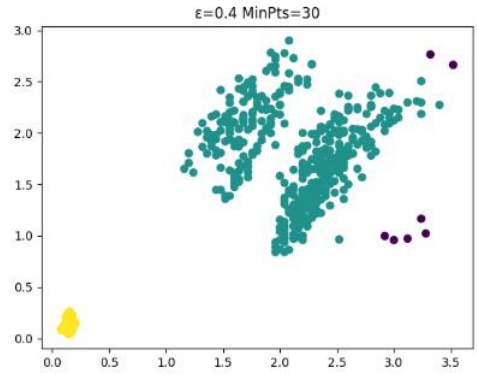
Παρατηρούμε μία αντίστοιχη της προηγούμενης κατάστασης, αλλά σίγουρα η κανονικοποίηση βοήθησε, μιας και πλέον έχουμε περισσότερα core και border points. Παρ' όλα αυτά όμως ο αλγόριθμος θεώρησε ότι υπάρχουν 5 clusters.

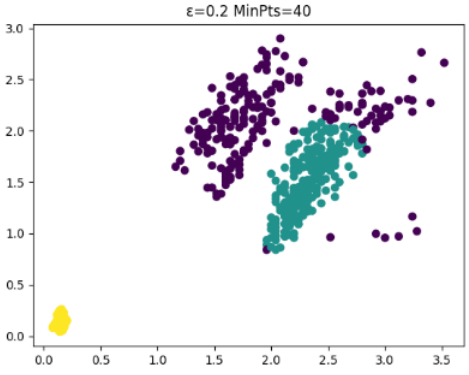
Εφαρμογή στο xV.mat

Για αυτή την εφαρμογή θα εξετάσουμε την σημασία των παραμέτρων ε και minPoints. Από την θεωρία καταλαβαίνουμε ότι η αύξηση της ακτίνας ε, θα οδηγήσει σε μεγαλύτερες συστάδες τον αλγόριθμο, κάτι το οποίο θα κάνει και η αύξηση του ελάχιστου αριθμού γειτόνων minPoints. Αρχικά εξετάζουμε τα δύο πρώτα features μετά την συσταδοποίηση με τις παραμέτρους που φαίνονται στους τίτλους των διαγραμμάτων παρκάτω:



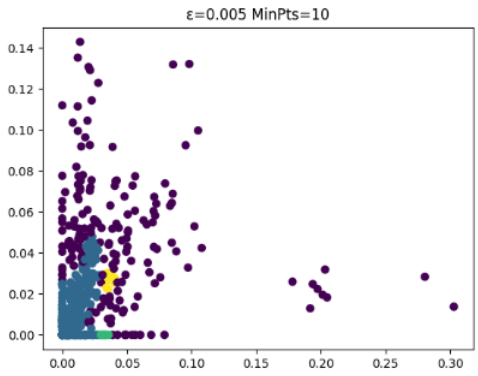


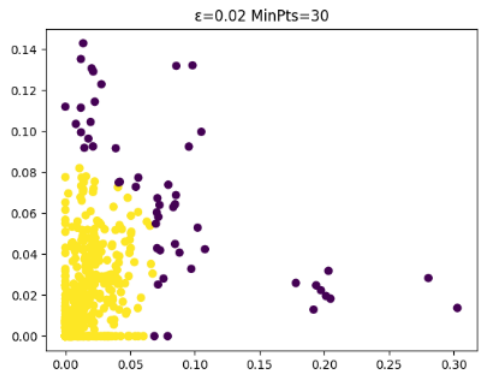


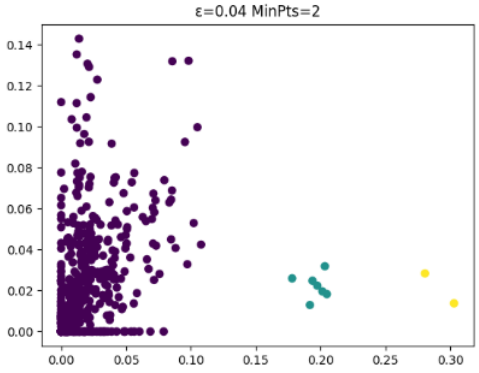


Παρατηρούμε ότι στην πρώτη περίπτωση (κατά σειρά) έχουμε καλό διαχωρισμό των δεδομένων με πολύ λίγα noise points. Αφού αυξήσαμε την ακτίνα ε, όπως ήταν φυσικό οι πάνω συστάδες, που ήταν κοντά η μία στην άλλη, θεωρούνται πλέον μία συστάδα (στα πειράματα 2 και 3). Ενδιαφέρον παρουσιάζει και η τέταρτη περίπτωση, για πολύ μικρή ακτίνα ε παρατηρούμε ότι αρκετά δεδομένα γίνονται Noise points.

Στην συνέχεια, θα εξετάσουμε τα δύο τελευταία features του xV.mat για τις διάφορες τιμές του ε και minPoints. Τα αποτελέσματα φαίνονται παρακάτω:







Μιας και τα δεδομένα είναι πολύ πιο κοντά το ένα στο άλλο για τα δύο τελευταία features, μειώθηκε αρκετά η ακτίνα ε. Στην πρώτη περίπτωση παρατηρούμε ότι οι γειτονιές είναι πολύ μικρές μιας και οι συστάδες που δημιουργούνται δεν είναι καλά διαχωρισμένες. Για τον λόγο αυτό αυξήσαμε τα minPoints και πήραμε τα αποτελέσματα του δεύτερου διαγράμματος. Τέλος, μειώνοντας τα MinPoints παρατηρούμε ότι μπορούμε να αναγνωρίσουμε τις απομακρυσμένες συστάδες, όπως φαίνεται στα αποτελέσματα του 3ου πειράμματος.