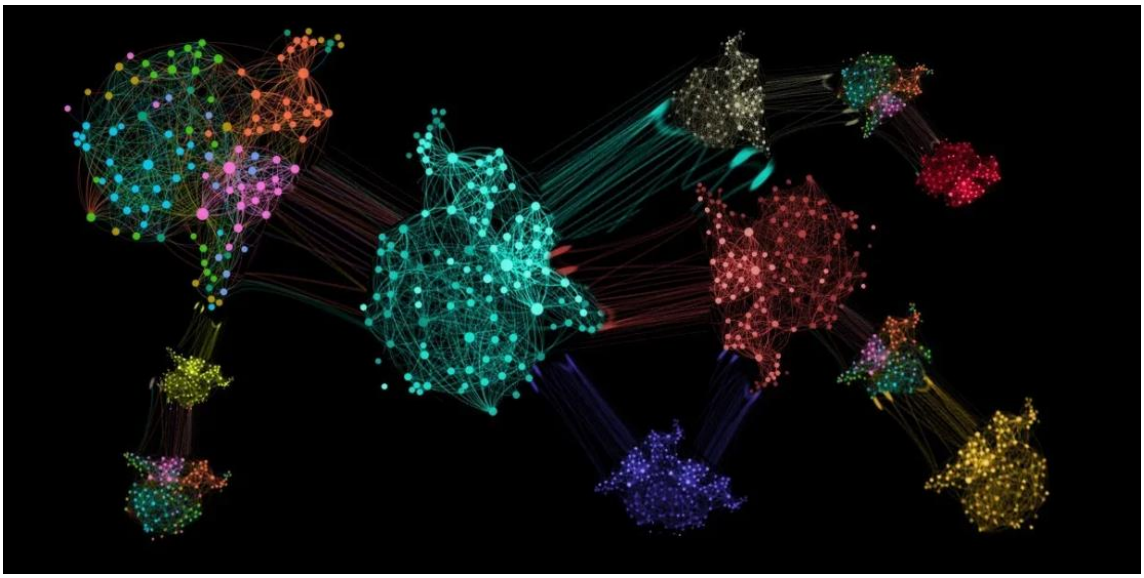
ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«ΣΥΣΤΑΔΟΠΟΙΗΣΗ ΒΑΣΕΙ ΠΥΚΝΟΤΗΤΑΣ: ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΗ ΜΕΛΕΤΗ»



της φοιτήτριας
Μπαλάσκα Σταυρούλα
Αρ. Μητρώου: 154501

Επιβλέπων Καθηγητής
Ουγιάρογλου Στέφανος

Ημερομηνία:

Τίτλος Δ.Ε. Συσταδοποίηση βάσει πυκνότητας: βιβλιογραφική ανασκόπηση και πειραματική μελέτη
Κωδικός Δ.Ε. 24134
Ονοματεπώνυμο φοιτητή: Μπαλάσκα Σταυρούλα
Ονοματεπώνυμο εισηγητή: Ουγιάρογλου Στέφανος
Ημερομηνία ανάληψης Δ.Ε. 29/02/2024
Ημερομηνία περάτωσης Δ.Ε. …

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

«Αφιέρωση»

Με την ολοκλήρωση της πτυχιακής μου εργασίας θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέπων καθηγητή κ. Στέφανο Ουγιάρογλου για την υπόδειξη του θέματος, την αφιέρωση πολύτιμου χρόνου, την καθοδήγηση και τις πολύτιμες πληροφορίες και κυρίως για την εμπιστοσύνη του στις γνώσεις και τις ικανότητές μου.

Παράλληλα, θα ήθελα να ευχαριστήσω και τον πατέρα μου Βασίλειο, σαν μια ευκαιρία για να εκφράσω την ελάχιστη ένδειξη σεβασμού κι ευγνωμοσύνης για την υπομονή κι επιμονή που επέδειξε σε όλη τη διάρκεια των σπουδών μου.

Πρόλογος

Η εποχή μας χαρακτηρίζεται από μια πληθώρα δεδομένων. Ο τεράστιος όγκος τους, η ποικιλία και ο ολοένα αυξανόμενος ρυθμός δημιουργίας τους, καθιστούν δύσκολη την ανάλυσή τους χωρίς τη βοήθεια εργαλείων και τεχνικών εξαγωγής της πληροφορίας. Η προσπάθεια της εξερεύνησης διαφόρων τεχνικών εύρεσης συστάδων βάσει πυκνότητας, στην παρούσα εργασία, κατέληξε στη σύγκριση διαφόρων αλγορίθμων ομαδοποίησης και στην καταγραφή των θετικών και αρνητικών πλευρών του καθενός. Μέσω της πειραματικής μελέτης που ακολουθεί, θα ανακαλύψουμε τους παράγοντες που οδηγούν στον τύπο της κατάλληλης τεχνικής συσταδοποίησης που θα χρησιμοποιηθεί εν τέλει. Προκειμένου ωστόσο να επιτευχθεί το βέλτιστο αποτέλεσμα, απαιτείται τεράστια προσοχή, επιμονή και υπομονή.

Περίληψη

Η τεχνική της ομαδοποίησης των δεδομένων κατά συστάδες αποτελεί ισχυρό εργαλείο που μπορεί να εφαρμοστεί σε περιπτώσεις όπου υπάρχει πλήθος δεδομένων και κρίνεται σημαντική η επεξεργασία τους. Η συσταδοποίηση η οποία βασίζεται στην πυκνότητα αποτελεί το κύριο θέμα της παρούσας εργασίας και βάση της θεωρητικής ανασκόπησης και προγραμματιστικής ανάλυσης των πέντε αλγορίθμων DBSCAN, DENCLUE, Mean Shift, OPTICS και HDBSCAN σε γλώσσα προγραμματισμού Python, καθώς και της εκτενής πειραματικής μελέτης που πραγματοποιήθηκε σε δώδεκα σύνολα δεδομένων, με την χρήση της Silhouette Score αναδείχθηκε η ποιότητα της συσταδοποίησης στην κάθε μοναδική περίπτωση που παράγεται μέσω αυτής της τεχνικής.

«Density-based clustering: Literature review and experimental study»

«Balaska Stavroula»

Abstract

The clustering technique is a powerful tool that can be applied in cases where there is a large amount of data and it is important to process it. Density-based clustering is the main topic of this paper and based on the theoretical review and programmatic analysis of the five algorithms DBSCAN, DENCLUE, Mean Shift, OPTICS and HDBSCAN in Python programming language, as well as the extensive experimental study conducted on twelve datasets, using Silhouette Score we highlight the quality of clustering, in each unique case generated through this technique.

Συντομογραφίες

ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Περιεχόμενα

Κατάλογος Πινάκων

Κεφάλαιο 1ο: Εισαγωγή

1.1 Συσταδοποίηση

Η μηχανική μάθηση (Machine learning) η οποία αποτελεί μέρος της επιστήμης τον υπολογιστών, έχει ως στόχο την ανακάλυψη μοτίβων σε δεδομένα χωρίς να έχει τη γνώση πάνω σε αυτά. Αυτό πετυχαίνεται με τη χρήση αλγορίθμων, οι οποίοι μελετούν δεδομένα και δημιουργούν σχέσεις μεταξύ τους ώστε να μπορούν να προβλέψουν αλλά και να παίρνουν αποφάσεις μεταγενέστερα. Χωρίζεται σε τρείς διαφορετικές κατηγορίες: την επιβλεπόμενη μάθηση (Supervised learning), η οποία λαμβάνει πληροφορίες που προέρχονται από ήδη έτοιμες ετικέτες (labels) και μαθαίνει να διαχωρίζει τα δεδομένα βάσει αυτών, όπως είναι η ταξινόμηση(Classification), την μη επιβλεπόμενη μάθηση (Unsupervised learning),η οποία αφορά αλγορίθμους που δεν λαμβάνουν υπόψη ή δε γνωρίζουν από πριν τις ετικέτες των δεδομένων και αναζητούν τη συσχέτιση μεταξύ τους βάση της απόστασης ή της ομοιογένειας τους, όπως είναι η συσταδοποίηση(Clustering) με την οποία θα ασχοληθούμε και η ενισχυτική μάθηση (Reinforcement learning), η οποία καλείται να λάβει αποφάσεις βασιζόμενη στην αλληλεπίδραση που έχει με το γύρω περιβάλλον της.[1][2]

Αναλύοντας περισσότερο το κεντρικό κομμάτι της εργασίας που αφορά τη συσταδοποίηση, εντοπίζονται ορισμένες αρχές που η τεχνική αυτή ακολουθεί με στόχο την διεκπεραίωση της. Η συσταδοποίηση, ανήκει στην μη επιβλεπόμενη μηχανική μάθηση και στόχος της είναι να ομαδοποιήσει τα δεδομένα που της δίνονται σύμφωνα με την ομοιότητα ή την απόσταση τους και εντοπίζοντας μοτίβα που μπορεί αυτά να δημιουργούν. Για να επιτευχθεί αυτό, χρησιμοποιείται κάποιο μέτρο ομοιότητας ή απόστασης όπως είναι η Ευκλείδεια απόσταση, η οποία εξετάζει τα σημεία σύμφωνα με τις αποστάσεις τους και ανάλογα τα κατατάσσει σε συστάδες. Ο διαχωρισμός των δεδομένων σε ομάδες είναι ιδιαίτερα σημαντικός αφού με τον τρόπο αυτό τα σύνολα δεδομένων γίνονται περισσότερο κατανοητά και εύκολα προς χρήση αργότερα. Επιπλέον, υπάρχει μια γκάμα αλγορίθμων που δημιουργήθηκαν για το σκοπό αυτό και η επιλογή τους αποτελεί πρόκληση με την αξιολόγηση του αργότερα να κρίνεται υψίστης σημασίας.

Συνεπώς, η τεχνική της συσταδοποίησης αποτελεί σημαντικό κομμάτι της ανάλυσης και μπορεί εύκολα να εφαρμοστεί σε ποικίλες περιπτώσεις όπως είναι η αναγνώριση προτύπων(φωνής, εικόνας κλπ.), η κατηγοριοποίηση πελατών ή προϊόντων στο μάρκετινγκ σύμφωνα με τα κοινά τους χαρακτηριστικά όπως τις προτιμήσεις τους για να μπορέσουν να εντοπιστούν οι τάσεις στην αγορά, η ομαδοποίηση των ασθενών στον τομέα της Ιατρικής για εξατομικευμένη φροντίδα αυτών, η ανάλυση κλιματικών αλλαγών που συμβαίνει από πόλη σε πόλη και στην λήψη μέτρων για την αποφυγή αυτών, αλλά και σε πολλούς επιπλέον κλάδους.[3][26]

1.2 Βασικές Τεχνικές Συσταδοποίησης

1.2.1 Συσταδοποίηση με βάση τη κατάτμηση (Partition-based clustering)

Οι αλγόριθμοι συσταδοποίησης με βάση τη κατάτμηση (Partition-based clustering[4]) ανήκουν στη μέθοδο μη επιβλεπόμενης μάθησης, η οποία έχει ως στόχο τον διαχωρισμό των δεδομένων σε έναν συγκεκριμένο αριθμό συστάδων (clusters). Η κεντρική ιδέα της τεχνικής αυτής βασίζεται στον διαχωρισμό των δεδομένων σε διακριτά κομμάτια μέσω μιας παραμέτρου k η οποία δίνεται από τον χρήστη και ορίζει τον ακριβή αριθμό συστάδων που θέλει να δημιουργήσει ο αλγόριθμος για το εκάστοτε σύνολο δεδομένων. Στόχος είναι η ομοιογένεια μεταξύ των αντικειμένων που θα ενταχθούν σε μια συστάδα σε σχέση με τα δεδομένα των υπόλοιπων ομάδων.

Εξετάζοντας τον τρόπο με τον οποίο λειτουργεί η partition-based συσταδοποίηση, ως πρώτη μέριμνα για κάθε αλγόριθμο ορίζεται ο καθορισμός των διαφορετικών περιοχών k από τον χρήστη όπως αναφέρθηκε. Ο αριθμός αυτός είναι υποχρεωτικός κατά την είσοδο ώστε να μπορέσει να πραγματοποιηθεί η κατάτμηση. Έπειτα, ο αλγόριθμος επιλέγει τυχαία σημεία του χώρου ώστε να ξεκινήσει. Στις πιο κοινές μεθόδους όπως είναι ο K-means, τα δεδομένα ανατίθενται στο κοντινότερο σε αυτά κέντρο (centroid). Τα centroid είναι σημεία που αντιπροσωπεύουν τις συστάδες και

αποτελούν το μέσο όρο όλων των σημείων που ανήκουν σε καθεμία από αυτές. Στην αρχή της διαδικασίας τα κέντρα αυτά δίνονται σε τυχαία σημεία ενώ κατά τη διάρκεια εκτέλεσης τα κέντρα υπολογίζονται σύμφωνα με το μέσο όρο κάθε φόρα και ανανεώνονται όσο προστίθενται νέα σημεία στις συστάδες. Η ενέργεια αυτή συνεχίζει επαναληπτικά μέχρι οι αλλαγές στα κέντρα των συστάδων να μην υπάρχουν ή να είναι αρκετά μικρές, ή μέχρι η διαδικασία να σταματήσει από κάποιο κριτήριο διακοπής. Στο τέλος, κάθε σημείο θα πρέπει να ανήκει αποκλειστικά και μόνο σε μία ομάδα και υποχρεωτικά όλα τα σημεία του χώρου να ανήκουν με τη σειρά τους σε κάποια συστάδα.

Η συσταδοποίηση με βάση την κατάτμηση παρουσιάζει πολλά οφέλη που την καθιστούν δημοφιλή και χρησιμοποιείται ευρέως σε διάφορες εφαρμογές όπως η επεξεργασία εικόνας στην οποία εντοπίζονται τα pixel με βάση το χρώμα τους ή η ανάλυση της έκφρασης των γονιδίων για κατανόηση διαφόρων βιολογικών διαδικασιών. Τέτοιοι αλγόριθμοι γίνονται εύκολα κατανοητικοί και έχουν απλή εφαρμογή, είναι αποδοτικοί χωρίς να χρειάζονται πολύ χρόνο και είναι αρκετά προσαρμοστικοί με αποτέλεσμα να είναι κατάλληλοι σε μεγάλους όγκους και διαφόρων διαστάσεων δεδομένων.

Από την άλλη πλευρά, όπως κάθε τεχνική, η partition-based clustering παρουσιάζει με τη σειρά της ορισμένες αδυναμίες. Η έλλειψη γνώσης για το πως είναι δομημένα τα δεδομένα μπορεί να οδηγήσει σε λανθασμένη επιλογή της παραμέτρου k με αποτέλεσμα να μη γίνει καλή ομαδοποίηση των δεδομένων ενώ και οι ακραίες τιμές που μπορεί να εντοπιστούν, μεταβάλλουν σημαντικά τη θέση των κέντρων. Αν και γρήγοροι αλγόριθμοι, η απαίτηση πολλών επαναλήψεων για την εύρεση του νέου κέντρου, μπορεί να αυξήσει το συνολικό υπολογιστικό κόστος.[24][25]

1.2.2 Ιεραρχική Συσταδοποίηση (Hierarchical Clustering)

Στην μηχανική μάθηση, η ιεραρχική συσταδοποίηση (Hierarchical clustering[5]) η οποία αναφέρεται συνήθως και ως ιεραρχική ανάλυση συστάδων (Hierarchical Cluster Analysis, HCA) είναι μια μέθοδος με την οποία τα δεδομένα ομαδοποιούνται σε διάφορα επίπεδα ως ένα δενδρόγραμμα (dendrogram).

Υπάρχουν δύο προσεγγίσεις στην ιεραρχική ανάλυση συστάδων:

Συσσωρευτική Ιεραρχική συσταδοποίηση (Agglomerative Hierarchical Clustering, AHC): Στην περίπτωση αυτή, κάθε σημείο του χώρου των δεδομένων αντιμετωπίζεται ως μια ανεξάρτητή συστάδα. Στόχος της είναι η δημιουργία ενός μοναδικού cluster στο οποίο θα εμπεριέχονται όλα τα δεδομένα. Για να το επιτύχουν αυτό, οι ιεραρχικοί αλγόριθμοι ενώνουν τις συστάδες μεταξύ τους βάση είτε της απόστασης είτε της ομοιογένειας τους. Η ενέργεια αυτή λειτουργεί με επαναληπτική συγχώνευση ομάδων έως ότου μείνει μόνο μια συστάδα η οποία θα περιέχει όλα τα δεδομένα. Με το πέρας της διαδικασίας, δημιουργείται το λεγόμενο δένδρο που δείχνει όλη τη σχέση μεταξύ των σημείων κατά τη διάρκεια της συσταδοποίησης.

Ο τρόπος με τον οποίο υπολογίζονται οι νέες αποστάσεις για τη συγχώνευση των συστάδων ποικίλει σύμφωνα με τη μέθοδο σύνδεσης που χρησιμοποιείται. Οι μέθοδοι αυτοί είναι:

- Single Linkage: η απόσταση ανάμεσα στα δύο νέα cluster, υπολογίζεται με βάση την μικρότερη απόσταση μεταξύ δύο σημείων που ενώνουν τις συστάδες αυτές. Είναι ευρέως γνωστή και ως η μέθοδος πλησιέστερου γείτονα.
- Complete Linkage: η απόσταση ανάμεσα στα δύο νέα cluster, υπολογίζεται με βάση την μεγαλύτερη απόσταση δυο σημείων που ενώνουν τις συστάδες αυτές.
- Average Linkage: η απόσταση ανάμεσα στα δύο νέα cluster υπολογίζεται βάση τον μέσο όρο κάθε πιθανού ζεύγους σημείων που μπορεί να δημιουργηθεί μεταξύ των συστάδων.

Διαχωριστική Ιεραρχική συσταδοποίηση (Divisive Hierarchical Clustering, DHC). Σε αντίθεση με την συσσωρευτική συσταδοποίηση, η μέθοδος αυτή ξεκινάει με το ανάποδο γεγονός. Δηλαδή όλα τα σημεία του χώρου δεδομένων αντιμετωπίζονται ως μια συστάδα η οποία σε κάθε βήμα διασπάται σε

κομμάτια. Κάθε φορά από τη συσταδοποίηση προκύπτουν δύο ή περισσότερες ομάδες οι οποίες διαχωρίζονται με τη σειρά τους σε μικρότερες έως ότου κάθε σημείο να αποτελεί τη δική του μοναδική συστάδα.

Η ιεραρχική συσταδοποίηση παρουσιάζει πολλά προτερήματα που καθιστούν τη μέθοδο αυτή μια ισχυρή μέθοδο για την ανακάλυψη μοτίβων σε διαφορετικά επίπεδα. Η παρέμβαση του χρήστη δεν είναι απαραίτητη σε αυτή την τεχνική με τον προσδιορισμό αριθμού ομάδων αφού ανάλογα με τη μέθοδο που χρησιμοποιείται κάθε σημείο δημιουργεί μια συστάδα ή όλα εντάσσονται σε μια ανάλογα, κατάσταση την οποία ορίζει η ίδια η τεχνική. Επιπλέον, η απεικόνιση που προσφέρει μέσω δενδρογράμματος δείχνει τα διάφορα επίπεδα που δημιουργούνται κατά τη συσταδοποίηση αλλά και τις σχέσεις που υπάρχουν μεταξύ των συστάδων ενώ η διαδικασία με αυτόν τον τρόπο γίνεται ιδιαίτερα κατανοητή. Παράλληλα, το γεγονός ότι διαθέτει αρκετές μεθόδους με τις οποίος μπορεί να υπολογιστεί η απόσταση, την βοηθά ώστε να προσαρμόζεται με μεγαλύτερη ευκολία στις απαιτήσεις του προβλήματος.

Από την άλλη μεριά, όπως συμβαίνει με κάθε τεχνική έτσι και στην ιεραρχική συσταδοποίηση υπάρχουν αρκετά μειονεκτήματα τα οποία θα πρέπει να ληφθούν υπόψιν πριν την επιλογή της. Δεν αποτελεί καλή επιλογή όταν αναφερόμαστε σε μεγάλα σύνολα δεδομένων, αφού για τον υπολογισμό των αποστάσεων και τη δημιουργία του δένδρου χρειάζεται μεγάλη υπολογιστική ισχύ σε μνήμη και χρόνο. Επιπλέον, όταν λαμβάνονται αποφάσεις για τον τρόπο διαχωρισμού ή ένωσης των δεδομένων, είτε στην περίπτωση του συσσωρευτικού ή του διαχωριστικού τρόπου, οι αποφάσεις αυτές πρέπει να παίρνονται με σύνεση διότι είναι μη αναστρέψιμες, δεν μπορούν να ανακαλεστούν και μπορεί εύκολα να οδηγήσουν σε μη επιθυμητά αποτελέσματα. Αν και η απεικόνιση με δενδρογράμματα σε μικρά σύνολα δεδομένων είναι αρκετά εύκολη στην κατανόηση όταν τα σύνολα δεδομένων γίνονται πολύπλοκα και υπάρχει μεγάλος αριθμός σημείων, η οπτικοποίηση γίνεται δυσκολότερη και δυσνόητη.

Στο σύνολο της η Ιεραρχική συσταδοποίηση αποτελεί ισχυρό εργαλείο που μπορεί να εφαρμοστεί σε πολλές περιπτώσεις όπως η ανάλυση κειμένου με σκοπό την κατηγοριοποίηση διαφόρων εγγράφων με παρόμοιο περιεχόμενο ακόμη και στα κοινωνικά δίκτυα για την εύρεση της σύνδεσης μεταξύ χρηστών και την ομαδοποίηση αυτών βάση τα κοινά τους ενδιαφέροντα.[25][26]

1.2.3 Συσταδοποίηση βάσει πυκνότητας (Density-Based Clustering)

Η συσταδοποίηση βάσει πυκνότητας (Density-Based Clustering[6]) αφορά μια τεχνική ομαδοποίησης των δεδομένων η οποία στηρίζεται στην ανίχνευση συστάδων βάσει της πυκνότητας των σημείων. Κύρια έννοια της density-based clustering είναι ο εντοπισμός περιοχών στο χώρο όπου τα σημεία τείνουν να συγκεντρώνονται περισσότερο και την αναγνώριση αυτών ως συστάδες. Περιοχές όπου τα σημεία δεν βρίσκονται αρκετά κοντά μεταξύ τους και δεν προσφέρουν χρήσιμες πληροφορίες για τη δημιουργία μιας ομάδας ή βρίσκονται απομονωμένα και αρκετά μακριά από μια άλλη ομάδα, τότε ορίζονται ως θόρυβος (noise) ή ακραίες τιμές (outlier) αντίστοιχα. Ωστόσο, τόσο τα σημεία θορύβου όσο και οι ακραίες τιμές μπορεί εύκολα πολλές φορές να προκύψουν και από λανθασμένη μέτρηση. Σε αντίθεση με την τεχνική συσταδοποίησης με βάση την κατάτμηση, οι density-based αλγόριθμοι δεν απαιτούν εκ των προτέρων τον αριθμό των συστάδων αφού ο διαχωρισμός προκύπτει βάση της πυκνότητας και τις απαιτήσεις κάθε αλγορίθμου.

Η συσταδοποίηση με βάση την πυκνότητα παρέχει πληθώρα πλεονεκτημάτων, καθιστώντας την εξαιρετικά χρήσιμη για την ανάλυση δεδομένων ιδιαίτερα σε περιπτώσεις όπου άλλες μέθοδοι μπορεί να αποτύχουν ή να είναι λιγότερο αποτελεσματικές. Οι density-based αλγόριθμοι έχουν την ικανότητα να μπορούν να αναγνωρίζουν συστάδες διαφόρων σχημάτων χωρίς να περιορίζονται από αυτό. Ακόμη, μπορούν να διαχειριστούν με μεγαλύτερη ευκολία τον θόρυβο και να τον απομονώσουν από τα δεδομένα χωρίς να επηρεάζουν τη συνολική ποιότητα της συσταδοποίησης. Επιπλέον, ο μη υποχρεωτικός προκαθορισμός του αριθμού συστάδων, βοηθά στο να σχηματίζονται συστάδες ανάλογα με την πυκνότητα των δεδομένων, χωρίς περιορισμό προς το πλήθος τους, γεγονός

σημαντικό όταν βρισκόμαστε αντιμέτωποι με ανομοιογενείς ομάδες ενώ δεδομένου ότι στηρίζεται στην τοπική πυκνότητα των δεδομένων, μειώνει σε μεγάλο βαθμό τη δημιουργία λανθασμένων συστάδων.

Συνεπώς, έχει αποδειχθεί πολύτιμη σε πολλές περιπτώσεις όπου έχει εφαρμοστεί, με ορισμένες από τις οποίες να είναι η ανίχνευση γεωγραφικών περιοχών υψηλής πυκνότητας δηλαδή στον εντοπισμό περιοχών όπου συγκεντρώνονται πολλά οχήματα και στο μοτίβο που αυτά δημιουργούν για τις μεταγενέστερες αποφάσεις, στην ανίχνευση ανωμαλιών σε δίκτυα δηλαδή στην ανίχνευση ασυνήθιστων συμπεριφορών στα δεδομένα που μπορεί να δείχνουν σε κακόβουλες δραστηριότητες αλλά ακόμα και στην αστρονομία, στην ανάλυση της κατανομής των αστεριών και στις πυκνές δομές που δημιουργούν μεταξύ τους, βοηθώντας έτσι τους επιστήμονες να μπορούν να τα αναγνωρίζουν και να τα μελετούν με μεγαλύτερη ευκολία και δίνοντας διάφορες χρήσιμες πληροφορίες για την κατάσταση ολόκληρου του γαλαξία.

Αν και οι αλγόριθμοι συσταδοποίησης που βασίζονται στην πυκνότητα προσφέρουν αρκετά προτερήματα, παρουσιάζουν επίσης και ορισμένα ελαττώματα. Αρκετοί από αυτούς, απαιτούν τον ορισμό συγκεκριμένων παραμέτρων στην αρχή της διαδικασίας και η λανθασμένη επιλογή τους μπορεί να οδηγήσει στην αχρήστευση σημαντικών για την συσταδοποίηση πληροφοριών. Επιπλέον, η ανακριβής παραμετροποίηση μπορεί να δημιουργήσει έναν υπερβολικό διαχωρισμό στα δεδομένα σε πολλές συστάδες ή μια συγχώνευση σημαντικών πληροφοριών σε μικρότερες ενώ η επιλογή τους κρίνεται δύσκολη ακόμη και σε περιπτώσεις όπου η πυκνότητα αλλάζει σημαντικά από περιοχή σε περιοχή. Ακόμη, όταν τα δεδομένα εμφανίζονται σε πολλές διαστάσεις, γίνονται πιο αραιά και η εύρεση των γειτονικών τους σημείων γίνεται λιγότερο ξεκάθαρη.

Σε σύγκριση με άλλες τεχνικές ομαδοποίησης, λόγω των διαφορετικών σχημάτων που παράγονται αλλά και του θορύβου που μπορεί να παρουσιάζουν, τα αποτελέσματα μπορεί να γίνουν αρκετά δυσνόητα και οι αλγόριθμοι να δυσκολευτούν να εντοπίσουν τα cluster. Ομοίως δύσκολη κρίνεται και η διαχείριση σημείων που μπορεί να βρίσκονται πολύ κοντά στα όρια μεταξύ δυο διαφορετικών ομάδων ή να αλληλεπικαλύπτονται μεταξύ τους και να μη μπορεί ο αλγόριθμος να προσδιορίσει την συστάδα στην οποία ανήκουν. Παρόλο που η συσταδοποίηση με βάση την πυκνότητα μπορεί να είναι αποδοτική για διάφορα μεγέθη dataset, όταν το μέγεθος των δεδομένων μεγαλώνει, αυξάνονται και οι απαιτήσεις τους προβλήματος, κατάσταση η οποία μπορεί να επιβραδύνει τον αλγόριθμο που θα χρησιμοποιηθεί.[25][26]

1.3 Κίνητρο και Συνεισφορά

Η επιθυμία μου να εξερευνήσω βιβλιογραφικά διάφορες τεχνικές μηχανικής μάθησης και να πειραματιστώ προγραμματιστικά με νέους αλγορίθμους ιδίως στην γλώσσα Python, αλλά και έπειτα από τη συμβολή και την παρότρυνση του επιβλέποντα καθηγητή μου κ. Στέφανου Ουγιάρογλου, προήλθε η επιλογή του θέματος «Συσταδοποίηση βάσει πυκνότητας (Density-based Clustering)». Στόχος μου, με το πέρας της εργασίας και της πειραματικής μελέτης που πραγματοποιήθηκε, είναι η προσφορά μιας σαφής εικόνας του τρόπου λειτουργίας των αλγορίθμων συσταδοποίησης που βασίζονται στην πυκνότητα σε κάθε πιθανό ενδιαφερόμενο προς το θέμα αυτό.

1.4 Οργάνωση της εργασίας

Η παρούσα εργασία έχει οργανωθεί σε πέντε κεφάλαια καθένα από τα οποία εξετάζει μια διαφορετική πτυχή. Αρχίζοντας από το πρώτο κεφάλαιο, γίνεται μια εισαγωγή στην μηχανική μάθηση και συγκεκριμένα στο κομμάτι της συσταδοποίησης αλλά και στις διάφορες τεχνικές που υπάρχουν για το σκοπό αυτό. Έπειτα, το δεύτερο κεφάλαιο εστιάζεται αποκλειστικά στην τεχνική της συσταδοποίησης που είναι βασισμένη στην πυκνότητα και αποτελεί το κύριο θέμα της εργασίας μου, αναλύοντας βιβλιογραφικά κάθε αλγόριθμο που εμπεριέχεται σε αυτήν (DBSCAN, DENCLUE, Mean Shift, OPTICS και HDBSCAN). Στο τρίτο κεφάλαιο γίνεται η προγραμματιστική ανάλυση αυτών των αλγορίθμων στον τρόπο εκτέλεσής τους με τη χρήση της βιβλιοθήκης Scikit-learn και τη γλώσσα προγραμματισμού Python, ενώ στο τέταρτο κεφάλαιο παρουσιάζεται η πειραματική μελέτη που

πραγματοποιήθηκε πάνω σε δώδεκα διαφορετικά σύνολα δεδομένων με τον σχολιασμό, την απεικόνιση αυτών σε σχεδιαγράμματα αλλά και την αξιολόγηση τους με τη χρήση της μετρικής Silhouette score. Τέλος, το πέμπτο κεφάλαιο είναι αφιερωμένο στα συμπεράσματα και τις προτάσεις βελτίωσης που αποκομίσθηκαν από όλη την εργασία.

Κεφάλαιο 2ο: Αλγόριθμοι Συσταδοποίησης βάσει πυκνότητας (Density-based algorithms)

2.1 Συσταδοποίηση Βάσει Πυκνότητας

Αλγόριθμοι που δημιουργήθηκαν με σκοπό να διαχωρίζουν τα δεδομένα βάσει της πυκνότητας τους αποτελούν οι DBSCAN[8], DENCLUE[10], Mean Shift[12], OPTICS[14] και HDBSCAN[17] και για κάθε ένα από αυτούς δίνεται παρακάτω η βιβλιογραφική ανάλυση:

2.1.1 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Αν και οι αλγόριθμοι με βάση την κατάτμηση και οι Ιεραρχικοί αλγόριθμοι ομαδοποιούσαν για χρόνια τα δεδομένα είτε βασισμένοι σε ένα αντιπροσωπευτικό σημείο κέντρο είτε βρίσκοντας την ιεραρχική δομή αυτών αντίστοιχα, όταν ο όγκος των πληροφοριών άρχισε να αυξάνεται, ο χρόνος εκτέλεσης γινόταν όλο ένα και μεγαλύτερος μέχρι που δεν ήταν πλέον αποδεκτός. Η απαλοιφή των σημείων θορύβου από τα δεδομένα δεν ήταν πια αποτελεσματική και η ανίχνευση συστάδων διαφορετικών σχημάτων από τα μέχρι τότε προκαθορισμένα αποδείχθηκε δύσκολη. Η ανάγκη δημιουργίας ενός νέου τρόπου διαχωρισμού των δεδομένων ήταν η μόνη λύση.

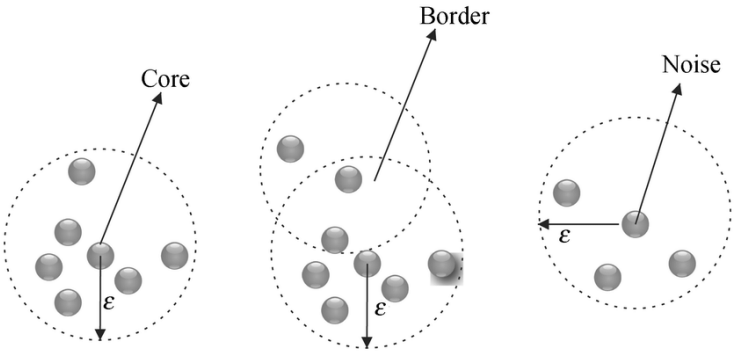
Συνεπώς, το 1996 επιστήμονες κατέθεσαν μια νέα ιδέα για έναν αλγόριθμο συσταδοποίησης ο οποίος θα αντιμετώπιζε τα προβλήματα που είχαν δημιουργηθεί. Η θεωρία αυτή πρότεινε τον διαχωρισμό των δεδομένων από την πλευρά της πυκνότητας. Ο νέος αλγόριθμος αυτός ονομάστηκε Density-Based Spatial Clustering of Applications with Noise, ή αλλιώς DBSCAN, και αποδείχθηκε αρκετά πιο ανθεκτικός στη αντιμετώπιση των δυσκολιών που προκαλούσε ο θόρυβος και προσαρμοζόταν με μεγαλύτερη ευκολία στις ανάγκες που απαιτούσε κάθε ανάλυση. Πλέον, ο DBSCAN αποτελεί έναν από τους πιο ευρέως γνωστούς αλγορίθμους συσταδοποίησης που βασίζεται στην πυκνότητα των δεδομένων.

Βασικός στόχος του αλγορίθμου DBSCAN, αποτελεί ο εντοπισμός συστάδων όπου υπάρχει η μεγαλύτερη συγκέντρωση σημείων και αγνοώντας τμήματα όπου τα σημεία δεν είναι τόσο κοντά μεταξύ τους για να δημιουργήσουν μια πυκνή περιοχή. Σε αντίθεση με άλλους αλγορίθμους συσταδοποίησης, ο DBSCAN δεν απαιτεί τον προκαθορισμό του αριθμού των συστάδων αλλά είναι σημαντικό για τη λειτουργία του να δοθεί ο εκ των προτέρων προσδιορισμός δύο άλλων παραμέτρων από των χρήστη, ο Eps (ϵ) και ο MinPts (Minimum Points). Η παράμετρος Eps αναφέρετε σε μία απόσταση ή ακτίνα γύρω από κάθε σημείο του χώρου και δείχνει την μέγιστη απόσταση η οποία είναι επιτρεπτή ώστε δυο σημεία να θεωρούνται ότι ανήκουν στο ίδιο cluster ενώ η παράμετρος MinPts από την άλλη πλευρά ορίζει τον ελάχιστο αριθμό σημείων που θα πρέπει να εμπεριέχονται εντός της εμβέλειας Eps που έχει δοθεί ώστε να θεωρηθεί η περιοχή του σημείου ως πυκνή.[7][8]

Υπάρχουν τρείς κατηγορίες σημείων στις οποίες μπορούν να χωριστούν τα σημεία στο χώρο κατά τη διαδικασία της συσταδοποίησης:

- Core points: Τα σημεία πυρήνα (core points), είναι τα σημεία εκείνα που εντός μιας ακτίνας Eps γύρω τους, περιέχουν τουλάχιστον τόσα γειτονικά σημεία όσα έχουν οριστεί από την παράμετρο MinPts, συμπεριλαμβάνοντας σε αυτά και τον εαυτό τους. Η λέξη πυρήνας προσδιορίζει ότι τα σημεία αυτά βρίσκονται στο κέντρο της συστάδας κατά την διάρκεια έλεγχου τους.
- Border point: Τα σημεία ορίου (border points), είναι τα σημεία εκείνα που εντός μιας ακτίνας Eps γύρω τους, δεν περιέχουν αρκετά γειτονικά σημεία σύμφωνα με το κριτήριο του MinPts ώστε να θεωρηθούν ως σημεία πυρήνα αλλά βρίσκονται στα όρια της γειτονιάς ενός άλλου σημείου core.
- Noise point: Τα σημεία θορύβου (Noise points), είναι τα σημεία εκείνα που εντός μιας ακτίνας Eps γύρω τους, δεν περιέχουν τόσα γειτονικά σημεία όσα ορίζει ο MinPts ώστε να θεωρηθούν ως σημεία πυρήνα αλλά ούτε βρίσκονται στα όρια μιας γειτονιάς ενός core ώστε

να θεωρηθούν ως border. Αποτελούν δηλαδή σημεία απομονωμένα που δεν σχετίζονται με άλλα σημεία στον χώρο των δεδομένων.[7][24]



https://www.researchgate.net/figure/DBSCAN-core-border-and-noise-points_fig1_258442676

Εικόνα 2.1 Κατηγορίες σημείων στο χώρο

Η διαδικασία συσταδοποίησης με τη χρήση του αλγορίθμου DBSCAN γίνεται ως εξής: Αρχικά ορίζονται από τον χρήστη οι παράμετροι Eps και MinPts. Όλα τα σημεία στην αρχή θεωρούνται ως μη επισκέψιμα (unvisited). Έπειτα ο αλγόριθμος DBSCAN επιλέγει ένα τυχαίο σημείο στο χώρο των δεδομένων. Για να μπορέσει αυτό το σημείο να τοποθετηθεί σε μία συστάδα θα πρέπει να πληρεί ορισμένα κριτήρια που ορίζουν οι παράμετροι Eps και MinPts που δίνονται. Ο αλγόριθμος τότε ελέγχει το σημείο σύμφωνα με την ακτίνα και τα αντίστοιχα γειτονικά του σημεία. Στον πρώτο έλεγχο που διεξάγεται, μπορούν να υπάρχουν μόνο δύο εκδοχές για το αποτέλεσμα. Αν το σημείο πληρεί το κριτήριο του MinPts, τότε ορίζεται ως core point αφού είναι το πρώτο σημείο εξέτασης και δεν υπάρχουν άλλα κατηγοριοποιημένα σημεία. Στην περίπτωση αυτή, δημιουργείται μια συστάδα με αντιπρόσωπο το σημείο που εξετάστηκε ενώ όλα τα σημεία που εντοπίστηκαν γύρω από αυτό και εντός της ακτίνας Eps αποτελούν πλέον μια γειτονιά. Ως επόμενο σημείο προς εξέταση επιλέγεται κάποιο από τα σημεία γείτονες. Στην αντίθετη περίπτωση που το σημείο δεν πληρεί το κριτήριο του MinPts, τότε θεωρείται ως μη ταξινομημένο έως ότου εξεταστούν τα γειτονικά του σημεία σε κάποιον επόμενο έλεγχο και επιλέγεται ένα επόμενο τυχαίο σημείο του χώρου για αναζήτηση. Η διαδικασία ελέγχου συνεχίζεται για όλα τα σημεία του χώρου μέχρι να ταξινομηθούν όλα αναλόγως. Για κάθε σημείο πυρήνα που θα εντοπίσει ο αλγόριθμος, είτε το προσθέτει σε κάποια ήδη υπάρχουν συστάδα εφόσον αποτελεί στοιχείο της ίδιας γειτονιάς είτε δημιουργεί μια νέα ομάδα και κάθε φορά συνεχίζει τον έλεγχο από τα γειτονικά στοιχεία του τελευταίου Core point που εντόπισε. Όταν σταματήσουν να εντοπίζονται core points, ο αλγόριθμος προχωρά σε ένα άλλο μη επισκέψιμο σημείο και η διαδικασία αρχίζει από την αρχή. Σημεία που έχουν οριστεί ως Border point με τη σειρά τους εντάσσονται στη συστάδα της γειτονιάς στην οποία ανιχνεύθηκαν. Στο τέλος της διαδικασίας, σημεία που δεν κατάφεραν να ενταχθούν σε κάποια συστάδα, ορίζονται ως θόρυβος (Noise point).

Ο τρόπος λειτουργίας του DBSCAN θεωρείται σχετικά εύκολος να υλοποιηθεί και η πληθώρα πλεονεκτημάτων που έχει, οδήγησαν τον αλγόριθμο αυτό σήμερα να είναι ένας από τους πιο διαδεδομένους σε παγκόσμιο επίπεδο. Το γεγονός ότι βασίζεται στην πυκνότητα των δεδομένων και η ικανότητα του να εντοπίζει και να αναγνωρίζει συστάδες χωρίς συγκεκριμένη μορφή, τον έχουν ορίσει ως έναν αρκετά λειτουργικό αλγόριθμο. Σημαντική είναι εξίσου και η δυνατότητα που έχει να αναγνωρίζει με ευκολία τα σημεία θορύβου και να τα διαχειρίζεται, χωρίς να επηρεάζεται άμεσα από αυτά. Σε αντίθεση με άλλους αλγορίθμους, το γεγονός ότι δεν απαιτείται από τον χρήστη να προσδιορίσει εξ αρχής των αριθμό των ομάδων που θα δημιουργηθούν, δίνει στον αλγόριθμο μεγαλύτερη ελευθερία στο να σχηματίζει αυτές τις συστάδες δυναμικά βάσει της πυκνότητας τους.

Αν και αρκετά ωφέλιμος, πολλοί είναι και οι περιορισμοί του αλγορίθμου που θα πρέπει να ληφθούν υπόψιν. Οι δυο βασικοί παράμετροι που δίνονται από τον χρήστη, οι παράμετροι Eps και MinPts θα πρέπει να επιλέγονται με σύνεση έπειτα από πειραματισμούς και μέσω ειδικών γραφημάτων προσαρμοσμένων για το σκοπό αυτό για την αποφυγή τυχόν σφαλμάτων. Μία μικρή τιμή στην

ακτίνα Eps μπορεί εύκολα να οδηγήσει σε πολλές μικρές ομάδες ή ακόμα και σε περισσότερα noise points στο χώρο, ενώ μια μεγάλη τιμή αντίστοιχα, μπορεί να κατευθύνει τον αλγόριθμο στην συγχώνευση δεδομένων ανόμοιων μεταξύ τους και στην απώλεια χρήσιμων πληροφοριών για την ανάλυση που μπορεί να πρόσφεραν άλλες μικρότερες ομάδες. Αν και αποτελεσματικός σε διάφορες μορφές ομάδων, όταν τα δεδομένα εμφανίζονται με περισσότερες διαστάσεις, η αποδοτικότητα του μειώνεται αφού τα δεδομένα τότε τείνουν να γίνονται πιο αραιά ενώ ακόμη, μπορεί να θεωρηθεί και ακατάλληλος σε μεγάλα σύνολα δεδομένων όπου ανεβαίνει ο υπολογιστικός χρόνος που απαιτείται για την εύρεση των γειτόνων κάθε σημείου.[24][25]

2.1.2 DENCLUE (Density-based Clustering)

Αν και ο αλγόριθμος DBSCAN είχε φέρει μεγάλες αλλαγές στον έως τώρα διαχωρισμό των δεδομένων χρησιμοποιώντας την πυκνότητα και εντοπίζοντας τη γειτνίαση μεταξύ των σημείων, η επιστήμη αποφάσισε το 1998 πως ορισμένες αδυναμίες του θα μπορούσαν να καλυφθούν με τη δημιουργία ενός νέου αλγορίθμου που θα βασίζεται στην ίδια ιδέα της πυκνότητας αλλά θα προσφέρει μια άλλη οπτική στα δεδομένα, χρησιμοποιώντας έναν μαθηματικό τρόπο. Τότε δημιουργήθηκε ο αλγόριθμος DENCLUE (Density-based Clustering).

Ο αλγόριθμος DENCLUE, βασίζεται στην ιδέα ότι κάθε σημείο στο χώρο των δεδομένων συνεισφέρει με τον τρόπο του στην πυκνότητα της περιοχής στην οποία βρίσκεται. Για να το καταφέρει αυτό, λαμβάνει υπόψη του την κατανομή των σημείων στο χώρο, διαχωρίζοντας τις πυκνές από τις αραιές περιοχές, αλλά και εντοπίζοντας πως η συγκέντρωση αυτή μεταβάλλεται ανά τμήμα. Συγκεκριμένα, ο αλγόριθμος DENCLUE εφαρμόζει μία συνάρτηση πυκνότητας, μια γκαουσιανή συνάρτηση (Gaussian function), η οποία ορίζει πως η επιρροή που έχει κάθε σημείο στην περιοχή του, πηγάζει από την απόσταση που έχουν μεταξύ τους. Όσο πιο κοντά βρίσκονται τα σημεία, τόσο περισσότερο αυξάνεται και η πυκνότητα της περιοχής. Η συνάρτηση αυτή για να μπορέσει να υπολογίσει τη συνεισφορά κάθε σημείου λαμβάνει υπόψη την πυκνότητα γύρω του ενώ βασική παράμετρος για την εκτέλεση του αλγορίθμου και των εντοπισμό των επιρροών εν τέλει αποτελεί η bandwidth η οποία δίνεται από τον χρήστη και αφορά την ακτίνα γύρω από την οποία ο αλγόριθμος εξετάζει τα σημεία κάθε φορά.[11][24]

Η διαδικασία συσταδοποίησης με τη χρήση του αλγορίθμου DENCLUE γίνεται ως εξής: Αρχικά ο χρήστης δίνει τιμή στην παράμετρο Bandwidth. Για την εκτέλεση δεν επιλέγεται ένα σημείο όπως στον αλγόριθμο DBSCAN, αλλά η διαδικασία ξεκινάει με όλα τα σημεία ταυτόχρονα. Συγκεκριμένα, υπολογίζονται οι Ευκλείδειες αποστάσεις μεταξύ όλων των σημείων του χώρου και για κάθε ένα ξεχωριστά εφαρμόζεται η Gaussian συνάρτηση. Ο μαθηματικός τρόπος αυτός για να μπορέσει να εντοπίσει την επιρροή που ασκεί κάθε σημείο στην περιοχή του, λαμβάνει υπόψη για κάθε ένα μόνο τα σημεία εκείνα που βρίσκονται εντός του εύρους bandwidth που έχει οριστεί αλλά και την Ευκλείδεια απόσταση που υπολογίστηκε για αυτά σε προηγούμενο βήμα. Η συνάρτηση ορίζει πως όσο πιο μικρή είναι η απόσταση μεταξύ δύο σημείων τόσο μεγαλύτερη είναι και η συνεισφορά τους. Η συνολική πυκνότητα για κάθε σημείο προκύπτει από το άθροισμα των συνεισφορών όλων των σημείων που αποτελούν την γειτονιά του εντός του bandwidth. Στη συνέχεια, αφού ο αλγόριθμος εκτιμήσει την πυκνότητα για κάθε σημείο, και με σκοπό να εντοπίσει τις συστάδες, χρησιμοποιεί τις τιμές που προέκυψαν ώστε να κατευθυνθεί από το ένα στοιχείο στο άλλο. Η μέθοδος που χρησιμοποιεί για να το επιτύχει αυτό ονομάζεται Gradient ascent. Ειδικότερα, η μέθοδος αυτή επιλέγει ένα τυχαίο σημείο και με μια ανοδική πορεία προχωράει από αυτό στο επόμενο. Στόχος της είναι η εύρεση μιας κορυφής ώστε να δημιουργηθεί η συστάδα.

Έστω ότι επιλέγεται ένα σημείο x1. Ο αλγόριθμος θα βρει την περιοχή που καταλαμβάνει το x1 σύμφωνα με το bandwidth που έχει δοθεί, και θα επιλέξει το επόμενο σημείο x2 βάσει της πυκνότητας γύρω του. Εφόσον υπάρχει σημείο x2 που έχει πυκνότητα μεγαλύτερη από το x1, τότε η διαδικασία συνεχίζεται με τον ίδιο τρόπο μέχρι να φτάσει σε σημείο xi όπου κανένας γείτονας δεν έχει τιμή πυκνότητας μεγαλύτερη από τη δική του ή οι τιμές τους συγκλίνουν. Εκείνη τη στιγμή, η διαδικασία θεωρεί ότι βρήκε την κορυφή, το τοπικό μέγιστο της περιοχής, σταματά και δημιουργείται μια συστάδα. Στην περίπτωση που το σημείο που επιλεγεί για να ξεκινήσει έχει ήδη την μέγιστη πυκνότητα από τα σημεία γύρω του, τότε ορίζεται εξ αρχής ως κορυφή και δημιουργεί συστάδα με μοναδικό στοιχείο, έως ότου συγκλίνει ως τοπικό μέγιστο άλλης περιοχής. Οι συστάδες

σχηματίζονται από τα σημεία που ενώνονται όσο κατευθύνονται σε κάποιο τοπικό μέγιστο και εφόσον πληρούν το κριτήριο που ορίζει ένα κατώφλι (threshold). Το κατώφλι είναι αυτό που θα αποφασίσει αν θα δημιουργηθεί η συστάδα. Για να θεωρηθεί μια περιοχή πυκνή και να ανιχνευθεί ως συστάδα, θα πρέπει το τοπικό της μέγιστο να έχει πυκνότητα μεγαλύτερη από το κατώφλι που έχει δοθεί. Σημεία που δε μπορούν να ενωθούν με κάποια κορυφή και δεν ικανοποιούν την συνθήκη του threshold ορίζονται ως θόρυβος (Noise points).

Παρόλο που ο αλγόριθμος DENCLUE δεν είναι ιδιαίτερα διαδεδομένος, τα πλεονεκτήματα που προσφέρει στη συσταδοποίηση είναι αρκετά σημαντικά. Το μαθηματικό μοντέλο της γκαουσιανής συνάρτησης πάνω στο οποίο βασίζεται, δίνει στον αλγόριθμο την δυνατότητα να προσαρμόζεται με μεγαλύτερη ευκολία σε διάφορες δομές δεδομένων αφού εντοπίζει περιοχές με βάση την πυκνότητα και όχι το σχήμα τους ενώ με την ακρίβεια που γίνονται οι υπολογισμοί βάση αυτού, του εξασφαλίζουν παράλληλα μεγαλύτερη αξιοπιστία. Ακόμη, η αναγνώριση θορύβου απλουστεύεται αφού κατά τη διάρκεια του Gradient ascent τα σημεία που δεν ανήκουν πουθενά απομονώνονται κατευθείαν και με την ικανότητα του να υπολογίζει από πριν την πυκνότητα των σημείων, ο υπολογιστικός χρόνος που χρειάζεται για να παράγει σαφή αποτελέσματα μειώνεται αρκετά.

Παρά την αποτελεσματικότητα του, ο αλγόριθμος DENCLUE δε παύει να παρουσιάζει και αυτός με τι σειρά του κάποια αρνητικά σημεία, ιδιαίτερα όταν έρχεται αντιμέτωπος με μεγάλα ή πολύπλοκα σύνολα. Υπάρχει μεγάλη ευαισθησία στην επιλογή τιμής της παραμέτρου bandwidth πάνω στην οποία βασίζεται όλη η εκτέλεση του αλγορίθμου αφού μια πολύ μικρή ή μεγάλη τιμή μπορεί να οδηγήσει σε μικρότερες μη χρήσιμες συστάδες ή να χαθούν πολύτιμες πληροφορίες από τη συγχώνευση αντίστοιχα. Επιπλέον, παρά το γεγονός ότι ο υπολογισμός μαζικά της πυκνότητας των σημείων μειώνει την υπολογιστική ισχύ που χρειάζεται ο αλγόριθμος, όταν τα δεδομένα εμφανίζονται σε υψηλές διαστάσεις, οι απαιτήσεις κάθε ανάλυσης αυξάνονται ανάλογα και αυτό οδηγεί σε μεγάλες καθυστερήσεις.[10][11]

2.1.3 Mean Shift

Αν και ο αλγόριθμος DENCLUE έφερε πολλές αλλαγές με την μαθηματική προσέγγιση που προτείνει, στις αρχές του 2000 προτάθηκε ένας νέος αλγόριθμος ο οποίος αντί να χρησιμοποιεί την κατανομή των δεδομένων θα εκτιμούσε τις θέσεις των σημείων εντός ενός προκαθορισμένου εύρους bandwidth. Ο αλγόριθμος αυτός ονομάστηκε Mean Shift.

Ο αλγόριθμος Mean Shift αποτελεί μια μέθοδο διαχωρισμού των δεδομένων μετατοπίζοντας κάθε φορά το μέσο όρο των σημείων από περιοχή σε περιοχή μέχρι να βρεθεί ένα τοπικό μέγιστο όπως συμβαίνει και στον αλγόριθμο DENCLUE. Και στην περίπτωση του Mean Shift, αυτό πετυχαίνεται με τη βοήθεια ενός μαθηματικού μοντέλου γκαουσιανής συνάρτησης, η οποία υπολογίζει την εκτίμηση της πυκνότητας γύρω από κάθε σημείο αλλά μετέπειτα, μετακινεί τα δεδομένα σύμφωνα με το κέντρο βάρος τους και όχι μιας αυξητικής πορείας. Τόσο για τον προσδιορισμό της πυκνότητας όσο και για την εύρεση του μέσου μεταξύ των σημείων που σχηματίζουν μια γειτονιά, σημαντικότερος παράγοντας αποτελεί και πάλι η παράμετρος bandwidth. Στην περίπτωση του Mean Shift, η τιμή της καθορίζεται μέσω μιας διαδικασίας που εκτιμά το εύρος επιρροής βάσει της δομής των δεδομένων αυτόματα από το σύστημα αλλά μπορεί να δοθεί και χειροκίνητα από τον χρήστη.[12]

Η διαδικασία συσταδοποίησης με τη χρήση του αλγορίθμου Mean Shift γίνεται ως εξής: Αρχικά αν η τιμή του bandwidth δεν εκτιμηθεί από το σύστημα, τότε ο χρήστης οφείλει να δώσει τιμή στην παράμετρο. Σε αντίθεση με τον DENCLUE, η διαδικασία ξεκινάει επιλέγοντας ένα τυχαίο σημείο x από το χώρο των δεδομένων από όπου και θα αρχίσει η αναζήτηση. Ο αλγόριθμος ψάχνει τα σημεία κοντά στον x και κρατάει εκείνα που βρίσκονται εντός της ακτίνας bandwidth γύρω από αυτό. Τα σημεία που εντοπίζονται, αποτελούν πλέον μια γειτονιά και για κάθε ένα από αυτά υπολογίζεται η επιρροή του προς το εξεταζόμενο σημείο x. Η βαρύτητα που ασκεί καθένα πάνω στο x, εξαρτάται από την απόσταση που χρειάζονται ξεχωριστά για να φτάσουν σε αυτό. Όσο πιο κοντά βρίσκεται ένα σημείο, τόσο μεγαλύτερο είναι και το βάρος του στο x. Αφού προσδιοριστούν όλες οι συνεισφορές της περιοχής όπου βρίσκεται ο x, τότε υπολογίζεται ο μέσος όρος, δηλαδή το κέντρο βάρος των γειτονικών σημείων. Ο μέσος όρος στον Mean Shift λαμβάνει υπόψη την τοποθεσία που βρίσκεται

κάθε σημείο της γειτονιάς καθώς και το βάρος του προς τον x και επιστρέφει μία τιμή που αντιστοιχεί σε νέες συντεταγμένες και καταλαμβάνει μια καινούρια θέση στο χώρο. Έπειτα, ο αλγόριθμος συνεχίζει τη διαδικασία με τον ίδιο τρόπο πάνω στη νέα θέση που δημιουργήθηκε αναζητώντας τα γειτονικά σημεία αυτής και εντοπίζοντας και πάλι μια νέα τοποθεσία στο χώρο με την οποία θα συνεχίσει. Η ροή αυτή διακόπτεται όταν δεν υπάρχει πλέον μετακίνηση της θέσης και είτε μένει σταθερή είτε μετατοπίζεται ελάχιστα. Τότε ο αλγόριθμος θεωρεί ότι έφτασε στο τοπικό μέγιστο και δημιουργεί μια συστάδα. Έως ότου κατανεμηθούν όλα τα σημεία η διαδικασία συνεχίζει επιλέγοντας κάθε φορά ένα τυχαίο σημείο μέχρι να βρεθεί και πάλι σε ένα τοπικό μέγιστο. Σημεία που δεν έχουν κανένα γείτονα στην περιοχή τους ή μετά από κάποιο κριτήριο δεν έχουν αρκετούς ώστε να θεωρηθούν ως συστάδα, τότε ορίζονται ως θόρυβος. Αφού ο αλγόριθμος προσδιορίσει όλα τα σημεία του χώρου τερματίζει.

Υπάρχουν πολλά χαρακτηριστικά του αλγορίθμου Mean Shift που τον ξεχωρίζουν από τους υπόλοιπους αλγορίθμους και τον καθιστούν ιδιαίτερα χρήσιμο και δημοφιλή σε διάφορες εφαρμογές συσταδοποίησης. Ένα από αυτά και αρκετά σημαντικό όπως και στους προηγούμενους αλγορίθμους είναι η δυνατότητα που έχει να προσδιορίζει τον αριθμό των συστάδων δυναμικά με τη διαδικασία που εκτελεί χωρίς να χρειάζεται να δοθεί εξ αρχής ένας αριθμός και το οποίο συμβάλει θετικά σε σύνολα δεδομένων όπου η δομή τους δεν είναι γνωστή αλλά και στην αποφυγή λαθών που μπορεί να συμβούν από την χειροκίνητη ρύθμιση αυτής της παραμέτρου. Επιπλέον, ο τρόπος με τον οποίο διαχειρίζεται τα δεδομένα με τη χρήση του μαθηματικού μοντέλου, όπως και στον DENCLUE, του επιτρέπει να ανιχνεύει συστάδες διαφορετικών σχημάτων ενώ παράλληλα μπορεί πολύ εύκολα να διακρίνει και να απομονώνει τα σημεία θορύβου κατά τη διάρκεια της μετακίνησης.

Από μία διαφορετική οπτική, παρά τα θετικά στοιχεία που χαρακτηρίζουν τον Mean Shift, υπάρχουν και κάποια στοιχεία αδυναμίας που περιορίζουν την χρήση του σε ορισμένα μόνο προβλήματα. Ο αλγόριθμος εξαρτάται άμεσα από την παράμετρο bandwidth που παρότι μπορεί να εκτιμηθεί αυτόματα κατά τη διαδικασία, στην περίπτωση που δοθεί από τον χρήστη, μια λανθασμένη τιμή σε αυτήν μπορεί εύκολα να οδηγήσει σε μη σαφή διαχωρισμό των συστάδων. Επιπλέον, ακριβώς επειδή οι υπολογισμοί γίνονται ξεχωριστά για κάθε σημείο του χώρου και όχι ταυτόχρονα όπως στον DENCLUE, η απαίτηση που έχει σε υπολογιστική ισχύ είναι μεγάλη και θεωρείται ακατάλληλη όταν αντιμετωπίζει μεγάλα σύνολα δεδομένων.[12][13]

2.1.4 OPTICS (Ordering Points To Identify the Clustering Structure)

Παρόλο που πλέον υπήρχαν αρκετοί αλγόριθμοι που χώριζαν τα δεδομένα βάσει της πυκνότητας τους και η διαδικασία γινόταν όλο ένα και ευκολότερη, υπήρχαν ακόμα ορισμένες αδυναμίες που θα μπορούσαν καλυφθούν από τη δημιουργία ενός νέου αλγορίθμου που θα εύρισκε λύση σε δεδομένα με ακανόνιστο σχήμα χωρίς την ανάγκη μιας ακτίνας Eps ή κάποιου μαθηματικού μοντέλου. Τότε το 1999 ένας νέος αλγόριθμος αναπτύχθηκε και βασίστηκε πάνω στην ιδέα του DBSCAN. Ο αλγόριθμος αυτός ονομάστηκε OPTICS (Ordering Points To Identify the Clustering Structure).

Ο αλγόριθμος OPTICS, όπως και κάθε density-based αλγόριθμος, βασίζεται στην έννοια της συσταδοποίησης των δεδομένων με τη χρήση της πυκνότητας και ενώ λαμβάνει υπόψη του στοιχεία από τον DBSCAN, χρησιμοποιεί έναν διαφορετικό τρόπο προσέγγισης ομαδοποίησης των δεδομένων. Ο OPTICS διατηρεί την έννοια του MinPts που χρησιμοποιεί και ο DBSCAN για τον ελάχιστο αριθμό σημείων ώστε να σχηματιστεί μια συστάδα. Η τιμή της δίνεται συχνά χειροκίνητα ενώ στον κώδικα μπορεί να παρουσιαστεί και ως min_samples. Αν η τιμή της παραμέτρου δε δοθεί από τον χρήστη, τότε ο αλγόριθμος χρησιμοποιεί μια προεπιλεγμένη τιμή αυτόματα. Παράλληλα, εισάγεται μια νέα παράμετρος η min_cluster_size, η τιμή της οποίας μπορεί να δοθεί εξίσου από το χρήστη και ορίζεται ως ο ελάχιστος αριθμός σημείων που είναι απαραίτητος ώστε μια συστάδα να είναι έγκυρη. Η τιμή αυτή μπορεί να είναι απόλυτη ή να αναφέρεται σε ποσοστό επί της εκατό του συνολικού πλήθους του dataset. Αν και πάλι η τιμή δε δοθεί χειροκίνητα, τότε ο αλγόριθμος μπορεί να την προσαρμόσει αυτόματα ανάλογα με τα δεδομένα. Ακόμη, για την ανίχνευση των συστάδων ο OPTICS ορίζει δύο νέες έννοιες. Την απόσταση πυρήνα (Core distance) που βλέπει την πυκνότητα γύρω από ένα σημείο και ανάλογα το τοποθετεί σε συστάδα ή όχι, αλλά και την απόσταση προσβασιμότητας

(Reachability distance) που κοιτάζει την «απόσταση» μεταξύ δυο σημείων για να προσδιορίσει αν ανήκουν στην ίδια συστάδα.[15]

Η διαδικασία συσταδοποίησης με τη χρήση του αλγορίθμου OPTICS γίνεται ως εξής: Αρχικά, ο χρήστης πρέπει να δώσει τιμές στις παραμέτρους min_cluster_size και min_samples. Ο αλγόριθμος με τη σειρά του επιλέγει τυχαία ένα σημείο x για να ξεκινήσει την διαδικασία και υπολογίζει την Ευκλείδεια απόσταση του από κάθε άλλο σημείο που βρίσκεται εντός του χώρου των δεδομένων. Οι αποστάσεις που προκύπτουν, μπαίνουν σε αύξουσα σειρά. Με βάση τις τιμές αυτές, επιλέγονται τα σημεία που βρίσκονται πιο κοντά στο σημείο x που ελέγχεται, και παραμένουν μόνο εκείνα που πληρούν το κριτήριο του min_samples. Έπειτα, ως απόσταση πυρήνα (Core Distance), ορίζεται η απόσταση του x προς τη μεγαλύτερη τιμή από τις αποστάσεις των κοντινών σημείων που εντοπίστηκαν για τη γειτονιά του και το σημείο x ορίζεται ως core point. Ένα σημείο ονομάζεται ως core point εφόσον σε απόσταση πυρήνα περιέχει τουλάχιστον τόσα σημεία όσα ορίζει η παράμετρος min_samples. Εφόσον το σημείο αποτελεί πλέον σημείο πυρήνα, τότε για κάθε γείτονα του υπολογίζεται η απόσταση προσβασιμότητας του. Ως Reachability distance ορίζεται η μέγιστη απόσταση που χρειάζεται το κάθε γειτονικό σημείο για να φτάσει στο x core point λαμβάνοντας υπόψη την core distance του x αλλά και την πραγματική απόσταση μεταξύ των σημείων που ελέγχονται κάθε φορά. Αφού υπολογιστούν όλες οι αποστάσεις προσβασιμότητας της περιοχής, τοποθετούνται σε μια ουρά προτεραιότητας με αύξουσα σειρά και πάλι, και ως επόμενο σημείο για έλεγχο πηγαίνει αυτό με την μικρότερη τιμή. Εφόσον το νέο σημείο είναι και αυτό με τη σειρά του core point, τότε η διαδικασία συνεχίζεται υπολογίζοντας την core distance και τις αποστάσεις προσβασιμότητας του νέου αυτού σημείου και η ουρά ανανεώνεται. Οι νέες τιμές που προστίθενται ταξινομούνται και πάλι και σημεία που βρέθηκαν κοινά στις δύο γειτονιές, δηλαδή σημεία που προϋπήρχαν από πριν στην περιοχή του x υπάρχουν και στην νέα αναζήτηση, τότε αν η νέα τιμή τους είναι μικρότερη αντικαθιστάται αλλιώς παραμένει ως έχει. Η ουρά προτεραιότητας έχει ως στόχο κάθε φορά να εξετάζεται το σημείο με τη μικρότερη reachability distance. Η διαδικασία επαναλαμβάνεται έως ότου εξεταστούν όλα τα σημεία του χώρου. Ωστόσο, στην αντίθετη περίπτωση που το σημείο δεν είναι core point τότε ελέγχεται αν είναι border point για να ενταχθεί σε μια συστάδα ή ορίζεται ως θόρυβος. Παράλληλα, σημεία που έχουν ήδη εξεταστεί αγνοούνται από τον αλγόριθμο ακόμη και αν βρεθούν ως γείτονες κάποιου άλλου σημείου. Όταν πλέον πάψουν να υπάρχουν σημεία προς εξέταση ο αλγόριθμος δημιουργεί ένα διάγραμμα προσβασιμότητας (Reachability plot), το οποίο περιέχει όλα τα σημεία του χώρου με την σειρά που εξετάστηκαν και την αντίστοιχη απόσταση προσβασιμότητας για το κάθε ένα. Μια ακολουθία χαμηλών τιμών Reachability distance θεωρείται ως συστάδα ενώ όταν υπάρχει κενό ή αύξηση της τιμής στο διάγραμμα τότε ο αλγόριθμος λόγω της μεγάλης αυτής αλλαγής θεωρεί ότι εκεί η συστάδα είτε σταματάει είτε βρίσκει το όριο και την έναρξη μιας νέας ομάδας στοιχείων.

Σε σύγκριση με άλλους αλγορίθμους συσταδοποίησης που βασίζονται στην πυκνότητα, ο OPTICS διαθέτει ισχυρούς παράγοντες επιλογής. Ο τρόπος με τον οποίο λειτουργεί και επεξεργάζεται τις συστάδες του παρέχει μεγάλη προσαρμοστικότητα σε διάφορα σύνολα δεδομένων ακόμα και αν αυτά δεν έχουν συγκεκριμένη μορφή. Ιδιαίτερα χρήσιμο εργαλείο είναι εξίσου και το διάγραμμα προσβασιμότητας που χρησιμοποίει (Reachability plot) για να απεικονίσει την εξέλιξη των δεδομένων. Μέσω αυτού αλλά και των αποστάσεων προσβασιμότητας που εντοπίζονται γίνεται ευκολότερη η αναγνώριση των σημείων θορύβου ή των ακραίων τιμών.

Αν και αρκετά ισχυρός αλγόριθμος, ο OPTICS διαθέτει ορισμένα αρνητικά στοιχεία που τον χρήζουν λιγότερο πρακτικό. Το υψηλό υπολογιστικό κόστος που χρειάζεται για να μετρήσει όλες τις αποστάσεις (Ευκλείδειες, Πυρήνα και Προσβασιμότητας) ειδικότερα αν τα δεδομένα είναι πολλά και αρκετά πυκνά μεταξύ τους ίσως είναι και το σημαντικότερο αυτών. Επιπλέον, αν και μπορεί να λειτουργήσει σε διάφορα σύνολα δεδομένων όταν αναφερόμαστε σε μεγάλους όγκους ο χρήστης που θα το διαχειριστεί, θα δαπανήσει απεριόριστο χρόνο σε δοκιμές και διαφορετικούς συνδυασμούς μέχρι να βρει τις κατάλληλες τιμές για τις παραμέτρους, αν αποφασίσει να το κάνει χειροκίνητα. Ακόμη, αν και το διάγραμμα προσβασιμότητας είναι αρκετά βοηθητικό, αν ο χρήστης δεν διαθέτει την γνώση που χρειάζεται η ανάλυση του καθίσταται δύσκολη και μπορεί να δοθεί λάθος ερμηνεία.[14]

2.1.5 HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)

Έπειτα από αρκετά χρόνια και αρκετές μελέτες, το 2013 αποφασίστηκε να δημιουργηθεί ένας επιπλέον αλγόριθμος που θα χρησιμοποιεί τη λογική της συσταδοποίησης των δεδομένων βάσει της πυκνότητας γύρω τους και θα επεκτείνει τη λογική του DBSCAN συνδυάζοντας σημεία που προέρχονται από τις μεθόδους της Ιεραρχικής συσταδοποίησης. Ο αλγόριθμος αυτός ονομάστηκε HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise).

Ο αλγόριθμος HDBSCAN, λαμβάνοντας υπόψη την έννοια της γειτνίασης για τον εντοπισμό των πυκνών περιοχών, αλλά και σε συνδυασμό με την ιεραρχική προσέγγιση που προσφέρει, καταφέρνει να δημιουργήσει μια νέα δομή στο dataset με την μορφή ενός δένδρου, το οποίο δείχνει τις σχέσεις μεταξύ των σημείων κατά τη διάρκεια της συσταδοποίησης αλλά και τα διαφορετικά επίπεδα κόμβων που δημιουργούνται. Αντί όμως να χρησιμοποιεί μια σταθερή τιμή ακτίνας Eps όπως ο DBSCAN, για να εντοπίσει τις πυκνές περιοχές, προσθέτει μια νέα έννοια που αφορά τις αποστάσεις αμοιβαίας προσβασιμότητας (Mutual Reachability distance) και υπολογίζουν πέρα από την μικρότερη απόσταση που πρέπει να έχουν δύο σημεία για να ανήκουν στην ίδια συστάδα και την ίδια την τοπική τους πυκνότητα. Παράλληλα, ο HDBSCAN χρησιμοποιεί δύο παραμέτρους την min_cluster_size, η οποία εξασφαλίζει την εγκυρότητα των συστάδων που ανιχνεύονται και η τιμή της δίνεται χειροκίνητα από τον χρήστη όπως και στον OPTICS αλλά και την έννοια του MinPts που ορίζει ο DBSCAN, χωρίς να είναι απαραίτητος ο καθορισμός της τιμής του από τον χρήστη.[16]

Η διαδικασία συσταδοποίησης με τη χρήση του αλγορίθμου HDBSCAN γίνεται ως εξής: Αρχικά ο χρήστης πρέπει να δώσει τιμή στην παράμετρο min_cluster_size. Ομοίως μπορεί να δοθεί τιμή και στην παράμετρο min_samples. Αν δε δοθεί χειροκίνητα, τότε η παράμετρος παίρνει αυτόματα την ίδια τιμή με την παράμετρο min_cluster_size. Ο αλγόριθμος ξεκινάει υπολογίζοντας τις Ευκλείδειες αποστάσεις κάθε πιθανού ζεύγους σημείων που μπορεί να υπάρξουν στο χώρο. Για κάθε σημείο λαμβάνονται υπόψη μόνο οι τιμές εκείνες που προέκυψαν και η απόσταση τους είναι χαμηλή προς αυτό, άρα βρίσκονται και πιο κοντά του. Από αυτές κρατάει τόσες, όσες ορίζει η παράμετρος min_samples. Η απόσταση με τη μεγαλύτερη τιμή από αυτές σε κάθε περιοχή που ελέγχεται, ονομάζεται ως απόσταση πυρήνα (Core distance) για το σημείο που εξετάζεται όπως ορίστηκε και στον αλγόριθμο OPTICS. Έπειτα, ο HDBSCAN για κάθε ζεύγος σημείων υπολογίζει την απόσταση αμοιβαίας προσβασιμότητας. Συγκεκριμένα, η Mutual reachability distance βρίσκει την μέγιστη απόσταση που χρειάζονται δύο σημεία για να ενωθούν λαμβάνοντας υπόψη τις αποστάσεις πυρήνα που εντοπίστηκαν για κάθε ένα από αυτά αλλά και την Ευκλείδεια απόσταση μεταξύ τους. Στη συνέχεια, για την ανίχνευση των συστάδων ο αλγόριθμος θα ενώσει όλες τις αποστάσεις αμοιβαίας προσβασιμότητας, δημιουργώντας ένα δένδρο ελάχιστης συνδεσιμότητας (Minimum Spanning Tree, MST). Η σύνδεση ξεκινάει από ένα τυχαίο σημείο προς το σημείο με τι μικρότερη αμοιβαία προσβασιμότητα προς αυτό και οι κόμβοι ενώνονται μεταξύ τους κάθε φορά με το μικρότερο δυνατό κόστος, δηλαδή την μικρότερη τιμή απόστασης που βρέθηκε για αυτά. Η διαδικασία συνεχίζει με αυτόν τον τρόπο ενώνοντας όλα τα σημεία του χώρου χωρίς να επαναλαμβάνει διαδρομές και πάντα με την μικρότερη τιμή αμοιβαίας προσβασιμότητας που διαθέτει το καθένα. Το δένδρο ολοκληρώνεται όταν πλέον δεν υπάρχουν άλλα σημεία προς σύνδεση. Στη συνέχεια, για την εύρεση των συστάδων, ο HDBSCAN αφαιρεί τις τιμές προσβασιμότητας που θεωρεί μεγάλες. Οι ακμές αυτές που θα αποχωρήσουν από το δένδρο ορίζονται είτε βάσει ενός ορίου που θα αποφασίσει από ποια τιμή και πάνω δεν είναι αποδεκτή η απόσταση είτε σύμφωνα με την δομή και την πυκνότητα των δεδομένων. Με αυτόν τον τρόπο τα ενωμένα σημεία που παραμένουν αντικατοπτρίζουν πιο πυκνές περιοχές και μετέπειτα συστάδες. Τέλος, για να θεωρηθεί μια συστάδα ως έγκυρη θα πρέπει να έχει σημεία τόσα ή περισσότερα από όσα ορίζει η παράμετρος min_cluster_size ενώ σημεία που δεν ενώνονται με κάποια ακμή ή δεν πληρούν το κριτήριο της έγκυρης συστάδας ορίζονται ως θόρυβος.

Σε πιο μεγάλα σύνολα δεδομένων με πολύπλοκη δομή και αρκετά σημεία θορύβου, που απαιτούν ακρίβεια, τα θετικά στοιχεία που προσφέρει ο αλγόριθμος HDBSCAN αποτελούν την πιο ιδανική λύση για την εύρεση των συστάδων. Αν και επεκτείνει την έννοια της γειτονιάς, ο HDBSCAN δεν

χρειάζεται ένα προκαθορισμένο εύρος για την αναζήτηση και τον εντοπισμό ομάδων αφού δημιουργούνται αυτόματα από το δένδρο που παράγει, γεγονός που τον χρήζει αρκετά ευέλικτο. Επιπλέον, δεν είναι απαραίτητο να ενταχθούν όλα τα σημεία του χώρου σε κάποια συστάδα, ιδίως αν δεν ικανοποιούν τα κριτήρια του min_samples και min_cluster_size, δίνοντας του έτσι την δυνατότητα να εντοπίζει πιο εύκολα τα σημεία θορύβου. Ακόμη, η δενδρική δομή που χρησιμοποιεί, προσφέρει μια πιο πλήρης εικόνα των δεδομένων και την κατάσταση της συσταδοποίησης.

Ωστόσο, υπόψιν πρέπει να ληφθούν και τα αρνητικά στοιχεία του HDBSCAN πριν την επιλογή του. Όπως και οι υπόλοιπο αλγόριθμοι, παρουσιάζει μεγάλη ευαισθησία στις παραμέτρους και πρέπει να δίνονται πάντα με σαφήνεια και βάση το πρόβλημα που διαχειρίζεται ο χρήστης κάθε φορά. Μία μη σωστή ρύθμιση των παραμέτρων min_samples και min_cluster_size μπορεί εύκολα να οδηγήσει σε πολλές μικρές μη σημαντικές συστάδες ή το ανάποδο. Παρόλο που μπορεί να προσαρμοστεί πολύ πιο εύκολα σε κάθε ανάλυση, ο χρόνος που χρειάζεται για να υπολογίσει όλες τι απαιτούμενες αποστάσεις αλλά και για την κατασκευή του δένδρου αργότερα είναι αρκετά απαιτητικός, γεγονός που όταν αυξάνεται ο όγκος των πληροφοριών τον καθιστούν πιο αργό ενώ υψηλή είναι και η απαίτηση της μνήμης που χρειάζεται ώστε να αποθηκεύσει όλα αυτά τα αποτελέσματα.[17]

2.2 Σύγκριση αλγορίθμων

Αξίζει να συγκρίνουμε τους πέντε αλγορίθμους – DBSCAN, DENCLUE, Mean Shift, OPTICS, και HDBSCAN - λαμβάνοντας πάντα υπόψη τον τρόπο που λειτουργούν αλλά και τα χαρακτηριστικά που προσθέτει καθένας από αυτούς στη συσταδοποίηση. Αναλύοντας τους έναν προς έναν, γίνεται εύκολα αντιληπτό ότι καθένας προτείνει μια διαφορετική προσέγγιση για το διαχωρισμό των δεδομένων σε ομάδες αν και όλοι στηρίζονται στην ιδέα της πυκνότητας που δημιουργούν στο χώρο.

Προεκτείνοντας την σκέψη μου, ως ορισμένες διαφορές που καθορίζουν κάθε αλγόριθμο θα μπορούσαν να θεωρηθούν οι εξής: Ο DBSCAN είναι ένας αλγόριθμος που ανιχνεύει συστάδες υποθέτοντας πως η πυκνότητα των σημείων παραμένει ομοιόμορφη και χρησιμοποιεί την ακτίνα Eps για την αναζήτηση ενώ οι υπόλοιποι αλγόριθμοι προσαρμόζουν με τέτοιο τρόπο τη λειτουργία τους, με αποτέλεσμα να μπορούν να διαχειρίζονται με μεγαλύτερη ευκολία τα δεδομένα που τους δίνονται ακόμα και αν η πυκνότητα τους μεταβάλλεται ανά τμήμα. Παρόλο που οι υπόλοιποι τέσσερις διαθέτουν την ικανότητα αυτή, και πάλι διαφέρουν ως προς τον τρόπο που πραγματοποιούν τον διαχωρισμό των δεδομένων. Ο αλγόριθμος DENCLUE όπως και ο Mean Shift χρησιμοποιούν μαθηματικά μοντέλα για τον εντοπισμό την πυκνότητας των σημείων ενώ ο OPTICS έχει την δική του λογική ταξινομώντας τις πυκνότητες. Από την άλλη, και ο HDBSCAN παρά του ότι αποτελεί επέκταση του DBSCAN, χρησιμοποιεί μια ιεραρχική προσέγγιση στα δεδομένα για να το επιτύχει αυτό. Επιπλέον, ενώ ο DBSCAN αντιμετωπίζει προβλήματα όταν υπάρχει αλλαγή στην πυκνότητα από περιοχή σε περιοχή, χαρακτηρίζεται ως ο πιο απλός και γρήγορος από τους υπόλοιπους. Ωστόσο, ακριβώς επειδή για τον αλγόριθμο αυτό η συγκέντρωση σημείων θεωρείται πιο ομοιόμορφη τον κάνει λιγότερο αποδοτικό σε διαφορετικά σχήματα συστάδων και έτσι αλγόριθμοι με μεγαλύτερη πολυπλοκότητα όπως είναι οι υπόλοιποι, θεωρούνται πιο κατάλληλοι στην αντιμετώπιση τέτοιων προβλημάτων.

Από μία άλλη οπτική όμως και οι πέντε αλγόριθμοι μοιράζονται κάποια κοινά χαρακτηριστικά. Αν δεν γίνει σωστή ρύθμιση των παραμέτρων που προτείνει ο καθένας, κανένας από αυτούς δεν θα λειτουργήσει ορθά και δε θα φέρει τα επιθυμητά αποτελέσματα. Επιπρόσθετα, παρά τις διαφοροποιήσεις που έχουν στον τρόπο με τον οποίο διαχειρίζονται τα δεδομένα για να πραγματοποιήσουν τη συσταδοποίηση, κατά κύριο λόγο, κανένας από αυτούς δεν θεωρείται ικανοποιητικός στην αντιμετώπιση δεδομένων που εμφανίζονται με πολλές διαστάσεις, αφού η βάση τους είναι η πυκνότητα των δεδομένων και σε τέτοιες περιπτώσεις είναι δύσκολο να αναγνωριστεί. Ακόμη, η υπολογιστική ισχύ σε χρόνο και πόρους που απαιτεί κάθε ανάλυση για όλους τους υπολογισμούς που χρειάζεται κάθε αλγόριθμος ξεχωριστά, ειδικά όταν έρχονται αντιμέτωποι με μεγάλα σύνολα δεδομένων που απαιτούν εκτεταμένη αναζήτηση και πολλούς υπολογισμούς και αναλύσεις, κρίνεται ιδιαίτερα υψηλή.

Κατά συνέπεια, ως κύριο συμπέρασμα από την παραπάνω σύγκριση καταλήγουμε ότι η επιλογή του σωστού για κάθε περίσταση αλγορίθμου εξαρτάται αποκλειστικά από την φύση των δεδομένων αλλά και από τις απαιτήσεις που έχει κάθε εφαρμογή ή ο εκάστοτε χρήστης.

Κεφάλαιο 3ο: Υλοποίηση αλγορίθμων στην Python

3.1 Εισαγωγή στη Scikit-learn

Ως μέρος ενός project ξεκίνησε η δημιουργία της Scikit-learn[19] που σήμερα αποτελεί μία από τις πιο διαδεδομένες βιβλιοθήκες στο χώρο της μηχανικής μάθησης και η οποία στηρίζεται στην γλώσσα προγραμματισμού Python. Ο λόγος που οδήγησε την επιστήμη στην ανάπτυξη της, ήταν η ανάγκη δημιουργίας μιας ισχυρής βιβλιοθήκης που θα παρέχει τα απαραίτητα εργαλεία στους ανθρώπους που θέλουν να ασχοληθούν και να εφαρμόσουν αλγορίθμους μηχανικής μάθησης χωρίς να χρειαστεί να αναπτύξουν κώδικα από το μηδέν. Η δημιουργία της πρόσφερε μεγάλη βοήθεια σε χρήστες που δεν διέθεταν την γνώση που χρειάζεται πάνω σε περίπλοκες τεχνικές και προγραμματιστικές λεπτομέρειες.

Από τότε μέχρι και σήμερα, για να μπορέσει ο κάθε χρήστης να επικοινωνήσει με τη βιβλιοθήκη, χρησιμοποιεί μια διεπαφή (Application Programming Interface, API) η οποία επιτρέπει την αλληλεπίδραση μεταξύ τους μέσω ορισμένων εντολών και μεθόδων. Η Scikit-learn μπορεί να χαρακτηριστεί επίσης ως λογισμικό ανοιχτού κώδικα (open-source) αφού επιτρέπει την πρόσβαση σε όλους, προγραμματιστές και μη, φοιτητές, ερευνητές ακόμη και σε εταιρίες, δίνοντας τους την δυνατότητα να μπορούν να τροποποιούν τον κώδικα που υπάρχει μέσα σε αυτήν, να επιδιορθώνουν τυχόν σφάλματα που μπορεί να προκύψουν αλλά και την ευχέρεια να προσθέτουν νέες λειτουργίες για την βελτίωση και την μετέπειτα ανάπτυξη της βιβλιοθήκης.

Υπάρχουν πολλοί λόγοι σήμερα, που ξεχωρίζουν την βιβλιοθήκη Scikit-learn και η απλότητα στη χρήση της είναι μόνο ένας από αυτούς και τις δυνατότητες που προσφέρει στη μηχανική μάθηση. Η Scikit-learn περιλαμβάνει ένα πλήθος τεχνικών που είναι χρήσιμοι στην ανάλυση των δεδομένων, όπως είναι η Ταξινόμηση (Classification), η οποία χωρίζει τα δεδομένα σε κατηγορίες, η Παλινδρόμηση (Regression), η οποία χρησιμοποιείται για την πρόβλεψη αριθμών αλλά και η Συσταδοποίηση (Clustering) με την οποία ασχολούμαστε στην παρούσα εργασία, η οποία ομαδοποιεί τα όμοια δεδομένα μεταξύ τους. Παρέχει επίσης μια ποικιλία μεθόδων, όπως η κανονικοποίηση των δεδομένων (Normalization) και η κλιμάκωση (Scaling), που επεξεργάζονται τα δεδομένα για να τα μετατρέψουν σε πιο κατάλληλα για χρήση. Καταφέρνει επίσης μέσω των αλγορίθμων που διαθέτει να δημιουργεί μοντέλα, να τα εκπαιδεύει σε διάφορα σύνολα δεδομένων που μπορεί να της διατεθούν και να βρίσκει σχέσεις μεταξύ τους ώστε να μπορεί αργότερα να προβλέπει και να παίρνει αποφάσεις σε νέα δεδομένα μέσα από αυτά. Ακόμη, υπάρχουν μετρικές αξιολογήσεις και εργαλεία όπως η διασταυρούμενη επικύρωση (Cross Validation) που εξετάζουν την απόδοση τέτοιων μοντέλων και εντοπίζουν πόσο ακριβείς ή σωστές είναι οι προβλέψεις που κάνουν. Τέλος, επιτρέπει την παράλληλη συνεργασία με άλλες βιβλιοθήκες όπως είναι η NumPy και η Pandas, οι οποίες βοηθούν στη δημιουργία πινάκων, λιστών και data frame για την καλύτερη διαχείριση των δεδομένων, η Matplotlib και η Seaborn οι οποίες χρησιμοποιούνται για να απεικονίσουν τα δεδομένα σε σχεδιαγράμματα αλλά και πολλές ακόμα.

Αξίζει να σημειωθεί ωστόσο ότι, παρόλο που η Scikit-learn αποτελεί μια ιδιαίτερα ευέλικτη και ευρέως γνωστή βιβλιοθήκη στην μηχανική μάθηση, υπάρχουν ορισμένοι περιορισμοί στη χρήση της που την καθιστούν ακατάλληλη σε κάποιες περιπτώσεις. Ακριβώς επειδή η Scikit-learn χρησιμοποιεί τον επεξεργαστή (Central Processing Unit, CPU) για την εκτέλεση των διαφόρων εργασιών της και δεν είναι σχεδιασμένη να αξιοποιεί την κάρτα γραφικών (Graphics Processing Unit, GPU) η οποία έχει την ικανότητα να εκτελεί πολλές ενέργειες ταυτόχρονα, όταν έρχεται αντιμέτωπη με πολύπλοκα προβλήματα, η αποδοτικότητα της μειώνεται, ειδικότερα σε σχέση με άλλες βιβλιοθήκες που έχουν τη δυνατότητα χρήσης της GPU. Επιπλέον, επειδή τα δεδομένα τα οποία διαχειρίζεται η βιβλιοθήκη φορτώνονται στην μνήμη (Random Access Memory, RAM), όταν δουλεύει με πολύ μεγάλα σύνολα υπάρχει πάντα ο κίνδυνος η χωρητικότητα της μνήμης να γεμίσει ή να μην είναι αρκετή ώστε να γίνουν οι απαραίτητες διαδικασίες και το σύστημα να βρεθεί σε θέση να μην μπορεί να ανταπεξέλθει, να υπάρχουν μεγάλες καθυστερήσεις ή ακόμη και να παγώσει και κατά συνέπεια να μην ολοκληρώσει τους υπολογισμούς που χρειάζονται. Ακόμη, η Scikit-learn είναι σχεδιασμένη για απλούς αλγορίθμους και δεν διαθέτει ειδικά εργαλεία ή την υπολογιστική ισχύ που χρειάζεται για να

χρησιμοποιήσει της τεχνική της βαθιάς μάθησης (Deep learning), και έτσι θεωρείται ακατάλληλη για να λύσει σύνθετα προβλήματα που μόνο νευρωνικά δίκτυα θα μπορούσαν να επιλύσουν. Τέλος, παρόλο που υποστηρίζει κάποιας μορφή παράλληλης επεξεργασίας, δεν είναι πλήρως εκπαιδευμένη πάνω σε αυτό, με αποτέλεσμα συχνά να υπάρχουν μεγάλες καθυστερήσεις σε διαδικασίες και υπολογισμούς.[18]

Εστιάζοντας στην συσταδοποίηση που είναι το κέντρο της εργασίας, μπορεί να υποστηριχθεί ότι η βιβλιοθήκη Scikit-learn διευκολύνει τη διαδικασία αυτή παρέχοντας μια πλατφόρμα με έτοιμους αλγορίθμους καθώς και διάφορα εργαλεία που βοηθούν στην επεξεργασία των δεδομένων ώστε η ανάλυση να γίνει πολύ πιο εύκολη. Έτσι η διαδικασία της συσταδοποίησης επιτυγχάνεται απλώς με μερικά βήματα. Στην αρχή το σύστημα προ επεξεργάζεται τα δεδομένα ώστε να αποκτήσουν μια κοινή μορφή και λαμβάνεται ο κατάλληλος αλγόριθμος σύμφωνα με τη δομή του προβλήματος ή από την επιλογή που θα κάνει ο εκάστοτε χρήστης. Ο αλγόριθμος παίρνει τα δεδομένα του dataset που θα του δοθεί, τα εκπαιδεύει ώστε να εντοπίσει τις σχέσεις που υπάρχουν μεταξύ τους σε χαρακτηριστικά, σε μοτίβα που μπορεί να δημιουργούν ή οτιδήποτε άλλο μπορεί να θεωρήσει ως κοινό στοιχείο μεταξύ τους, τα κατηγοριοποιεί σύμφωνα με το μέτρο ομοιότητας που επιλέγει και τα τοποθετεί στις αντίστοιχες ομάδες. Με τον τρόπο αυτόν και τη χρήση μόνο συγκεκριμένων ενεργειών δίνεται στο χρήστη η ικανότητα να μπορεί να κατανοήσει και να ερμηνεύσει τα αποτελέσματα που παράγονται χωρίς να χρειάζεται να εμβαθύνει σε περίπλοκες τεχνικές λεπτομέρειες. Επιπλέον, εάν είναι επιθυμητό, για να διασφαλισθεί η ακρίβεια και η αξιοπιστία του αλγορίθμου που χρησιμοποιείται, η Scikit-learn παρέχει όπως έχει ήδη αναφερθεί ορισμένες μετρικές αξιολόγησης που εκτιμούν την απόδοση του.[19]

3.2 Εκτέλεση των αλγορίθμων συσταδοποίησης στην γλώσσα Python

Παρακάτω περιγράφεται η εκτέλεση των αλγορίθμων συσταδοποίησης βάσει πυκνότητας που παρουσιάστηκαν βιβλιογραφικά στο προηγούμενο κεφάλαιο. Η ανάλυση επικεντρώνεται στον τρόπο εκτέλεσης των αλγορίθμων DBSCAN, DENCLUE, Mean Shift, OPTICS και HDBSCAN με τη χρήση ή μη της βιβλιοθήκης Scikit-learn. Λόγω της απαίτησης της βιβλιοθήκης αλλά και για μεγαλύτερη ευκολία οι κώδικες είναι γραμμένοι όλοι στην γλώσσα προγραμματισμού Python ενώ η υλοποίηση τους πραγματοποιήθηκε στο περιβάλλον Spyder, το οποίο πρόσφερε τα εργαλεία για την εκτέλεση του κώδικα.

3.2.1 Εκτέλεση DBSCAN

Ο αλγόριθμος DBSCAN όπως ορίστηκε στο Κεφάλαιο 2ο, εντοπίζει τις συστάδες βασισμένος στις γειτονιές των σημείων και θεωρώντας πως η συγκέντρωση τους είναι όμοια σε όλες τις περιοχές. Η υλοποίηση του παρέχεται από την βιβλιοθήκη Scikit-learn, γεγονός που επιτρέπει στον χρήστη να εφαρμόσει τον αλγόριθμο εύκολα και γρήγορα. Ωστόσο, τα βήματα που χρειάζονται για την εκτέλεση του δίνονται παρακάτω.

Η πρώτη ενέργεια που εντοπίζει κανείς και είναι απαραίτητη σε κάθε αλγόριθμο είναι ο προσδιορισμός και η εισαγωγή των βιβλιοθηκών που θα χρειαστούν παρακάτω. Είθισται για μεγαλύτερη ευκολία και καλύτερη οργάνωση να δηλώνονται όλα στην αρχή του κώδικα. Η ελλιπής ρύθμιση, μπορεί να οδηγήσει σε σφάλματα ενώ ανά πάσα στιγμή κατά τη συγγραφή του κώδικα μπορεί εύκολα να προστεθεί οποιαδήποτε νέα βιβλιοθήκη είναι απαραίτητη.

```
8   from sklearn.cluster import DBSCAN
9   import pandas as pd
10  import numpy as np
11  import seaborn as sns
12  import matplotlib.pyplot as plt
13  from pandas.plotting import parallel_coordinates
14  from sklearn.metrics import silhouette_score
15  from sklearn.neighbors import NearestNeighbors
```

Εικόνα 3.1 Στιγμιότυπο από τον αλγόριθμο DBSCAN στο Spyder

Αρχίζοντας από την γραμμή κώδικα 8 *from sklearn.cluster import DBSCAN* αποτελεί ίσως την κυριότερη εντολή που είναι απαραίτητη για την εισαγωγή κάθε αλγορίθμου που χρησιμοποιεί την

βιβλιοθήκη Scikit-learn στο σύστημα, με παράδειγμα αυτή την στιγμή τον DBSCAN. Η λέξη *sklearn* αναφέρεται στην ίδια τη βιβλιοθήκη και διατυπώνοντας την εντολή με αυτό τον τρόπο, το σύστημα αναζητά από την αντίστοιχη ενότητα *cluster* που την περιέχει να την εισάγει. Έτσι, ενεργοποιείται η δυνατότητα χρήσης της έτοιμης δομής στον κώδικα χωρίς να χρειάζεται η υλοποίηση της από την αρχή. Επιπλέον, η έννοια της γειτνίασης που χρησιμοποιεί ο DBSCAN, εισάγεται μέσω της ενότητας *neighbors* από την κλάση *NearestNeighbors* όπως φαίνεται στη γραμμή κώδικα 15. Η μέθοδος αυτή, θα επιτρέψει στον αλγόριθμο να εντοπίσει τα γειτονικά σημεία που είναι απαραίτητα κάθε φορά.

Με την γραμμή κώδικα 9 *import pandas as pd* εισάγουμε την βιβλιοθήκη Pandas, η οποία χρησιμοποιείται για την επεξεργασία και την ανάλυση των δεδομένων σε διάφορες μορφές όπως έχει οριστεί και είναι χρήσιμη για τον αλγόριθμο. Οι υπόλοιπες γραμμές κώδικα, αφορούν διαφορετικά σχεδιαγράμματα που χρησιμοποιούνται για να απεικονίσουν τα αποτελέσματα που θα παράγει ο αλγόριθμος καθώς και την εισαγωγή της μετρικής αξιολόγηση Silhouette score που θα εκτιμήσει την απόδοση του αργότερα.

```
18    x = pd.read_csv('x.csv')
19
20    data = x.iloc[:, 0:5]
21
22    neighb = NearestNeighbors(n_neighbors=5)
23    nbrs = neighb.fit(data)
24    distances, indices = nbrs.kneighbors(data)
```

Εικόνα 3.2 Στιγμιότυπο από τον αλγόριθμο DBSCAN στο Spyder

Έπειτα, το επόμενο απαραίτητο βήμα είναι ο καθορισμός του dataset πάνω στο οποίο θα δουλέψει ο αλγόριθμος. Με την εντολή της γραμμής 18 *x = pd.read_csv ('x.csv')*, το σύστημα λαμβάνει και διαβάζει το περιεχόμενο ενός αρχείου x που του δίνεται. Στην προκειμένη περίπτωση το αρχείο είναι της μορφής κειμένου (Comma-Separated Values, CSV) αλλά θα μπορούσε να δοθεί οποιοσδήποτε τύπος αρχείου αρκεί τα δεδομένα να διαμορφωθούν σε μια πιο κατανοητή για τον αλγόριθμο μορφή ώστε να μπορέσει να τα διαχειριστεί. Αυτό επιτυγχάνεται με τη χρήση της βιβλιοθήκης Pandas, η οποία μετατρέπει τα δεδομένα σε data frame ώστε να είναι πιο εύκολα για χρήση. Το data frame είναι μία δομή δεδομένων που μοιάζει στον τρόπο της με πίνακα αλλά αρκετά πιο ευέλικτη αφού σε αυτό μπορούν να αποθηκευτούν διαφόρων τύπων δεδομένα.

Στη συνέχεια, στη γραμμή κώδικα 20 *data = x.iloc[:, 0:5]*, η μέθοδος *iloc* χρησιμοποιείται ώστε να μπορέσουμε να επιλέξουμε ένα συγκεκριμένο σύνολο στοιχείων από το data frame που δημιουργήθηκε. Η άνω και κάτω τελεία στη αρχή, υποδηλώνει την επιλογή όλων των γραμμών του πίνακα ενώ το μέρος *0:5* αναφέρεται στις στήλες. Στο παράδειγμα εδώ θα πάρει συνολικά 5, δηλαδή τις στήλες 0 1 2 3 4. Σε περίπτωση που η τιμή αυτή αντί για 5 είναι αρνητική -1, αυτό σημαίνει πως το σύστημα θα λάβει τις πληροφορίες από όλες τις στήλες του data frame εκτός από την τελευταία. Αναλόγως την πλευρά που θα τοποθετηθεί ένας αρνητικός ή θετικός αριθμός θα αφαιρεθούν και οι ανάλογες στήλες ή γραμμές. Στους αλγορίθμους συσταδοποίησης συνήθως παραλείπεται η τελευταία στήλη των data set, καθώς εκεί συχνά περιέχεται κάποια ετικέτα και όχι κάποιο σημαντικό προς αξιολόγηση χαρακτηριστικό των δεδομένων.

Με την εντολή στην γραμμή 22 *neighb = NearestNeighbors(n_neighbors=5)* ο αλγόριθμος DBSCAN μέσω της κλάσης *NearestNeighbors*, αποπειράται να δημιουργήσει ένα αντικείμενο το οποίο θα εντοπίζει για κάθε σημείο στο σύνολο δεδομένων τόσα γειτονικά σημεία όσα οριστούν από τον χρήστη (π.χ. 5) ενώ έπειτα θα πρέπει να εκπαιδεύσει το data frame βάση την απόφασης αντικειμένου αυτής ώστε να εντοπίσει τις σχέσεις μεταξύ τους. Αυτό επιτυγχάνεται με την χρήση της μεθόδου *fit()*. Στη συνέχεια, εντοπίζει τους γείτονες κάθε σημείου και επιστρέφει την απόσταση καθενός (*distances*) από το εξεταζόμενο σημείο αλλά και την αντίστοιχη θέση του (*indices*) στο χώρο.

```
37    dbscan = DBSCAN(eps=5.5, min_samples=5)
38    labels = dbscan.fit_predict(data)
39
40    x['cluster'] = labels
```

Εικόνα 3.3 Στιγμιότυπο από τον αλγόριθμο DBSCAN στο Spyder

Η κλήση του αλγορίθμου DBSCAN πραγματοποιείται μέσω της γραμμής κώδικα 37. Για να καταστεί όμως λειτουργικός ο αλγόριθμος, χρειάζεται η ρύθμιση των απαραίτητων παραμέτρων που τον καθορίζουν, δηλαδή την προσθήκη τιμών στο *Eps* και *min_samples* ή MinPts. Με τον τρόπο αυτό, δημιουργείται ένα νέο αντικείμενο το οποίο θα ανιχνεύει τις συστάδες βάσει της συσταδοποίησης που πραγματοποιεί ο αλγόριθμος DBSCAN.

Κατόπιν, το data frame εκπαιδεύεται και πάλι σύμφωνα όμως με το νέο αντικείμενο και δημιουργεί μια πρόβλεψη για την συστάδα που ανήκει κάθε σημείο του χώρου. Η εκτίμηση αυτή επιστρέφει ως ετικέτα ως ένας αριθμός που μπορεί να είναι θετικός και να ορίζει την συστάδα που ανήκει το σημείο ή αρνητικός -1 αν αποτελεί θόρυβο. Η γραμμή κώδικα 40 *x['cluster'] = labels* χρησιμοποιείται για τη δημιουργία μιας νέας ή την τροποποίηση μιας προ υπάρχουσας στήλης στο data frame με το όνομα *cluster* και στην οποία θα τοποθετηθεί για κάθε σημείο η ετικέτα συστάδας που βρέθηκε νωρίτερα.

```
50    print(x)
51    banana.to_csv('x_clusterAssignmentsDBScan.csv')
```

Εικόνα 3.4 Στιγμιότυπο από τον αλγόριθμο DBSCAN στο Spyder

Τελευταίο βήμα και αν το επιθυμεί ο χρήστης αποτελεί η εκτύπωση του τελικού dataset στο τερματικό και η αποθήκευση του σε ένα νέο αρχείο στον υπολογιστή.

3.2.2 Εκτέλεση DENCLUE

Ο αλγόριθμος DENCLUE αφορά μια μέθοδο συσταδοποίησης η οποία ανιχνεύει τις ομάδες σύμφωνα με την συνεισφορά κάθε σημείου στην περιοχή του. Αν και η βιβλιοθήκη Scikit-learn διαθέτει μια μεγάλη γκάμα αλγορίθμων συσταδοποίησης, ο DENCLUE είναι από τους λίγους που αποτελούν εξαίρεση από αυτήν. Ωστόσο, η υλοποίηση του μπορεί να πραγματοποιηθεί με τη βοήθεια μιας εξωτερικής βιβλιοθήκης της PyClustering που περιέχει με τη σειρά της αλγορίθμους και αποτελεί εξίσου μια βιβλιοθήκη ανοιχτού κώδικα γραμμένη σε κώδικα Python. Η βιβλιοθήκη αυτή συνδυάζεται με την με την Scikit-learn και εκτελούν τον αλγόριθμο μαζί. Διαφορετικά ο χρήστης μπορεί να επιλέξει να δημιουργήσει έναν ειδικά προσαρμοσμένο κώδικα που θα ακολουθεί τα βήματα του αλγορίθμου DENCLUE. Παρακάτω παρουσιάζεται μια μορφή για κάθε περίπτωση:

Σενάριο 1ο

Ξεκινώντας από την πιο απλή μορφή, αυτήν της εξωτερικής βιβλιοθήκης, τα βήματα που πρέπει να ακολουθήσει ο χρήστης είναι συγκεκριμένα. Αρχικά για την ομαλή λειτουργία του DENCLUE, ο χρήστης θα πρέπει να εγκαταστήσει την PyClustering. Η χρήση της είναι απλή και αποτελεί αναγκαίο κομμάτι για να μπορέσουμε να εισάγουμε την έτοιμη δομή του DENCLUE στον κώδικα. Η εγκατάσταση της πραγματοποιείται μέσω ενός τερματικού και με την εισαγωγή της εντολής *pip install pyclustering* σε αυτό.

```
10    from pyclustering.cluster.denclue import denclue
11    from sklearn.preprocessing import StandardScaler
12    import numpy as np
13    import pandas as pd
```

Εικόνα 3.5 Στιγμιότυπο από το 1ο σενάριο του DENCLUE στο Spyder

Ακολούθως, όπως και στον αλγόριθμο DBSCAN, στην αρχή της συγγραφής του κώδικα ορίζονται οι απαραίτητες βιβλιοθήκες που θα χρησιμοποιηθούν εντός αυτού. Η γραμμή κώδικα 10 *from pyclustering.cluster.denclue import denclue* αποτελεί βασική εντολή για την εισαγωγή του αλγορίθμου DENCLUE στον κώδικα με τη χρήση της εξωτερικής βιβλιοθήκης. Ο αλγόριθμος DENCLUE απαιτεί την ομοιόμορφη δομή των δεδομένων για να μπορέσει να τα διαχωρίσει και για να το επιτύχει αυτό αργότερα, προσθέτει την κλάση *StandardScaler* της Scikit-learn. Οι δύο βιβλιοθήκες ήδη φαίνεται να συνυπάρχουν.

Παράλληλα, εντάσσετε στον κώδικα μία νέα βιβλιοθήκη η NumPy (Numerical Python) η οποία προσφέρει διάφορα εργαλεία για τους μαθηματικούς υπολογισμούς που μπορεί να χρειαστούν κατά τη

διάρκεια εκτέλεσης του. Η βιβλιοθήκη αυτή αναφέρεται επίσης στην Εικόνα 3.1 που αφορά στιγμιότυπο του αλγορίθμου DBSCAN αλλά δεν χρησιμοποιείται για την διαχείριση των δεδομένων από τον αλγόριθμο παρά μόνο για την χρήση σχεδιαγραμμάτων.

```
18    x = pd.read_csv('x.csv')
19     data = x.iloc[:, :-1].values
```

Εικόνα 3.6 Στιγμιότυπο από το 1ο σενάριο του DENCLUE στο Spyder

Με την ίδια λογική που ακολουθεί ο DBSCAN, εισάγεται το επιθυμητό σύνολο δεδομένων και μετατρέπεται σε μια πιο εύκολη μορφή προς διαχείριση. Στη συνέχεια, επιλέγονται οι γραμμές και στήλες του data frame που θα χρησιμοποιηθούν για τη συσταδοποίηση και επιστρέφει τα δεδομένα σε μορφή που να μπορεί να την επεξεργαστεί ο DENCLUE. Αυτό επιτυγχάνεται με την ιδιότητα values που προστίθεται στο τέλος.

```
21    scaler = StandardScaler()
22     data_scaled = scaler.fit_transform(data)
```

Εικόνα 3.7 Στιγμιότυπο από το 1ο σενάριο του DENCLUE στο Spyder

Σε αυτή τη φάση, τα δεδομένα κανονικοποιούνται. Αυτό σημαίνει πως τα νέα δεδομένα πλέον θα πάρουν ίδια μορφή και τύπο σε αντίθεση με τον αλγόριθμο DBSCAN που ο τρόπος λειτουργίας του δεν απαιτεί κάτι τέτοιο. Μέσω της γραμμής κώδικα 21 *scaler = StandardScaler()* δημιουργείται ένα αντικείμενο το οποίο θα δίνει το νέο αυτό σχήμα στα δεδομένα. Με την επόμενη οδηγία το σύστημα εκμεταλλεύεται αυτό το αντικείμενο, προσαρμόζει τα δεδομένα πάνω σε αυτό και τα μετατρέπει στην επιθυμεί μορφή. Η κανονικοποίηση θεωρείται αρκετά σημαντική, ιδίως σε αλγορίθμους που βασίζονται σε μαθηματικούς υπολογισμούς για την αποφυγή τυχών σφαλμάτων αλλά και για μεγαλύτερη ακρίβεια.

```
27    denclue = denclue (data_scaled, bandwidth=0.5, threshold=0.5)
28    denclue. process ()
```

Εικόνα 3.8 Στιγμιότυπο από το 1ο σενάριο του DENCLUE στο Spyder

Έπειτα, όπως φαίνεται και στις παραπάνω γραμμές, σειρά έχει η κλήση του αλγορίθμου DENCLUE, δίνοντας συγχρόνως τιμές στις αντίστοιχες παραμέτρους. Οι παράμετροι που λαμβάνει ο αλγόριθμος είναι τα όμοια πλέον δεδομένα, το εύρος της περιοχής μέσα στο οποίο συνεισφέρει κάθε σημείο αλλά και ένα κατώφλι (*threshold*) το οποίο από τη στιγμή που υπάρξει σύγκλιση των σημείων θα ορίσει αν δημιουργηθεί η συστάδα ή όχι. Ειδικότερα, το *threshold* αφορά την ελάχιστη πυκνότητα που θα πρέπει να έχει μια περιοχή ώστε να θεωρηθεί ως συστάδα. Όταν η τιμή της πυκνότητας περιοχής είναι μεγαλύτερη από το κατώφλι τότε η περιοχή αναγνωρίζεται ως πυκνή και δημιουργείται η συστάδα ενώ στην αντίθετη περίπτωση ορίζεται ως θόρυβος. Αμέσως, εκτελείτε η μέθοδος *process()* η οποία και θα εκκινήσει την όλη διαδικασία. Η μέθοδος αυτή είναι απαραίτητη για την περίπτωση του DENCLUE αφού η εκτέλεση του δεν γίνεται αυτόματα από την Scikit-learn αλλά από εξωτερική βιβλιοθήκη.

```
33    clusters = denclue.get_clusters()
34    noise= denclue.get_noise()
```

Εικόνα 3.9 Στιγμιότυπο από το 1ο σενάριο του DENCLUE στο Spyder

Τέλος, η ανάκτηση των συστάδων όπως και των noise points που εντοπίστηκαν στον αλγόριθμο, γίνεται μέσω της χρήσης των μεθόδων *get_cluster()* και *get_noise()* αντίστοιχα που επιστρέφουν τα αποτελέσματα που παράχθηκαν μετά την εκτέλεση του αλγορίθμου. Ο αλγόριθμος ολοκληρώνεται και ο χρήστης μπορεί να επιλέξει αν θέλει να εμφανίσει όλα τα στοιχεία, να δώσει ετικέτες στις κλάσεις ή να πραγματοποιήσει οποιαδήποτε άλλη επιπλέον ενέργεια επιθυμεί όπως στον DBSCAN.

Σενάριο 2ο

Πέρα από την απλή εκτέλεση του αλγορίθμου με τη χρήση της εξωτερικής βιβλιοθήκης, μια άλλη οπτική προτείνει έναν προσαρμοσμένο κώδικα που θα υλοποιείται κατά τη διάρκεια της συγγραφής του, αφού ο χρήστης αποφασίσει ότι θέλει να ομαδοποιήσει τα δεδομένα με τη τεχνική του

DENCLUE. Σε αυτή την περίπτωση όλες οι ενέργειες και τα βήματα που θα πρέπει να εκτελέσει ο αλγόριθμος θα δίνονται χειροκίνητα. Μια τέτοια μορφή υλοποίησης παρουσιάζεται παρακάτω.

Ο κώδικας ξεκινάει με τον προσδιορισμό των απαραίτητων εργαλείων που χρειάζονται για την υλοποίηση του DENCLUE και για να γίνει χρήση τους πρέπει να δηλωθούν στην αρχή. Οι μέθοδοι και οι βιβλιοθήκες αυτές είναι κοινές και για τα δύο σενάρια και εισάγονται με τον ίδιο τρόπο με μοναδική εξαίρεση την προσθήκη της κλάσης *KernelDensity* από την βιβλιοθήκη Scikit-learn η οποία θα επιτρέψει μέσω της γκαουσιανής συνάρτησης τον υπολογισμό της επιρροής πυκνότητας κάθε σημείου. Αυτό επιτυγχάνεται με την εντολή *from sklearn.neighbors import KernelDensity* στην αρχή του κώδικα. Στη συνέχεια, όπως και πριν, εισάγονται τα δεδομένα του data set επιλογής και ξεκινάει η δημιουργία του κώδικα υλοποίησης του DENCLUE.

```
18  def denclue (X, bandwidth=0.5, epsilon=0.01, max_iterations=100):
19      X_scaled = StandardScaler().fit_transform(X)
20      kde = KernelDensity(bandwidth=bandwidth). fit(X_scaled)
21      def gradient_ascent(point):
22          for _ in range(max_iterations):
23              density = np.exp(kde.score_samples([point]))[0]
24              gradient = np.mean(X_scaled - point, axis=0) * density
25              new_point = point + gradient
26              if np.linalg.norm(new_point - point) < epsilon:
27                  break
28              point = new_point
29          return point
30
31
32      attraction_points = np.array([gradient_ascent(point) for point in X_scaled])
33
34   unique_points, labels = np.unique(attraction_points.round(decimals=2), axis=0, return_inverse=True)
35    return labels, attraction_points
```

Εικόνα 3.10 Στιγμιότυπο από το 2ο σενάριο του DENCLUE στο Spyder

Η υλοποίηση της συνάρτησης DENCLUE όπως φαίνεται και στη γραμμή κώδικα 18, περιλαμβάνει ορισμένες προαπαιτούμενες παραμέτρους. Η πρώτη παράμετρος *X* αφορά τα δεδομένα που δόθηκαν για αναζήτηση, το *bandwidth* δηλαδή την περιοχή που επηρεάζει κάθε σημείο, το *epsilon* το οποίο στην προκειμένη περίπτωση θα δώσει το όριο μέχρι που θα φτάσει η μετακίνηση και το *max_iterations* που καθορίζει τον χρόνο εκτέλεσης για να μη τρέχει για πάντα.

Σε αυτό το σενάριο δεν υπάρχει παράμετρος threshold αφού χρησιμοποιούμε εξ ολοκλήρου την βιβλιοθήκη Scikit-learn, με προσθήκη συνάρτησης, και η οποία δεν το διαθέτει. Οπότε ως κατώφλι ορίζεται ένα συγκεκριμένο σύνολο επαναλήψεων μέσω της *max_iterations* το οποίο εξαρτάται άμεσα από τη δομή των δεδομένων, τα χαρακτηριστικά και την κατανομή τους στο χώρο. Για παράδειγμα, ένα μικρό σύνολο δεδομένων θα χρειαστεί λιγότερες επαναλήψεις από ότι ένα μεγαλύτερων διαστάσεων.

Στη συνέχεια τα δεδομένα προσαρμόζονται και κανονικοποιούνται όπως ορίζει η διαδικασία. Έπειτα, για την εύρεση της συνεισφοράς κάθε σημείου και σύμφωνα με την γραμμή κώδικα 20 *kde = KernelDensity (bandwidth = bandwidth) .fit(X_scaled),* ο αλγόριθμος δημιουργεί ένα μοντέλο που με βάση το εύρος *bandwidth* που του έχει δοθεί, θα εκτιμά την επιρροή αυτή.

Συνεχίζοντας, όπως αναφέρθηκε και στο Κεφάλαιο 2ο, ο αλγόριθμος DENCLUE για να δημιουργήσει τις συστάδες απαιτεί την εύρεση κορυφών μέσω μιας αυξητικής πορείας όπου η πυκνότητα εκεί θα συγκεντρώνεται στο μέγιστο. Αυτό επιτυγχάνεται με τη χρήση της μεθόδου *Gradient Ascent*. Για να μη τρέχει η μέθοδος αυτή επ' αόριστον , χρησιμοποιείται ένας βρόχος με επαναλήψεις τόσες όσες ορίστηκαν από το max_iterations. Έτσι, για κάθε σημείο του χώρου σύμφωνα με τη γραμμή κώδικα 23, υπολογίζεται η πυκνότητα του σύμφωνα με τη συνεισφορά που βρέθηκε σε προηγούμενο βήμα. Ο KDE για μεγαλύτερη ακρίβεια προσδιορίζεις τις τιμές αυτές ως λογάριθμους με τη βοήθεια του *score_samples* και έπειτα τις μετατρέπει σε κανονικές τιμές πυκνότητας μέσω της συνάρτησης *exp()*. Το *[0]* στο τέλος της συνάρτησης δίνει τη δυνατότητα, ακόμα και αν το αποτέλεσμα είναι μία μόνο τιμή, να την εξάγει.

Έπειτα, θα πρέπει να βρεθεί η κατεύθυνση στην οποία θα μετακινηθεί ο αλγόριθμος. Αυτό εντοπίζεται με τη γραμμή κώδικα 24. Για να βρεθεί αρχικά η θέση των υπόλοιπων σημείων σε σχέση με το σημείο που ελέγχεται κοιτάζουμε τη διαφορά μεταξύ τους (*x_scaled – point*). Η διαφορά αυτή είναι χρήσιμη ώστε να γνωρίζει ο αλγόριθμος προς τα πού θα κατευθύνει το *point* έπειτα. Ωστόσο, για να μετακινηθεί το σημείο θα πρέπει να βρεθεί μόνο μια τιμή που θα δείχνει τη νέα θέση στην οποία θα τοποθετηθεί ο αλγόριθμος. Για το λόγο αυτό, με τη χρήση της μεθόδου *mean()* και της ιδιότητας *axis=0* η οποία επιτρέπει τον υπολογισμό του x και y των συντεταγμένων των σημείων ξεχωριστά, η gradient ascent βρίσκει μια μέση τιμή από τις διαφορές και την πολλαπλασιάζει με την πυκνότητα του σημείου για να μπορέσει ο αλγόριθμος να μετακινηθεί σε γειτονική θέση που θα αφορά τη μεγαλύτερη πυκνότητα, όπως προτείνει η ίδια η λειτουργία του DENCLUE.

Η μετακίνηση στο επόμενο σημείο πραγματοποιείται με την εντολή της γραμμής κώδικα 25 *new_point = point + gradient*. Το σημείο τότε παίρνει νέες συντεταγμένες βάση της τοποθεσίας που βρισκόταν το εξεταζόμενο σημείο αλλά και της κατεύθυνσης που ορίστηκε από την προηγούμενη εντολή. Για να βεβαιωθεί ο αλγόριθμος ότι το εξεταζόμενο σημείο δεν είναι ήδη ένα τοπικό μέγιστο, χρησιμοποιεί μια συνθήκη ελέγχου. Συγκεκριμένα, ελέγχεται η διαφορά ανάμεσα στο νέο σημείο σε σχέση με το σημείο που εξετάζεται. Η μέθοδος *np.linalg.norm()* χρησιμοποιείται για να μπορέσει ο αλγόριθμος να υπολογίσει τη διαφορά μεταξύ της απόστασης των δύο σημείων. Αν η συνθήκη είναι αληθής και η απόσταση είναι μικρότερη από το όριο που δόθηκε, δηλαδή η αλλαγή είναι αρκετά μικρή ή το σημείο συγκλίνει ως τοπικό μέγιστο τότε ο αλγόριθμος δεν κάνει καμία άλλη κίνηση και επιστρέφει το σημείο. Αν η συνθήκη δεν είναι αληθής, τότε ορίζεται ως νέο σημείο προς εξέταση η θέση που βρέθηκε και η διαδικασία συνεχίζει από την αρχή έως ότου υπάρξει μια νέα κορυφή. Η *gradient_ascent* θα σταματήσει μόλις ολοκληρωθούν όλες οι επαναλήψεις.

Αφού δημιουργηθεί η συνάρτηση εντοπισμού κατεύθυνσης τότε καλείτε για κάθε σημείο του χώρου και δημιουργείται ένας πίνακας με την κορυφή που εντόπισε καθένα από αυτά. Η υλοποίηση του DENCLUE ολοκληρώνεται επιστρέφοντας τα τοπικά μέγιστα, δηλαδή τις τιμές που βρέθηκαν ότι συγκλίνει κάθε σημείο και τις ετικέτες των συστάδων όπου αντιστοιχούν αυτά. Συγκεκριμένα, για μεγαλύτερη ευκολία τα τοπικά μέγιστα στρογγυλοποιούνται σε δύο δεκαδικά ψηφία, αφαιρούνται οι διπλές τιμές που μπορεί να έχουν δημιουργηθεί και καθένα από αυτά πλέον αντιπροσωπεύει μια συστάδα. Κάθε σημείο του χώρου τότε κατηγοριοποιείται σύμφωνα με αυτό.

```
57    labels, attraction_points = denclue (data, bandwidth=0.5, epsilon=0.01, max_iterations=100)
```

Εικόνα 3.11 Στιγμιότυπο από το 2ο σενάριο της κλήσης του DENCLUE στο Spyder

Η κλήση του αλγορίθμου DENCLUE πραγματοποιείται στην μετέπειτα διάρκεια συγγραφής του κώδικα με τη χρήση και τη ρύθμιση των αντίστοιχων παραμέτρων, όπως φαίνεται στην Εικόνα 3.11.

Συγκρίνοντας τα δύο σενάρια, μπορούμε να πούμε ότι υπάρχουν σημαντικές διαφορές μεταξύ αυτών. Στην πρώτη περίπτωση με την εξωτερική βιβλιοθήκη PyClustering, η διαδικασία γίνεται σαφώς πιο εύκολα και γρήγορα αφού προσφέρεται μια έτοιμη δομή και τα βήματα που θα πρέπει να ακολουθήσει ο χρήστης είναι συγκεκριμένα. Επιπλέον, με τη βοήθεια της PyClustering εξοικονομείται περισσότερος χρόνος και ελαχιστοποιείται η περίπτωση να συμβεί κάποιο λάθος. Ωστόσο, με το σενάριο αυτό δεν μπορεί να κατανοηθεί πλήρως ο τρόπος λειτουργίας του αλγορίθμου.

Από την άλλη πλευρά, στην δεύτερη περίπτωση της υλοποίησης του DENCLUE με προσαρμοσμένο κώδικα, ο χρήστης έχει τη δυνατότητα να πειραματιστεί και του δίνεται μεγαλύτερος έλεγχος πάνω στον κώδικα. Όμως, και η προσέγγιση αυτή έχει τα ελαττώματα της αφού αν δεν υπάρχει γνώση γύρω από το αντικείμενο, ο χρήστης θα δυσκολευτεί, η διαδικασία θα γίνει αρκετά χρονοβόρα και ο κίνδυνος να συμβούν σφάλματα θα μεγαλώσει. Σε γενικές γραμμές, δεν ορίζεται συγκεκριμένα ποιο από τα δύο σενάριο είναι το καλύτερο αφού η επιλογή του εξαρτάται από την φύση και τη δομή των δεδομένων αλλά και την επιλογή που θα κάνει ο χρήστης.

3.2.3 Εκτέλεση Mean Shift

Ο αλγόριθμος Mean Shift αποτελεί επίσης μια επαναληπτική διαδικασία μετακίνησης θέσεων προς περιοχές υψηλότερη πυκνότητας, όπως ο DENCLUE, όμως αντί να χρησιμοποιεί την Gradient ascent για τον εντοπισμό της κλίσης των σημείων, μετακινεί τα δεδομένα βάσει του κέντρου βάρους της γειτονιά τους.

Όπως και ο DBSCAN, η υλοποίηση του δίνεται από την βιβλιοθήκη Scikit-learn και τα βήματα που ακολουθεί είναι κοινά. Οι βιβλιοθήκες και οι απαραίτητες μέθοδοι παραμένουν ίδιες και ορίζονται στην αρχή με μοναδική προσθήκη τη συνάρτηση *estimate_bandwidth()* η οποία εκτιμάει το εύρος που επηρεάζει κάθε σημείο σε πυκνότητα. Η έτοιμη δομή, εισάγεται από την βιβλιοθήκη με την γραμμή κώδικα *from sklearn.cluster import MeanShift, estimate_bandwidth* και ο χρήστης συνεχίζει με τα επόμενα βήματα. Τα δεδομένα τότε εισάγονται στο πρόγραμμα και μετατρέπονται σε όμοια.

```
24  bandwidth = estimate_bandwidth(data_scaled, n_samples=151)
25
26  meanshift = MeanShift(bandwidth=bandwidth)
27  labels = meanshift.fit_predict(data_scaled)
28
29  cluster_centers = meanshift.cluster_centers_
```

Εικόνα 3.12 Στιγμιότυπο από τον αλγόριθμο Mean Shift στο Spyder

Στο ενδεχόμενο ο χρήστης να αποφασίσει να μην ορίσει παραμετρικά το bandwidth όπου θα γίνεται η αναζήτηση των σημείων αλλά επιθυμεί να δώσει στο πρόγραμμα τη δυνατότητα να βρει ένα πιο αξιόπιστο εύρος, χρησιμοποιεί τον κώδικα της γραμμής 24 όπου η συνάρτηση *estimate_bandwidth()* καθορίζει με βάση τα όμοια πλέον δεδομένα αλλά και το πλήθος αυτών, μια τιμή για την ακτίνα αυτή. Στην περίπτωση του Mean Shift, η παράμετρος *n_samples* αφορά τα δεδομένα που θα πάνε προς εκτέλεση και μπορεί να πάρει οποιαδήποτε τιμή, όμως μια μικρότερη τιμή θα οδηγήσει μεν σε ταχύτερη εκτίμηση αλλά ανακριβής. Συνήθως επιλέγεται μία τιμή ίση με τα δείγματα του συνόλου δεδομένων. Αμέσως μετά καλείτε ο αλγόριθμος ο οποίος θα αποφασίσει πως θα διαχωρίσει τα σημεία σε συστάδες. Τα δεδομένα προσαρμόζονται πάνω στον Mean Shift και παράγεται η πρόβλεψη της συστάδας που ανήκει καθένα από αυτά. Παράλληλα, επιστρέφονται και τα κέντρα των συστάδων που έχουν εντοπιστεί, δηλαδή οι θέσεις με το μεγαλύτερο κέντρο βάρους σε κάθε περιοχή ομάδα και τα δεδομένα εμφανίζονται.

3.2.4 Εκτέλεση OPTICS και HDBSCAN

Ο αλγόριθμος OPTICS, σύμφωνα με τον τρόπο λειτουργίας που προτείνει, ορίζει τα δεδομένα σε συστάδες βάσει μιας ουράς προτεραιότητας ενώ ο αλγόριθμος HDBSCAN δημιουργεί μια δενδρική δομή στην ανίχνευση τους. Η υλοποίηση των δύο αλγορίθμων, όπως και των DBSCAN και Mean Shift, παρέχονται από την βιβλιοθήκη Scikit-learn και τα βήματα που ακολουθούν είναι κοινά. Ο χρήστης επιλέγει το dataset που θέλει να χρησιμοποιήσει, φορτώνει τα δεδομένα στο σύστημα και τα μετατρέπει σε συγκρίσιμη μορφή. Η κλήση των αλγορίθμων γίνεται ξεχωριστά με τις παρακάτω γραμμές κώδικα κατά την διάρκεια συγγραφής του κώδικα: *optics = OPTICS(min_samples=5, min_cluster_size=7)* και *hdb = HDBSCAN(min_cluster_size=7)* με τη ρύθμιση κάθε φορά των κατάλληλων παραμέτρων. Τα δεδομένα τότε προσαρμόζονται σε κάθε αλγόριθμο, δημιουργούν προβλέψεις συστάδων και εμφανίζονται στο τερματικό.

3.3 Συνολική εκτίμηση

Με μια ευρύτερη εικόνα, παρουσιάζεται μια συγκεκριμένη λογική με την οποία εκτελούνται και την οποία ακολουθούν όλοι οι αλγόριθμοι μηχανικής μάθησης ανεξάρτητα από την βιβλιοθήκη που χρησιμοποιούν. Τα βήματα παραμένουν κοινά με μοναδικές αλλαγές τις απαιτήσεις καθενός από αυτούς σε παραμέτρους ή υπολογισμούς και διαδικασίες. Η δομή είναι συγκεκριμένη για να εξασφαλίζει τη μέγιστη απλότητα και ευκολία στον χρήστη. Οι κώδικες των βιβλιοθηκών Scikit-learn και PyClustering και όλες οι έτοιμες δομές αλγορίθμων, δεν είναι εμφανής στο χρήστη κατά τη διάρκεια εκτέλεσης, αλλά βρίσκονται αποθηκευμένοι σε έναν οργανωμένο χώρο που ονομάζεται GitHub, μια διαδικτυακή πλατφόρμα με ανοιχτό λογισμικό όπου ο καθένας μπορεί να εξερευνήσει.

Κεφάλαιο 4ο: Πειραματική μελέτη

4.1 Πλαίσιο της πειραματικής μελέτης

Η ακόλουθη ενότητα εστιάζει στην πειραματική αξιολόγηση των πέντε αλγορίθμων συσταδοποίησης που μελετήθηκαν στα προηγούμενα Κεφάλαια, DBSCAN, DENCLUE, Mean Shift, OPTICS και HDBSCAN και βασίστηκε σε με μια μέθοδο που εκτίμησε την ποιότητα των συστάδων που παράγει καθένας από αυτούς. Στόχος είναι η έρευνα γύρω από την αποτελεσματικότητα και την ικανότητα αυτών των αλγορίθμων να εντοπίζουν σωστά κατηγοριοποιημένες συστάδες και με σταθερότητα, ακόμη και σε δεδομένα με πιο σύνθετη δομή ή ενδεχόμενο θόρυβο.

Πρωτεύον ρόλο στην πειραματική ανάλυση που διεξήχθη αναλαμβάνουν τα dataset τα οποία επιλέχθηκαν για το σκοπό αυτό μέσα από ένα φάσμα δεδομένων και μετά από δοκιμές. Τα σύνολα αυτά προέρχονται από διάφορους τομείς όπως η υγεία, η βιολογία, η επιστήμη των μηχανικών, το περιβάλλον, η οικονομία αλλά και οι κοινωνικές επιστήμες. Επιπλέον, περιέχουν μια ποικιλία χαρακτηριστικών και δειγμάτων και διατίθενται μέσω της διαδικτυακής πλατφόρμα, UCI Machine Learning Repository, η οποία προσφέρει μια συλλογή από ευρέως γνωστά σύνολα δεδομένων σε επιστήμονές, φοιτητές και διάφορους χρήστες οι οποίοι επιθυμούν να πειραματιστούν.[17]

Για κάθε αλγόριθμο συσταδοποίησης που εξετάστηκε παραπάνω, δόθηκαν προς δοκιμή και εκτέλεση δώδεκα διαφορετικά σύνολα δεδομένων για τα οποία εντοπίστηκαν όλες οι σχηματικές απεικονίσεις που δείχνουν την πορεία της συσταδοποίησης. Ο κώδικας Python για την εκτέλεση κάθε αλγορίθμου σε κάθε περίπτωση έχει δοθεί στο Κεφάλαιο 3ο, ωστόσο κάθε αλγόριθμος προσθέτει κάποια επιπλέον εργαλεία για την οπτικοποίηση των δεδομένων μέσω διαγραμμάτων. Ειδικότερα, παρόλο που και οι πέντε αλγόριθμοι προορίζονται για τον ίδιο σκοπό δεν χρησιμοποιούν απαραίτητα όλοι, τους ίδιου τρόπους απεικόνισης. Ως κοινές μέθοδοι αναπαράστασης γραφημάτων που χρησιμοποιήθηκαν ορίζονται οι εξής:

- Scatter plot: Το λεγόμενο διάγραμμα διασποράς μπορεί να χρησιμοποιηθεί από όλους τους αλγορίθμους συσταδοποίησης είτε βασίζονται στην πυκνότητα είτε όχι και δείχνει τις σχέσεις μεταξύ χαρακτηριστικών που εμπεριέχονται σε ένα σύνολο δεδομένων. Σε κάθε dataset, για τη δημιουργία του scatter plot επιλέγεται κάθε πιθανό ζεύγος χαρακτηριστικών που μπορεί να δημιουργηθεί μέσα από αυτό. Το διάγραμμα απεικονίζεται με δύο άξονες x και y, οι οποίοι αντιπροσωπεύουν τις τιμές των χαρακτηριστικών που επιλέγονται κάθε φορά ενώ κάθε σημείο του χώρου εμφανίζεται ως κουκίδα. Ο χρήστης στο τερματικό του λαμβάνει μια εικόνα που δείχνει πως λειτουργούν τα δεδομένα μεταξύ των γνωρισμάτων. Μια ανοδική πορεία των σημείων στο διάγραμμα σημαίνει μια θετική σχέση μεταξύ των χαρακτηριστικών όπου όταν αυξάνεται η τιμή του ενός, αυξάνεται και η τιμή του άλλου και ο αλγόριθμος θεωρεί ότι δημιουργούνται καλά διαχωρισμένες συστάδες. Αντίθετα, μια καθοδική πορεία ορίζει μια αρνητική σχέση μεταξύ των χαρακτηριστικών όπου όταν η τιμή του ενός αυξάνεται, η τιμή του άλλου μειώνεται και τότε ο αλγόριθμος θεωρεί αυτή την περίπτωση πιο απαιτητική. Αν τα σημεία εμφανίζονται διασκορπισμένα και δεν υπάρχει καμία συσχέτιση μεταξύ τους τότε πιθανών τα δεδομένα να βρίσκονται πιο αραιά και η συσταδοποίηση γίνεται εμφανώς πιο δύσκολα.[21] Ο κώδικας Python που χρησιμοποιήθηκε για τη λειτουργία του είναι ο εν λόγω:

```
39    for i in range(data.shape[1]):
40     for j in range(i+1, data.shape[1]):
41       sns.scatterplot(x=banana.columns[i], y=banana.columns[j], hue='cluster', data=banana, palette='viridis')
42       plt.title(f'DBSCAN Clustering of tae Dataset ({banana.columns[i]} vs {banana.columns[j]})')
43       plt.savefig(f'scatterplot_tae_{banana.columns[i]}_{banana.columns[j]}.png')
44       plt.show()
```

Εικόνα 4.1 Στιγμιότυπο από την υλοποίηση του Scatter Plot στο Spyder

Ο κώδικας του scatter plot μπορεί να πραγματοποιηθεί για κάθε ζεύγος ξεχωριστά με προσαρμοσμένο κώδικα αλλά για μεγαλύτερη ευκολία και εξοικονόμηση, ιδίως σε dataset με

πολλά χαρακτηριστικά, χρησιμοποιείται η βοήθεια βρόχων. Για την παρούσα εργασία, επιλέχθηκε αυτή η επαναληπτική διαδικασία.

Ξεκινώντας την ανάλυση, ο δείκτης *i* που εμφανίζεται στη γραμμή 39, αντιπροσωπεύει το κάθε χαρακτηριστικό που θα τοποθετηθεί στον άξονα x και το οποίο λαμβάνεται από το πλήθος των γνωρισμάτων του dataset. Ομοίως, ο δείκτης j για τον άξονα y παίρνει με τη σειρά του ένα άλλο χαρακτηριστικό παραλείποντας κάθε φορά εκείνο που έχει επιλέξει ήδη ο πρώτος βρόχος, ώστε να μην υπάρξουν διπλές επαναλήψεις. Για κάθε ζευγάρι χαρακτηριστικών που εντοπίζεται, δημιουργείται ένα διάγραμμα διασποράς καλώντας την συνάρτηση *scatterplot.* Η συνάρτηση αυτή είναι υπεύθυνη για τη δημιουργία του, εντοπίζεται στην βιβλιοθήκη Seaborn και προσδιορίζεται στην αρχή του κώδικα όπως αναφέρθηκε και στο Κεφάλαιο 3ο με την εντολή *import seaborn as sns*. Η *scatterplot* δίνει τα χαρακτηριστικά που εντοπίστηκαν σε κάθε δείκτη κάθε φορά στους άξονες x και y αντίστοιχα και επιλέγει τα cluster του dataset από όπου θα πάρει τις πληροφορίες. Τέλος, δόθηκαν ονόματα στους άξονες αλλά και στο ίδιο το γράφημα ώστε να γίνει πιο ευανάγνωστο.

- Parallel Coordinates: Το διάγραμμα παράλληλων συντεταγμένων αποτελεί εξίσου ένα τρόπο απεικόνισης των αποτελεσμάτων της συσταδοποίησης και μπορεί να εφαρμοστεί σε όλους τους αλγορίθμους ανεξαρτήτως, ενώ είναι ικανό να εμφανίσει τις σχέσεις μεταξύ όλων των χαρακτηριστικών ενός dataset αντίθετα με το Scatter plot. Συγκεκριμένα, στο διάγραμμα αυτό κάθε γνώρισμα αντιπροσωπεύεται από έναν κάθετο άξονα και καθένας από αυτούς τοποθετείται δίπλα στον άλλον παράλληλα και με την σειρά εμφάνισης τους στο dataset. Κάθε δείγμα από το σύνολο δεδομένων που επιλέγεται, εμφανίζεται στο γράφημα ως μια γραμμή που ακολουθεί ένα μονοπάτι ενώνοντας όλα τα χαρακτηριστικά μεταξύ τους ανάλογα με τις τιμές που παίρνει κάθε φορά. Με την αξιολόγηση του, συμπεραίνουμε ότι, όταν τα δεδομένα μιας συστάδας ακολουθούν παρόμοιο μονοπάτι δείχνουν μια πιο ισχυρή σχέση μεταξύ των χαρακτηριστικών ιδιαίτερα και πιθανών να δημιουργούν ένα μοτίβο. Αντίθετα, όταν οι γραμμές ακολουθούν τυχαίες διαδρομές τότε θεωρείται ότι οι τιμές τους δεν είναι κοινές στα χαρακτηριστικά, τα δεδομένα μπορεί να ανήκουν σε διαφορετικές ομάδες ή εντός μιας συστάδας να βρίσκονται πιο αραιά.[22] Ακολουθείται ο απαιτούμενος κώδικας σε γλώσσα Python για τη λειτουργία του:

```
49    df2 = pd.DataFrame(banana.iloc[:, 0:5])
50    df2['Clusters'] = dbscan.labels_
51    print(df2)
52    parallel_coordinates(df2, 'Clusters', color=['red', 'blue', 'green', 'yellow', 'black'])
53    plt.title('Parallel Coordinates for DBSCAN Clustering')
54    plt.savefig('parallel_coordinates_tae.png')
55    plt.show()
```

Εικόνα 4.2 Στιγμιότυπο από την υλοποίηση του Parallel Coordinates στο Spyder

Κατά τη διάρκεια συγγραφής του κώδικα αφού πραγματοποιηθεί η συσταδοποίηση με τον επιθυμητό αλγόριθμο και δημιουργηθεί το data frame των συστάδων, το διάγραμμα παράλληλων συντεταγμένων έρχεται για να δείξει την σχέση μεταξύ των χαρακτηριστικών στη διαδικασία της ομαδοποίησης. Για μεγαλύτερη ευκολία και για να αποφύγουμε ασήμαντες πληροφορίες, δημιουργείται ένα καινούριο data frame, το οποίο εμπεριέχει όλες τις γραμμές και στήλες από το dataset που χρησιμοποιήθηκε, εκτός από την τελευταία όπου περιέχονται συνήθως οι ετικέτες, προσθέτοντας σε αυτό παράλληλα τις συστάδες που εντοπίστηκαν από τον εκάστοτε αλγόριθμο. Στη συνέχεια καλείται η συνάρτηση *parallel_coordinates()*, η οποία εισάγεται μέσω της βιβλιοθήκης Pandas στην αρχή του κώδικα (*from pandas.plotting import parallel_coordinates*) ώστε να μπορέσει να κατασκευαστεί το γράφημα. Το διάγραμμα τότε δημιουργείται λαμβάνοντας υπόψη το νέο data frame και τις συστάδες που ανιχνεύθηκαν.

Και τα δύο αυτά σχεδιαγράμματα μπορούν να χρησιμοποιηθούν για την οπτικοποίηση των δεδομένων, εξυπηρετούν όμως διαφορετικούς σκοπούς και χρησιμοποιώντας και τα δυο αποκτάται μια πιο πλήρης εικόνα των δεδομένων.

Ωστόσο, οι αλγόριθμοι DBSCAN και OPTICS δεν περιορίζονται μόνο σε αυτά, αλλά προσθέσουν παράλληλα σχεδιαγράμματα που ενισχύουν την κατανόηση της πορείας της συσταδοποίησης. Συγκεκριμένα:

- K-distance graph: Ο αλγόριθμος DBSCAN προσθέτει το γράφημα k-απόστασης με το οποίο μπορεί να αναγνωρίσει την ιδανική απόσταση Eps που θα καθορίσει την περιοχή γειτονιάς για κάθε σημείο. Ειδικότερα, το k-distance graph εμφανίζει όλες τις αποστάσεις των σημείων από τους k γείτονες τους κάθε φορά. Το γράφημα αυτό αποτελείται από ένα κάθετο άξονα y, ο οποίος αντιπροσωπεύει αυτές τις αποστάσεις αλλά και από έναν οριζόντιο άξονα x, ο οποίος εκπροσωπεί όλα τα σημεία του dataset. Στο γράφημα πλέον δημιουργείται μια πορεία. Στη θέση όπου παρουσιάζεται μια απότομη αύξηση και υπάρχει μεγαλύτερη κλίση στην μέχρι τώρα πορεία των δεδομένων, εντοπίζεται η ιδανική τιμή της ακτίνας Eps ενώ συμπεραίνουμε πως σημεία πριν από αυτήν αποτελούν πυκνές περιοχές και σημεία που ξεκινάνε αμέσως μετά, αραιές περιοχές ή θόρυβο.[9][24] Ο κώδικας για την υλοποίηση του k-distance graph δίνεται παρακάτω:

```
25    distances = np.sort(distances, axis=0)
26    distances = distances[:, 1]
27    plt.plot(distances)
28    plt.title('K-Distance Plot')
29    plt.xlabel('Points sorted by distance')
30    plt.ylabel('Epsilon distance')
31    plt.savefig('dbscan_tae_kdistances.png')
32    plt.show()
```

Εικόνα 4.3 Στιγμιότυπο από την υλοποίηση του K-distance graph στο Spyder

Αρχικά, ταξινομούνται οι αποστάσεις μεταξύ των γειτόνων ενός σημείου που εντοπίστηκαν, με τη μέθοδο *sort()* και με βάση τα χαρακτηριστικά τους κάθε φορά ώστε να βρεθεί ο k κοντινότερος από αυτά. Έπειτα, από τον πίνακα που δημιουργείται επιλέγεται η δεύτερη στήλη που περιέχει τον κοντινότερο γείτονα προς κάθε σημείο, ενώ η πρώτη στήλη αγνοείται αφού αφορά τις αποστάσεις προς τον εαυτό τους που οι τιμές είναι πάντα μηδενικές. Ακολούθως κατασκευάζεται το σχεδιάγραμμα μέσω την μεθόδου *plot()* που εισάγεται από την βιβλιοθήκη Matplotlib στην αρχή του κώδικα (*import matplotlib.pyplot as plt*).

- Reachability plot: Όπως αναφέρθηκε και στο Κεφάλαιο 2ο, το διάγραμμα προσβασιμότητας είναι απαραίτητο εργαλείο στη λειτουργία του αλγορίθμου OPTICS, αφού εκεί αποθηκεύονται όλες οι αποστάσεις προσβασιμότητας των σημείων που εντοπίζονται. Ωστόσο, για να μπορέσει να ερμηνευτεί πιο εύκολα η δομή αυτής της ουράς, μπορεί να δημιουργηθεί ένα σχεδιάγραμμα το οποίο θα την αναπαριστά μέσω αξόνων. Στον οριζόντιο άξονα x δηλώνονται όλα τα σημεία του χώρου με της σειρά που εξετάστηκαν ενώ ο άξονας y λαμβάνει τις τιμές των αποστάσεων προσβασιμότητας για κάθε ένα από αυτά. Σύμφωνα με το γράφημα σημεία όπου είναι πολύ κοντά μεταξύ τους με μικρές αποστάσεις προσβασιμότητας, δείχνουν μία συστάδα ενώ σημεία με πιο μεγάλες τιμές και απότομες αυξομειώσεις που δε συνδέονται με κάποιο τρόπο με τις υπόλοιπες περιοχές υποδεικνύουν τα σημεία θορύβου. Δίνεται ο κώδικας Python για τη δημιουργία του γραφήματος reachability:

```
51    reachability = optics.reachability_[optics.ordering_]
52    labels_ordered = optics.labels_[optics.ordering_]
53
54    plt.figure(figsize=(10, 5))
55    plt.scatter(np.arange(len(reachability)), reachability, c=labels_ordered, cmap='viridis', marker='.')
56    plt.colorbar(label='Cluster Label')
57    plt.ylabel('Reachability Distance')
58    plt.title('OPTICS Reachability Plot')
59    plt.xlabel('Sample Index')
60    plt.savefig('optics_reachability_plot_colored.png')
61    plt.show()
```

Εικόνα 4.4 Στιγμιότυπο από την υλοποίησης του Reachability plot στο Spyder

Αρχικά, εισάγονται ταξινομημένα οι αποστάσεις προσβασιμότητας με τη σειρά εξέτασης των σημείων σε μια παράμετρο ενώ με τον ίδιο τρόπο και σειρά λαμβάνονται για καθένα από

αυτά και η συστάδα στην οποία ανήκει. Για την δημιουργία του reachability plot, καλείται η εντολή *scatter()* δίνοντας τις τιμές αυτές σε κάθε άξονα αλλά και προσφέροντας διάφορες ιδιότητες που θα κάνουν το γράφημα πιο ευανάγνωστο.

Συνολικά, τα διαγράμματα αποτελούν σημαντικό παράγοντα για να μπορέσει ο κάθε χρήστης να κατανοήσει και να αξιολογήσει έναν αλγόριθμο προσφέροντας ιδιαίτερες χρήσιμες πληροφορίες για αποφάσεις που θα λάβει αργότερα.

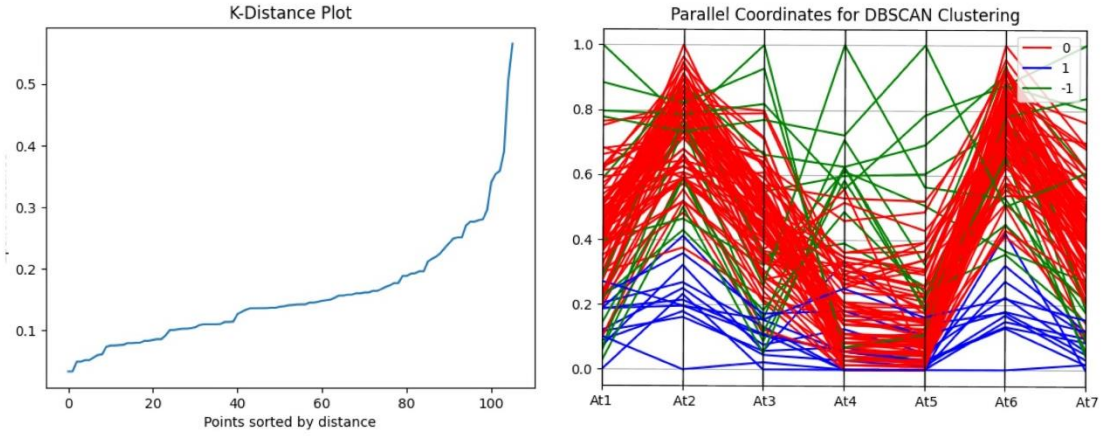
Παρακάτω αναλύονται με αλφαβητική σειρά μόνο τα data set τα οποία χρησιμοποιήθηκαν στην πειραματική μελέτη της παρούσας εργασίας, ο κώδικας κλήσης για όλους τους επιμέρους αλγορίθμους σε κάθε ένα από αυτά με τις αντίστοιχες παραμέτρους, οι πίνακες τον αποτελεσμάτων που παράχθηκαν αλλά και όλες οι απαραίτητες γραφικές αναπαραστάσεις που αναλύθηκαν παραπάνω:

4.1.1 Appendicitis

Το πρώτο dataset που χρησιμοποιήθηκε για μελέτη είναι το Appendicitis. Το σύνολο δεδομένων αυτό προέρχεται από τον κλάδο της Ιατρικής και περιέχει δείγματα ανθρώπων που εμφανίζουν ή όχι της πάθησης της σκωληκοειδίτιδας σύμφωνα με τα μοτίβα που δημιουργούν. Αν και ο αριθμός των δεδομένων μπορεί να φανεί ανεπαρκές σε Ιατρικές εφαρμογές ωστόσο αποτελεί χρήσιμο dataset σε περιπτώσεις κατανόησης και δοκιμής. Περιέχει 106 δείγματα ανθρώπων με 7 διαφορετικά χαρακτηριστικά που περιγράφουν κλινικές μετρήσεις και προσδιορίζουν ανάλογα τον ασθενή. Οι μετρήσεις αυτές μπορεί να αφορούν συμπτώματα που δείχνουν το πρόβλημα της σκωληκοειδίτιδας ή άλλα σχετιζόμενα με την υγεία των ασθενών κριτήρια.[20]

DBSCAN: dbscan = DBSCAN (eps= 0.3, min_samples= 9)

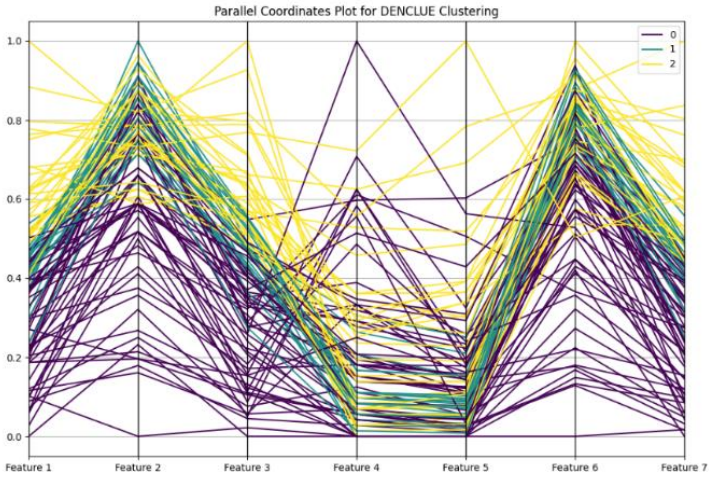
Σύμφωνα με το k-distance plot που δημιουργήθηκε για το dataset Appendicitis, η καλύτερη τιμή που μπορεί να πάρει η ακτίνα Eps στον αλγόριθμο DBSCAN για την δημιουργία αξιόπιστων συστάδων εντοπίζεται εκεί όπου υπάρχει απότομη κλίση. Το διάγραμμα της Εικόνας 4.5, δείχνει την τιμή αυτή να κυμαίνεται κοντά στο 0.3.



Εικόνα 4.5 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το Appendicitis

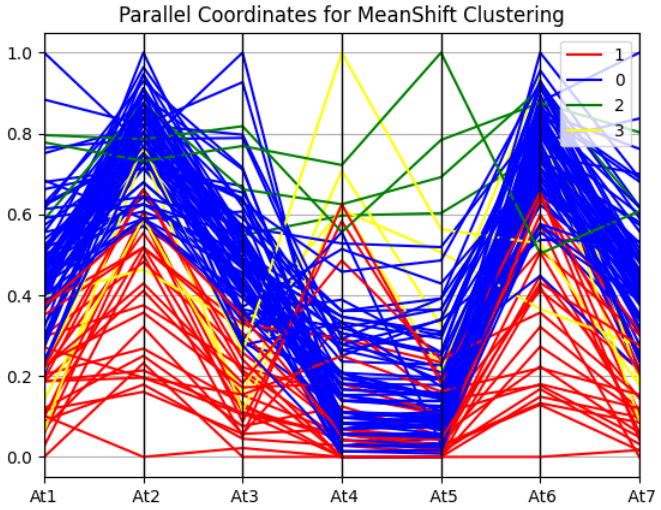
Αντίστοιχα, στο διάγραμμα παράλληλων συντεταγμένων που παράχθηκε μετά την εκτέλεση του αλγορίθμου, εντοπίζονται δύο συστάδες που ξεχωρίζουν με κόκκινο και μπλε χρώμα. Ο θόρυβος στο γράφημα είναι γραμμές που ακολουθούν διαφορετικά μονοπάτια και προσδιορίζεται με πράσινο χρώμα και την τιμή -1. Τα σημεία που εντάχτηκαν στο cluster 0 δείχνουν να είναι πιο συγκεντρωμένα σε σχέση με το cluster 1, δημιουργώντας ένα μοτίβο, αφού όλες οι τιμές αυξομειώνονται ανάλογα. Οι πιο υψηλές τιμές για αυτήν την ομάδα παρουσιάζονται στο At2 και At6 ενώ οι χαμηλότερες στο At4 και At5. Το cluster 1 από την άλλη μεριά, παρουσιάζει μια πιο ποικιλόμορφη δομή όπου τα δεδομένα φαίνονται να μην είναι τόσο πυκνά μεταξύ τους με τα σημεία να ακολουθούν διαφορετικές διαδρομές στα χαρακτηριστικά. Ο διαχωρισμός των δύο ομάδων φαίνεται πιο ξεκάθαρα στο At2 και At6.

DENCLUE: denclue = DENCLUE (data_scaled, bandwidth = 0.3, threshold=0.2)



Εικόνα 4.6 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το Appendicitis

Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples = 106)
Meanshift = Meanshift (bandwidth=bandwidth)



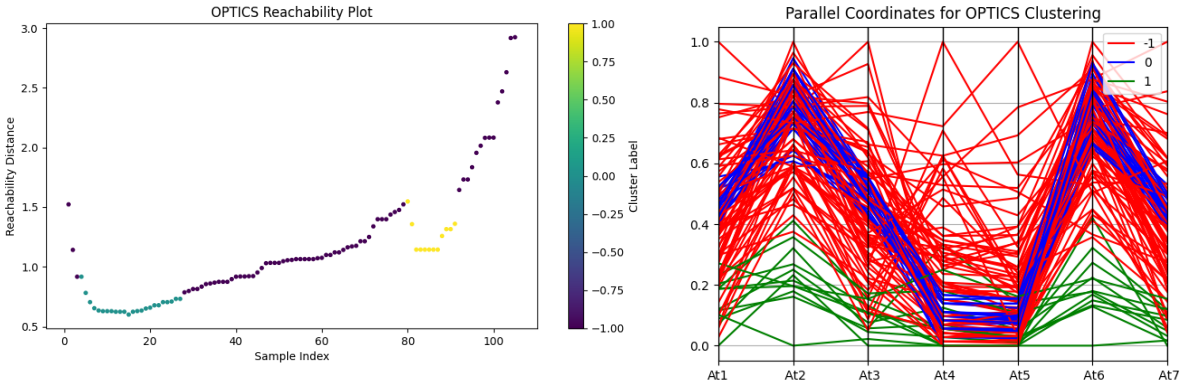
Εικόνα 4.7 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Appendicitis

Όπως διακρίνεται από τις Εικόνες 4.6 και 4.7, οι αλγόριθμοι DENCLUE και Mean Shift πάνω στο ίδιο σύνολο δεδομένων καταφέρνουν να ομαδοποιήσουν τα σημεία με τέτοιο τρόπο με αποτέλεσμα να μην υπάρχει κανένα noise point και όλα τα σημεία του χώρου να έχουν τοποθετηθεί σε συστάδες. Η ομαδοποίηση αυτή δεν αποτελεί απαραίτητα καλό στοιχείο των αλγορίθμων αφού μπορεί πολύ εύκολα να χαθούν χρήσιμες πληροφορίες για το διαχωρισμό. Μια τέτοια περίπτωση υπόκειται σε δυο σενάρια. Είτε τα δεδομένα είναι αρκετά όμοια μεταξύ τους, είτε η παραμετροποίηση ή χρήση των συγκεκριμένων αλγορίθμων δεν αποτελεί κατάλληλη επιλογή για το συγκεκριμένο dataset.

Σύμφωνα με τα παραγόμενα διαγράμματα, για τον αλγόριθμο DENCLUE, το cluster 1 φαίνεται πως είναι το μοναδικό που ακολουθεί κάποιο μοτίβο αφού τα χαρακτηριστικά του σε αυτό φαίνονται να σχετίζονται άμεσα μεταξύ τους και να δημιουργούν ομοιόμορφη συστάδα. Από την άλλη πλευρά, οι τιμές στα cluster 0 και 2, ακολουθούν δικά τους μονοπάτια ενώ φαίνεται να μην είναι αρκετά συμπαγής με μεγάλες διακυμάνσεις. Η διαφοροποίηση των τριών ομάδων γίνεται ελαφρά εμφανέστερη στα χαρακτηριστικά Feature 2 και Feature 6.

Όσον αφορά τον αλγόριθμο Mean Shift, δημιουργούνται 4 συστάδες εκ των οποίων μόνο το cluster 0 κατά γενική εικόνα οδηγεί σε ένα μοτίβο με τις τιμές να αυξομειώνονται αντίστοιχα. Το cluster 3 δεν είναι ιδιαίτερα εμφανίσιμο με αποτέλεσμα να προμηνύει κάποια επικάλυψη συστάδων.

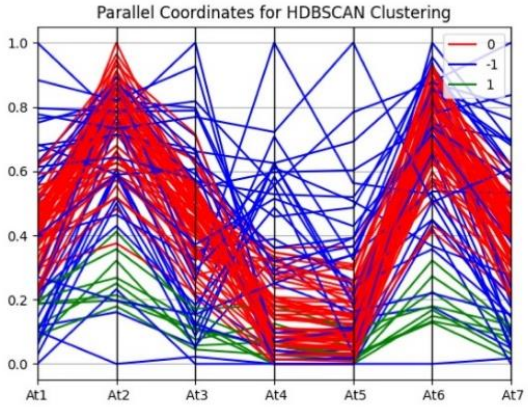
OPTICS: optics = OPTICS (min_samples = 8, min_cluster_size= 0.05)



Εικόνα 4.8 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICS για το Appendicitis

Στον αλγόριθμο OPTICS για το Appendicitis, μέσω του reachability plot μπορούμε να ξεχωρίσουμε τα σημεία όπου βρίσκονται κοντά μεταξύ τους και έχουν χαμηλές αποστάσεις προσβασιμότητας και εντάσσονται σε συστάδες. Τα σημεία διακρίνονται δημιουργώντας διαφορετικές κοιλάδες στο σχεδιάγραμμα. Οι συστάδες αυτές είναι φανερές και στο διάγραμμα παράλληλων συντεταγμένων το οποίο φαίνεται να τις διαχώρισε ομοιόμορφα. Ωστόσο, και στα δύο διαγράμματα, διακρίνονται αρκετά έντονα τα σημεία θορύβου, γεγονός που οδηγεί σε πιθανή λανθασμένη παραμετροποίηση ή τα δεδομένα είναι αρκετά διάσπαρτα ώστε ο αλγόριθμος OPTICS να τα εντάξει σε συστάδες. Ο διαχωρισμός των δύο συστάδων που εντοπίστηκαν είναι φανερός στα γνωρίσματα At2 και At6 με το cluster 0 να αποτελεί το πιο πυκνό.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size=7)



Εικόνα 4.9 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το Appendicitis

Στον αλγόριθμο HDBSCAN, παρατηρούνται δύο συστάδες. Το cluster 0 αποτελεί το πιο πυκνό και είναι αυτό που διατηρεί το μοτίβο ενώ σε βήματα ομοιογένειας βρίσκεται και το cluster 1 αν και πιο αραιό. Ωστόσο, αν και τα σημεία θορύβου είναι αρκετά η συσταδοποίηση φαίνεται να πραγματοποιήθηκε με επιτυχία.

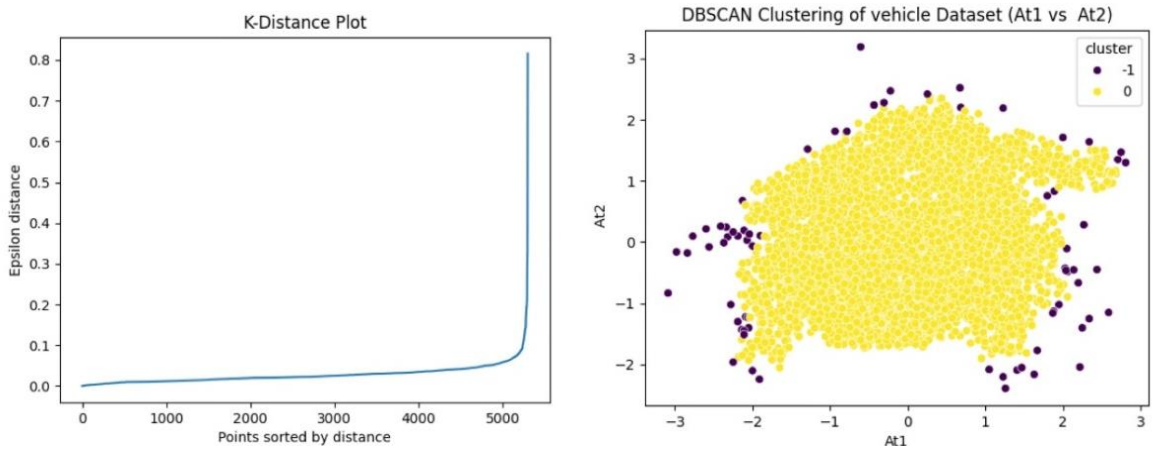
APPENDICITIS	CL0	CL1	CL2	CL3
DBSCAN	80	12	-	-
DENCLUE	52	24	30	-
MEANSHIFT	73	25	4	4
OPTICS	24	12	-	-
HDBSCAN	65	8	-	-

Πίνακας 1. Συνολικά Cluster του Appendicitis

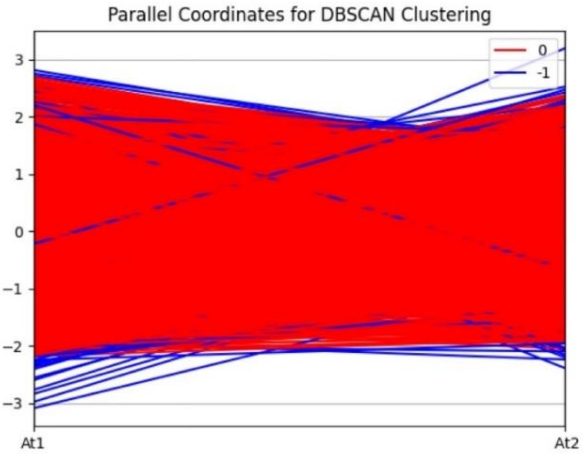
4.1.2 Banana

Το Banana dataset αφορά ένα σύνολο πληροφοριών που πήρε το όνομα του από τον τρόπο με τον οποίο απεικονίζονται τα δεδομένα του σε γράφημα. Τα σημεία εμφανίζονται με τέτοιο τρόπο, δημιουργώντας καμπύλες που φαίνονται σαν "μπανάνες" που αλληλεπικαλύπτονται. Αποτελείται από μόνο 2 χαρακτηριστικά που αφορούν τις διαστάσεις κάθε σημείου στο χώρο και περιέχει 5.300 δείγματα δεδομένων. Το σύνολο αυτό είναι ευρέως γνωστό αφού χρησιμοποιείται για τη δοκιμή και κατόπιν την αξιολόγηση διαφόρων αλγορίθμων.[20]

DBSCAN: dbscan = DBSCAN (eps=0.15, min_samples=6)



Εικόνα 4.10 Εξαγόμενα γραφήματα K-distance plot και Scatter plot του DBSCAN για το Banana

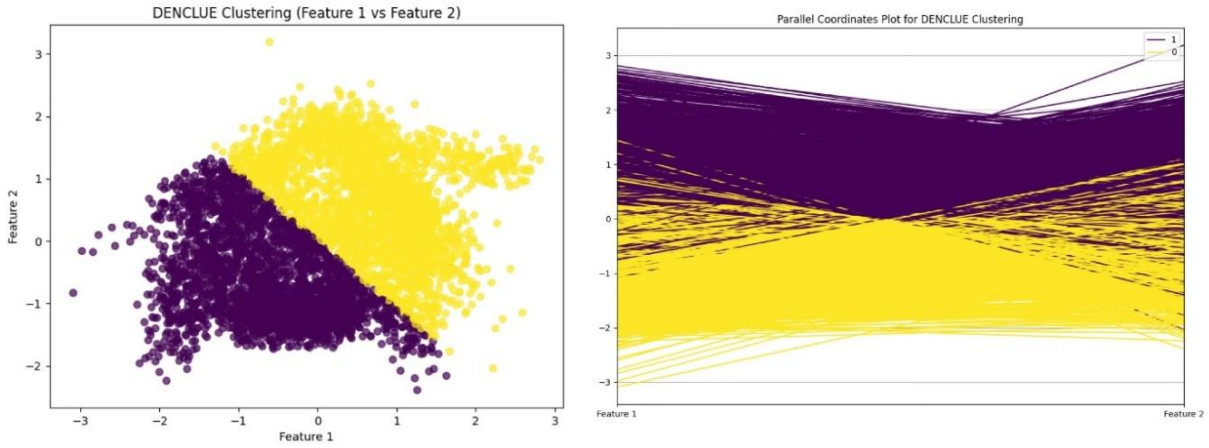


Εικόνα 4.11 Εξαγόμενο γράφημα Parallel Coordinates του DBSCAN για το Banana

Στον αλγόριθμο DBSCAN για το dataset Banana, η βέλτιστη τιμή για το ελάχιστο μέγεθος εμβέλειας αναζήτησης γειτόνων εντοπίζεται γύρω στο 0.1 όπως παρατηρείται από το k-distance graph Παράλληλα ανιχνεύει μόνο μια συστάδα αφού με βάση την ακτίνα Eps τα σημεία είναι αρκετά πυκνά και βρίσκονται πολύ κοντά μεταξύ τους ώστε να διαχωριστούν σε διαφορετικά cluster. Οι τιμές των χαρακτηριστικών φαίνεται να είναι αρκετά παρόμοιες και τα σημεία θορύβου ευδιάκριτα.

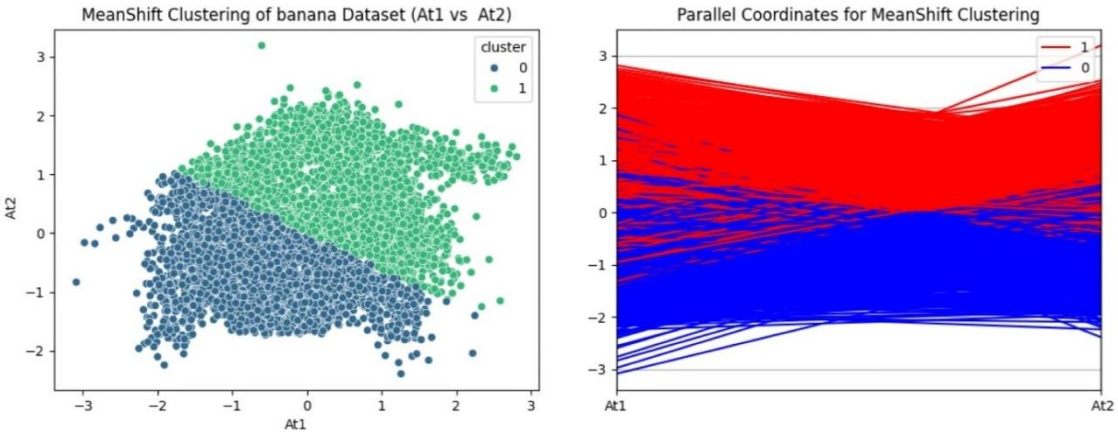
DENCLUE: denclue = DENCLUE (data_scaled, bandwidth=0.15, threshold=0.1)

Και στην περίπτωση του Banana, στους αλγορίθμους DENCLUE (Εικόνα 4.12) και Mean Shift (Εικόνα 4.13), τα δεδομένα διαχωρίστηκαν σχεδόν ισάξια σε δυο μόλις συστάδες χωρίς κανένα Noise point να κάνει την εμφάνιση του. Και οι δύο εκδοχές εμφανίζουν μια παρόμοια συσταδοποίηση αν και κάθε αλγόριθμος λειτούργησε με το δικό του τρόπο. Ο διαχωρισμός των cluster είναι εμφανής μεταξύ των δύο χαρακτηριστικών και στους δύο αλγορίθμους ενώ οι τιμές φαίνεται να ακολουθούν κοινή πορεία.



Εικόνα 4.12 Εξαγόμενα γραφήματα Scatter plot και Parallel Coordinates του DENCLUE για το Banana

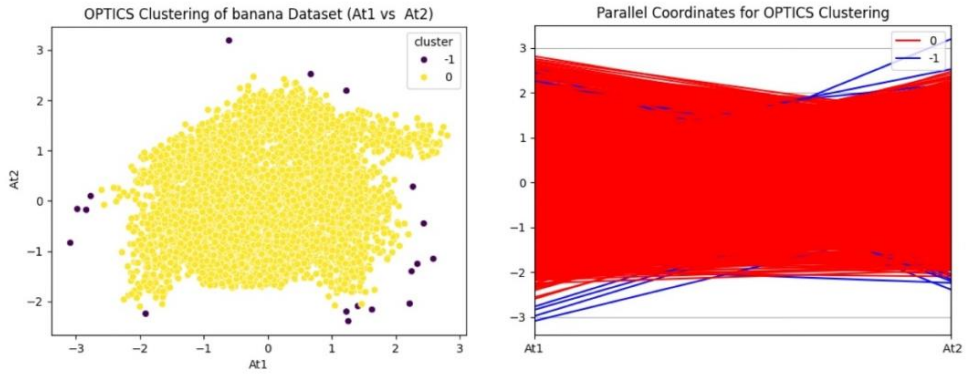
Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples=5300)
Meanshift = Meanshift (bandwidth=bandwidth)

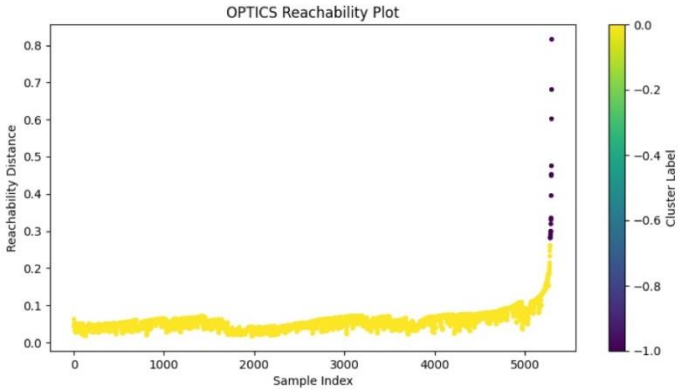


Εικόνα 4.13 Εξαγόμενα γραφήματα Scatter plot και Parallel Coordinates του Mean Shift για το Banana

OPTICS: optics = OPTICS (min_samples =6, min_cluster_size=0.2)

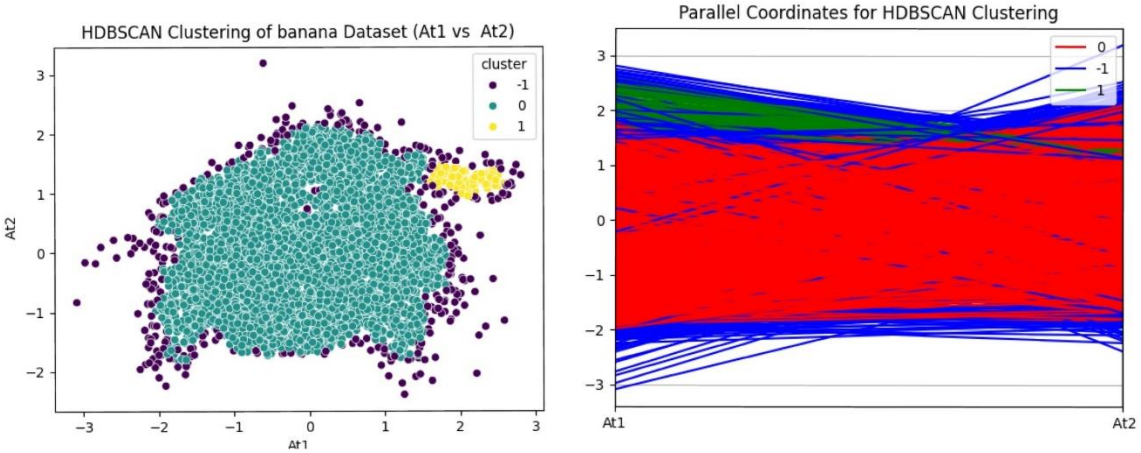
Στην άλλη μεριά, ο αλγόριθμος OPTICS, εντοπίζει μόνο μια συστάδα, τα σημεία της οποίας εμφανίζονται στον άξονα x του reachability plot και φαίνεται να έχουν σχεδόν την ίδια απόσταση προσβασιμότητας το κάθε ένα. Συνέπεια αυτού, να δημιουργούν μια μεγάλη επίπεδη κοιλάδα. Τα σημεία αυτά θεωρούνται ως μια πυκνή περιοχή, άρα μια συστάδα, η οποία καλύπτει σχεδόν ολόκληρο το σύνολο δεδομένων και τερματίζει τη στιγμή που οι τιμές προσβασιμότητας αρχίζουν να απομακρύνονται.





Εικόνα 4.14 Εξαγόμενα γραφήματα Scatter plot, Parallel Coordinates και Reachability plot του OPTICS για το Banana

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size= 8)



Εικόνα 4.15 Εξαγόμενα γραφήματα Scatter plot και Parallel Coordinates του HDBSCAN για το Banana

Εκτελώντας τον αλγόριθμο HDBSCAN τα αποτελέσματα που προκύπτουν είναι ο διαχωρισμός του dataset και πάλι σε δυο ομάδες όμως σαφώς πιο ομοιόμορφες μεταξύ τους. Συγκριτικά, το cluster 0 καταλαμβάνει μια μεγαλύτερη περιοχή σε σχέση με το cluster 1 ενώ ο διαχωρισμός τους είναι εμφανής και στο parallel coordinates στο χαρακτηριστικό At1.

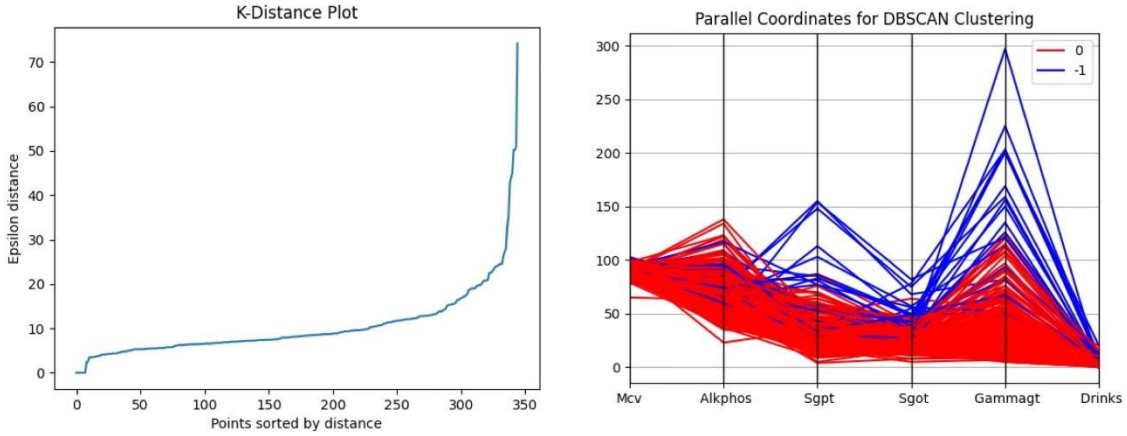
BANANA	CL0	CL1
DBSCAN	5231	-
DENCLUE	2754	2546
MEANSHIFT	2710	2590
OPTICS	5282	-
HDBSCAN	4881	87

Πίνακας 2. Συνολικά Cluster του Banana

4.1.3 Bupa

Το σύνολο δεδομένων Bupa αποτελεί επίσης dataset σχετικό με τον τομέα της υγείας και χρησιμοποιείται για την πρόβλεψη ύπαρξης ή μη προβλήματος του ήπατος. Αποτελείται από 345 δείγματα ασθενών με 6 διαφορετικά χαρακτηριστικά μετρήσεων που αφορούν τον τρόπο ζωής καθενός, παραδείγματος χάριν το επίπεδο αλκοόλ στο αίμα τους αλλά και την γενικότερη υγεία τους. Αν και χρησιμοποιείται κυρίως σε αλγορίθμους ταξινόμησης λόγω του απλού διαχωρισμού που κάνει σε δύο κατηγορίες αν κάποιος νοσεί ή όχι, μπορεί να χρησιμοποιηθεί και σε αλγορίθμους συσταδοποίησης ομαδοποιώντας τους ασθενείς ανάλογα με τα χαρακτηριστικά τους.[20]

DBSCAN: dbscan = DBSCAN (eps=25.0, min_samples=5)



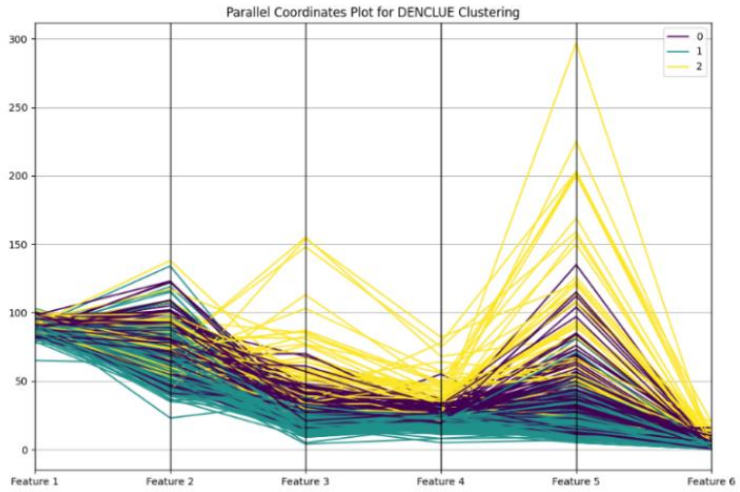
Εικόνα 4.16 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το Bupa

Η ιδανικότερη τιμή της ακτίνας Eps για το σύνολο δεδομένων Bupa εντοπίζεται κοντά στο σημείο 25. Το διάγραμμα παράλληλων συντεταγμένων απεικονίζει την μοναδική συστάδα που ανίχνευσε ο αλγόριθμος DBSCAN, με τις ακραίες τιμές να είναι ξεκάθαρες. Το cluster φαίνεται να παρουσιάζει ομοιογένεια ενώ δεν εμφανίζει ιδιαίτερες διακυμάνσεις μεταξύ των χαρακτηριστικών και συνεχίζει με κοινή πορεία.

DENCLUE:

Denclue = DENCLUE (data_scaled, bandwidth=1.5, threshold=0.6)

Στην περίπτωση του DENCLUE τα δείγματα καταφέρνουν να κατανεμηθούν σε τρείς διαφορετικές συστάδες χωρίς την εύρεση θορύβου. Ο διαχωρισμός μεταξύ τους είναι πιο ευδιάκριτος στο Feature 3 και 4 όπου κάθε συστάδα έχει τιμές από διαφορετικό εύρος. Τα cluster 0 και 1 είναι τα πιο ομοιογενή με τη μεγαλύτερη διακύμανση να παρουσιάζεται στο Feature 5. Το cluster 2 από την άλλη πλευρά εμφανίζεται ως μια πιο διάσπαρτη περιοχή με μεγάλες διαφορές τιμών μεταξύ των χαρακτηριστικών.

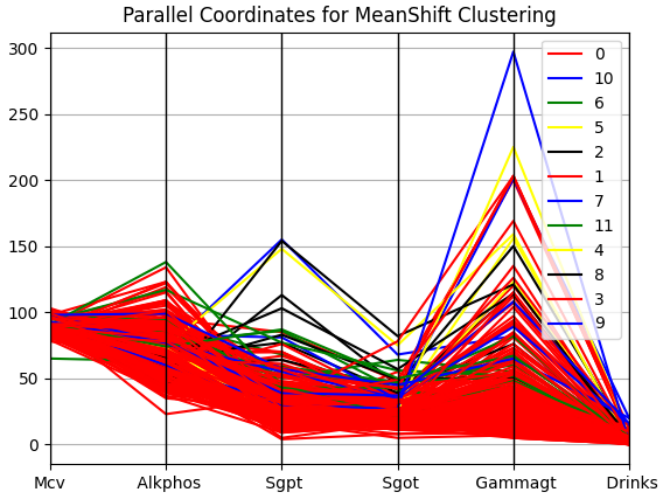


Εικόνα 4.17 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το Bupa

Mean Shift: Bandwidth = estimate_bandwidth (data_scaled, n_samples=345)
Meanshift = Meanshift (bandwidth=bandwidth)

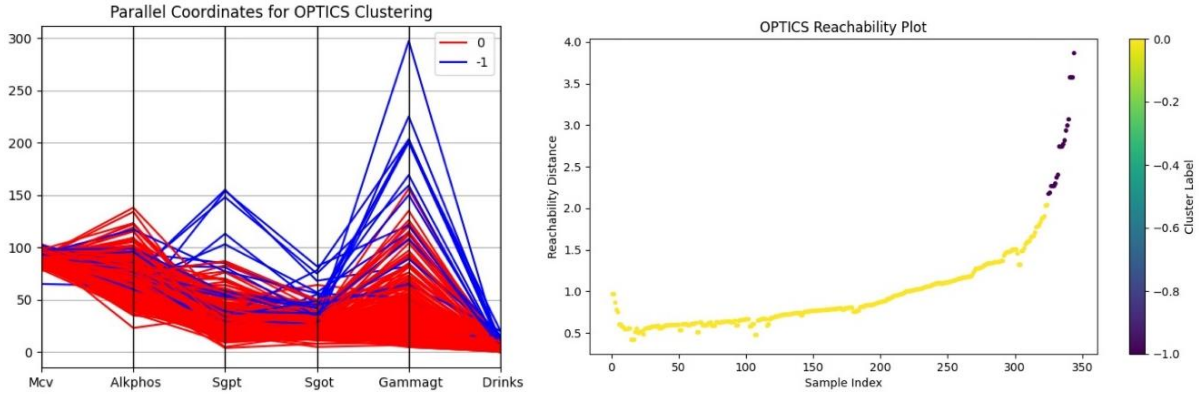
Ο Mean Shift στο dataset Bupa, δημιουργεί πολλά μικρά cluster χωρίς την ένδειξη θορύβου όπως φαίνεται και από την Εικόνα 4.18. Οι συστάδες έλαβαν κοινά χρώματα χωρίς να γίνεται ευδιάκριτη η

διαφορά μεταξύ τους και μεταξύ των χαρακτηριστικών τους, με τις μεγαλύτερες τιμές να εμφανίζονται στο Gammagt. Ακόμη, σύμφωνα με το πλήθος των σημείων που εισήγαγε η κάθε μια όπως φαίνεται παρακάτω στον Πίνακα 3, πολλές από αυτές δημιουργήθηκαν λανθασμένα αφού δε θα μπορούσε να θεωρηθεί πυκνή μια συστάδα με μόνο ένα σημείο.



Εικόνα 4.18 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Bupa

OPTICS: optics = OPTICS (min_samples =4, min_cluster_size=0.1)

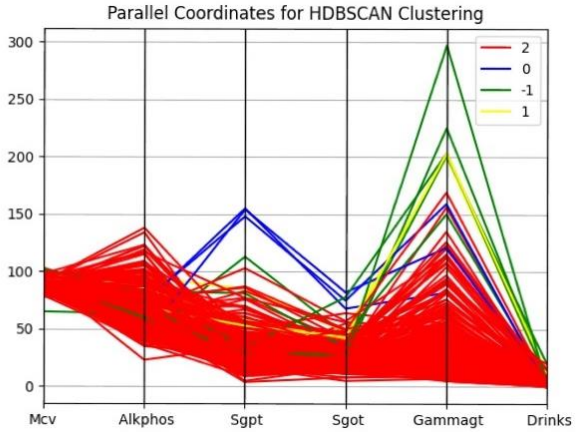


Εικόνα 4.19 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICS για το Bupa

Η μοναδική συστάδα που δημιουργείται στον αλγόριθμο OPTICS ακολουθεί κοινή διαδρομή σχεδόν σε όλα τα χαρακτηριστικά ενώ ο θόρυβος αποτελείται από διάσπαρτες γραμμές στο χώρο. Η ομαδοποίηση που πραγματοποίησε φαίνεται να μοιάζει ιδιαίτερα με του DBSCAN (Εικόνα 4.16), με το cluster να παρουσιάζει και πάλι σχετική ομοιογένεια με ελάχιστες διακυμάνσεις. Με την ανίχνευση μόνο μιας ομάδας ωστόσο πιθανών να έχουν χαθεί χρήσιμες πληροφορίες που να διαφοροποιούν τα δεδομένα μεταξύ τους.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size=2)

Σύμφωνα με τις Εικόνα 4.20, ο HDBSCAN ανίχνευσε τρείς συστάδες πάνω στο dataset Bupa. Το cluster 2 εμφανίζεται πιο πυκνό και φαίνεται να καταλαμβάνει το μεγαλύτερο χώρο. Αντίθετα, τα cluster 0 και 1 δεν είναι ιδιαίτερα εμφανή γεγονός που δείχνει πιθανή επικάλυψη μεταξύ αυτών ενώ παράλληλα δε παρουσιάζουν κάποιο μοτίβο στα δεδομένα τους.



Εικόνα 4.20 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το Bupa

BUPA	CL0	CL1	CL2	CL3	CL4	CL5	CL6	CL7	CL8	CL9	CL10	CL11
DBSCAN	322	-	-	-	-	-	-	-	-	-	-	-
DENCLUE	99	205	41	-	-	-	-	-	-	-	-	-
MEANSHIFT	309	8	5	1	1	3	4	10	1	1	1	1
OPTICS	325	-	-	-	-	-	-	-	-	-	-	-
HDBSCAN	3	3	333	-	-	-	-	-	-	-	-	-

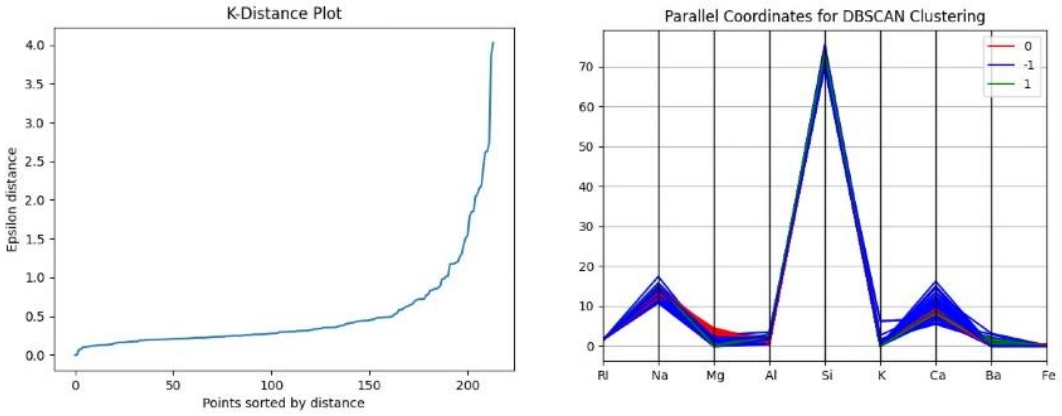
Πίνακας 3. Συνολικά Cluster του Bupa

4.1.4 Glass

Το Glass αποτελεί ένα σύνολο δεδομένων με πληροφορίες διαφόρων ειδών γυαλιών. Τα είδη χωρίζονται σύμφωνα με 9 διαφορετικά χαρακτηριστικά που αποτελούν τη σύνθεση τους, όπως είναι το νάτριο και το πυρίτιο. Το dataset αυτό περιέχει 214 δείγματα από γυαλιά που ανήκουν σε διαφορετικές κατηγορίες και δίνει τη δυνατότητα σε αλγορίθμους συσταδοποίησης να μπορούν εύκολα να ανιχνεύουν μοτίβα βάση αυτών.[20]

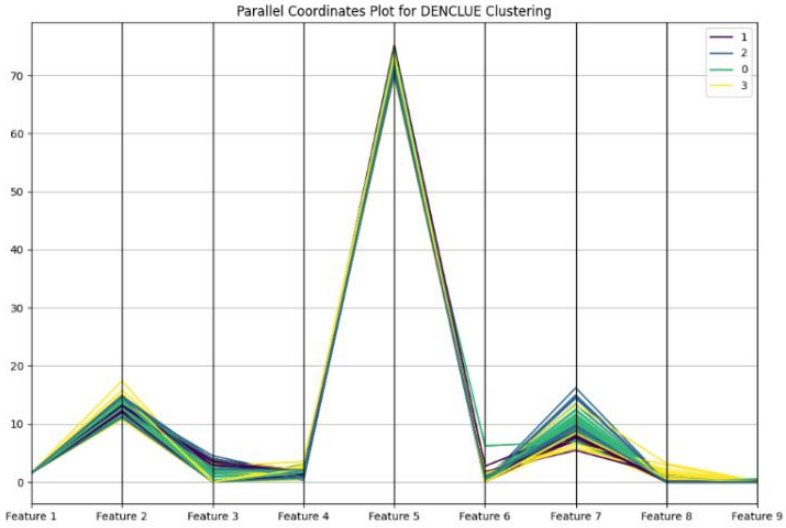
DBSCAN: dbscan = DBSCAN (eps=1.0, min_samples=7)

Πριν την εκτέλεση του DBSCAN για το Glass, ο προσδιορισμός της κατάλληλης ακτίνας εξέτασης ορίζεται από το k-distance graph που δείχνει τιμή περίπου 1.0. Ακολούθως, αφού ολοκληρωθεί η διαδικασία, στο διάγραμμα παράλληλων συντεταγμένων, είναι ορατή η ανίχνευση δύο συστάδων. Και οι δύο εξ αυτών, λαμβάνουν τις πιο υψηλές τιμές στη διάσταση Si ενώ ακολουθούν συγκεκριμένο μοτίβο με τα δεδομένα να αυξομειώνονται παρόμοια σχεδόν σε όλα τα χαρακτηριστικά. Τα δύο αυτά cluster, φαίνεται να επικαλύπτονται, κατάσταση η οποία δείχνει πιθανών μια μη ξεκάθαρη δομή στα δεδομένα και μπορεί εύκολα να δημιουργήσει προβλήματα στην ποιότητα του clustering του αλγορίθμου.



Εικόνα 4.21 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το Glass

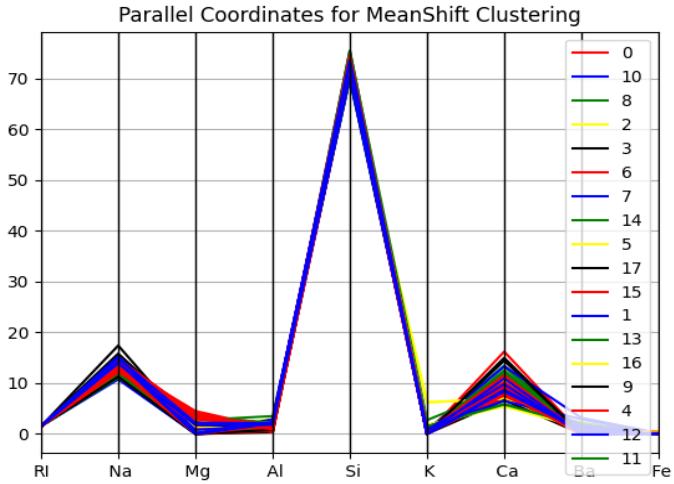
DENCLUE: denclue = DENCLUE (data_scaled, bandwidth=1.0, threshold=0.4)



Εικόνα 4.22 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το Glass

Ο DENCLUE με τη σειρά του δεν εντοπίζει κανένα σημείο θορύβου και ομαδοποιεί τα δεδομένα σε τέσσερις συστάδες που ακολουθούν κοινό μοτίβο. Οι μεγαλύτερες τιμές λαμβάνονται και πάλι στο Feature 5 ενώ ο διαχωρισμός τους δεν είναι ιδιαίτερα φανερός με τα χαρακτηριστικά Feature 3 και Feature 7 να είναι τα μοναδικά που δείχνουν αχνά τη διαφορά. Συνολικά το διάγραμμα οδηγεί στο συμπέρασμα ύπαρξης πιθανών αλληλεπικαλυπτόμενων ομάδων.

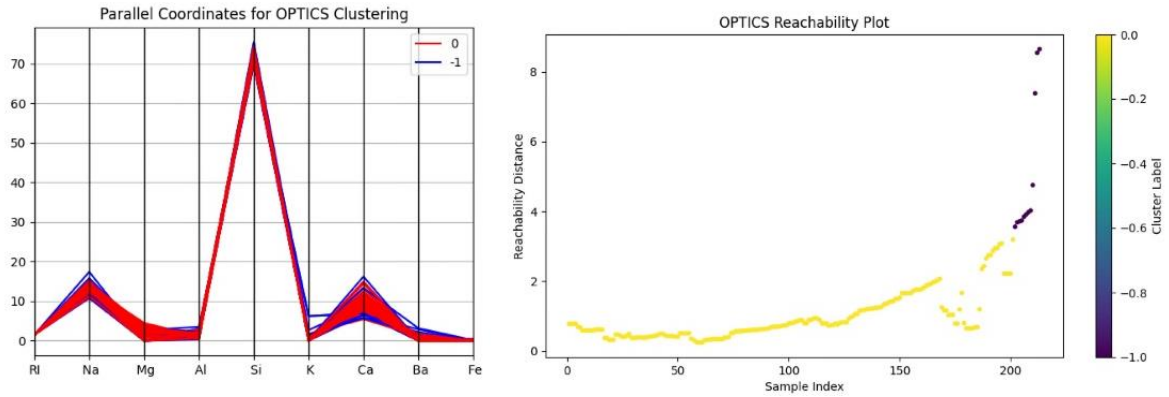
Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples=214)
Meanshift = Meanshift (bandwidth=bandwidth)



Εικόνα 4.23 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Glass

Από την άλλη πλευρά ο αλγόριθμος Mean Shift πάνω στο dataset Glass δημιουργεί πολλές μικρές συστάδες με μεγάλη πιθανότητα να χάνονται τα σημαντικότερα cluster. Επίσης, επειδή η λειτουργία του διαφέρει από τους υπόλοιπους αλγορίθμους και δεν εντοπίζει τα Noise points με τον ίδιο τρόπο, συστάδες πλήθους 1 βάσει του Πίνακα 4, πιθανών να απεικονίζουν ακραίες τιμές που λανθασμένα τοποθετήθηκαν σε συστάδες, γεγονός που χρήζει τον αλγόριθμο αυτόν ακατάλληλο για το συγκεκριμένο dataset.

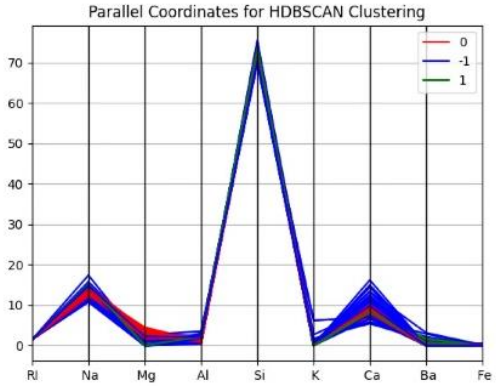
OPTICS: optics = OPTICS (min_samples =5, min_cluster_size=0.9)



Εικόνα 4.24 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICS για το Glass

Τα περισσότερα σημεία του χώρου των δεδομένων βάσει του OPTICS ορίζονται σε μια συστάδα με εξαίρεση εκείνα που αποτελούν θόρυβο. Το χαρακτηριστικό Si αποτελεί το σημαντικότερο εξ αυτών αφού παρουσιάζει τη μεγαλύτερη διακύμανση, με τα σημεία να ακολουθούν κοινή διαδρομή σε όλη την πορεία τους μεταξύ των γνωρισμάτων. Από την άλλη, στο reachability plot, ο θόρυβος εντοπίζεται έπειτα από το όριο που βρέθηκε λαμβάνοντας υψηλές τιμές αποστάσεων προσβασιμότητας και κάνοντας φανερή τη διαφορά που έχουν με την συστάδα και την κοιλάδα που δημιουργεί. Η περιοχή στην οποία η κοιλάδα σταματάει και επανέρχεται εντοπίζει σημεία της συστάδας τα οποία βρίσκονται πιο αραιά σε σχέση με τα υπόλοιπα αλλά λόγω της ρύθμισης των παραμέτρων, παραμένουν στην ίδια ομάδα.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size=8)



Εικόνα 4.25 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το Glass

Όπως διακρίνεται στην Εικόνα 4.25 ο αλγόριθμος HDBSCAN ανιχνεύει δύο συστάδες για το Glass με μεγαλύτερο ποσοστό του χώρου να φαίνεται πως καταλαμβάνουν τα σημεία θορύβου που δεν εντάχθηκαν σε καμία από τις δύο. Αυτό μπορεί να οφείλεται στην ακανόνιστη δομή των δεδομένων, σε λάθος της παραμετροποίησης, στις μη επαρκείς πληροφορίες των χαρακτηριστικών για τον σαφή διαχωρισμό των δεδομένων ή απλά ο αλγόριθμος να αποτελεί λάθος επιλογή για το συγκεκριμένο dataset. Η διάκριση μεταξύ τους παραμένει και πάλι μη εμφανής με τα cluster να δηλώνονται αρκετά πυκνά.

GLASS	DBSCAN	DENCLUE	MEANSHIFT	OPTICS	HDBSCAN
CL0	147	41	158	202	156
CL1	21	114	22	-	18
CL2	-	32	6	-	-
CL3	-	27	3	-	-

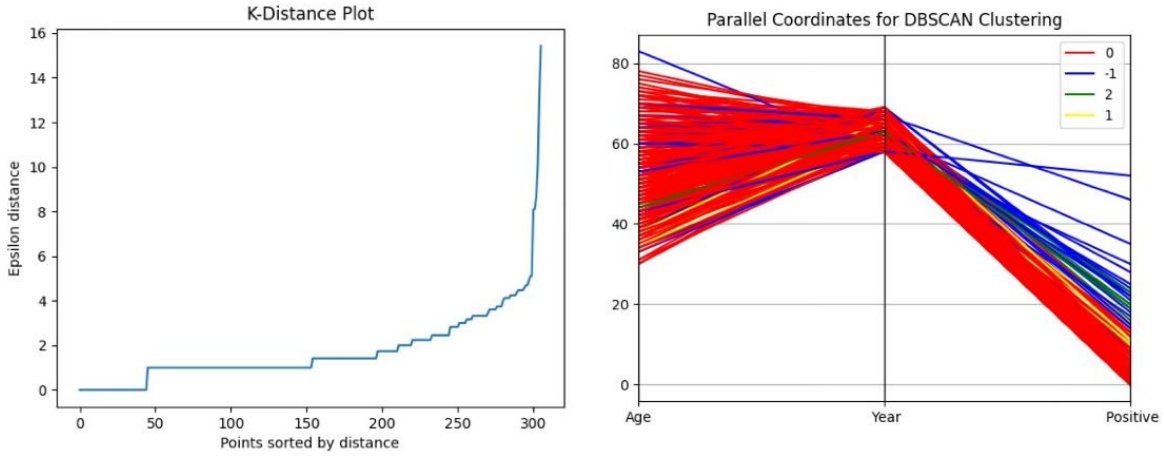
CL4	-	-	2	-	-
CL5	-	-	2	-	-
CL6	-	-	1	-	-
CL7	-	-	1	-	-
CL8	-	-	7	-	-
CL9	-	-	1	-	-
CL10	-	-	3	-	-
CL11	-	-	1	-	-
CL12	-	-	1	-	-
CL13	-	-	1	-	-
CL14	-	-	1	-	-
CL15	-	-	2	-	-
CL16	-	-	1	-	-
CL17	-	-	1	-	-

Πίνακας 4. Συνολικά Cluster του Glass

4.1.5 Haberman

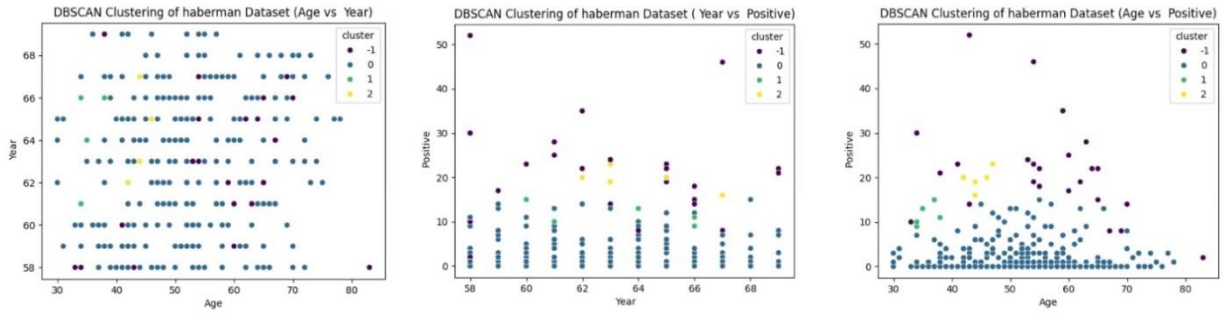
Άλλο ένα σύνολο δεδομένων που αφορά κλάδο της Ιατρικής είναι το dataset Haberman το οποίο περιέχει πληροφορίες για την πρόβλεψη επιβίωσης ή μη ασθενών μετά από χειρουργική επέμβαση λόγω καρκίνου του μαστού. Τα χαρακτηριστικά που διαθέτει είναι 3 με ενδεικτικά την ηλικία του κάθε ανθρώπου από την ημέρα που διαγνώστηκε αλλά και άλλες χρήσιμες πληροφορίες που τον κατατάσσουν ως επιζών ή όχι σε ένα πλαίσιο 306 δειγμάτων ασθενών. Με τον τρόπο αυτό δημιουργείται ένα μοντέλο που εκτιμά τους παράγοντες επιβίωσης και μη.[20]

DBSCAN: dbscan = DBSCAN (eps=5.0, min_samples=5)



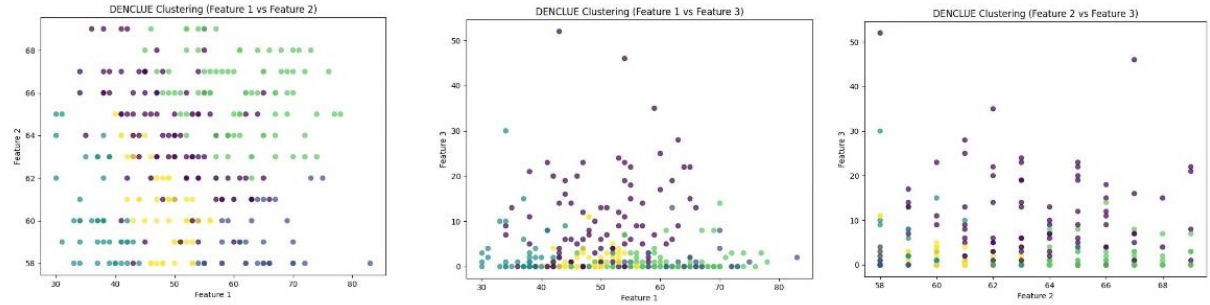
Εικόνα 4.26 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το Haberman

Προκειμένου το σωστό διαχωρισμό των δεδομένων σε clusters από τον DBSCAN, παρατηρώντας το k-distance plot, εντοπίζεται ο αγκώνας που δημιουργούν τα σημεία στην τιμή Eps=5.0. Ο αλγόριθμος ανιχνεύει τρείς συστάδες και εντοπίζει το θόρυβο. Και για τα τρία cluster, φαίνεται να υπάρχει ισχυρής συσχέτιση όταν αναφερόμαστε στα χαρακτηριστικά Year και Positive με τη διαφορά των συστάδων να είναι περισσότερο εμφανείς ενώ το χαρακτηριστικό Age δείχνει πως υπάρχει μεγάλη διασταύρωση των ομάδων μεταξύ τους που είναι φανερό και από τα scatter plot που δημιουργήθηκαν βάσει αυτού και εμφανίζονται στην Εικόνα 4.27 παρακάτω.

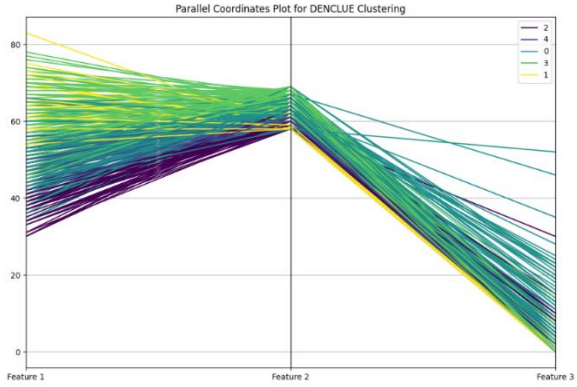


Εικόνα 4.27 Εξαγόμενα γραφήματα Scatter plot του DBSCAN για το Haberman

DENCLUE: denclue = DENCLUE (data_scaled, bandwidth=0.3, threshold=0.2)



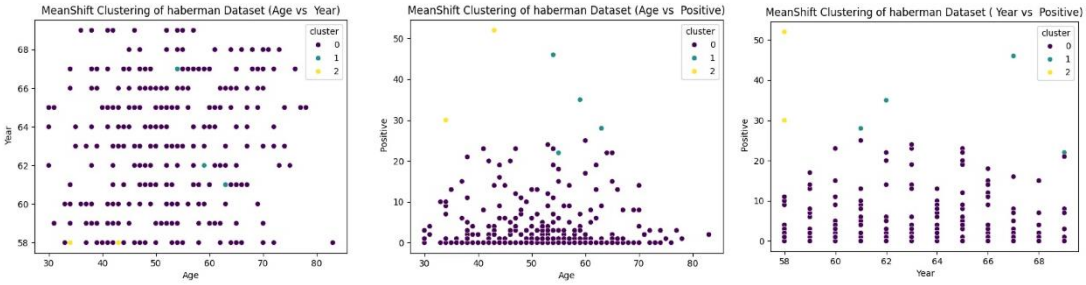
Εικόνα 4.28 Εξαγόμενα γραφήματα Scatter plot του DENCLUE για το Haberman



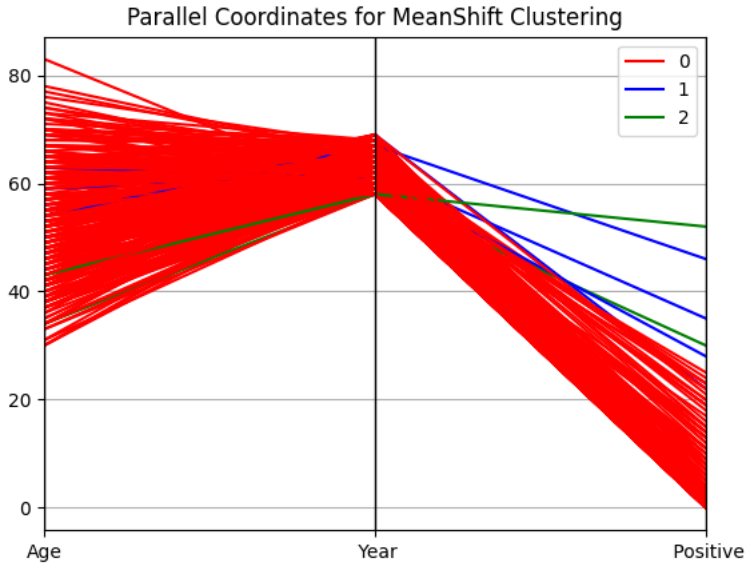
Εικόνα 4.29 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το Haberman

Ο αλγόριθμος DENCLUE καταφέρνει να μην εντοπίσει θόρυβο όμως και πάλι δημιουργούνται επικαλυπτόμενες πέντε συστάδες. Σύμφωνα με το scatter plot τα γνωρίσματα Feature 1 και Feature 2, που αντιπροσωπεύουν το Age και Year αντίστοιχα, φαίνεται να αποτελούν σημαντικά χαρακτηριστικά για το διαχωρισμό αφού τα τμήματα των συστάδων που εμφανίζονται δείχνουν τις διαφορετικές περιοχές που καταλαμβάνουν στο χώρο. Επιπλέον, ο διαχωρισμός των ομάδων αυτών γίνεται περισσότερο αντιληπτός στο Feature 1 αν και πάλι τα δεδομένα διασταυρώνονται σε πολλές τιμές μεταξύ τους.

Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples=306)
Meanshift = Meanshift (bandwidth=bandwidth)



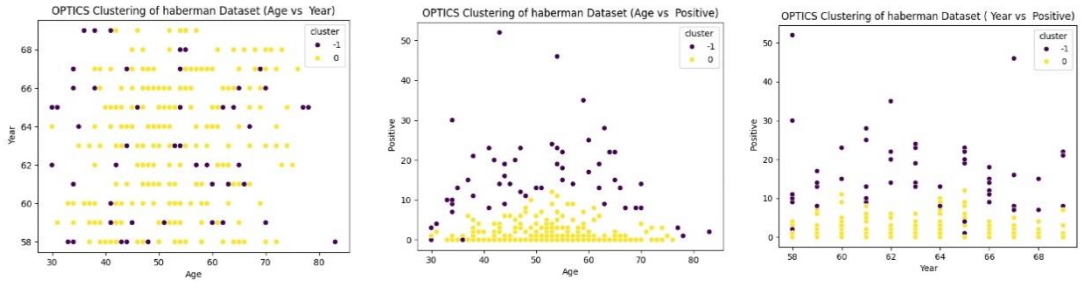
Εικόνα 4.30 Εξαγόμενα γραφήματα Scatter plot του Mean Shift για το Haberman



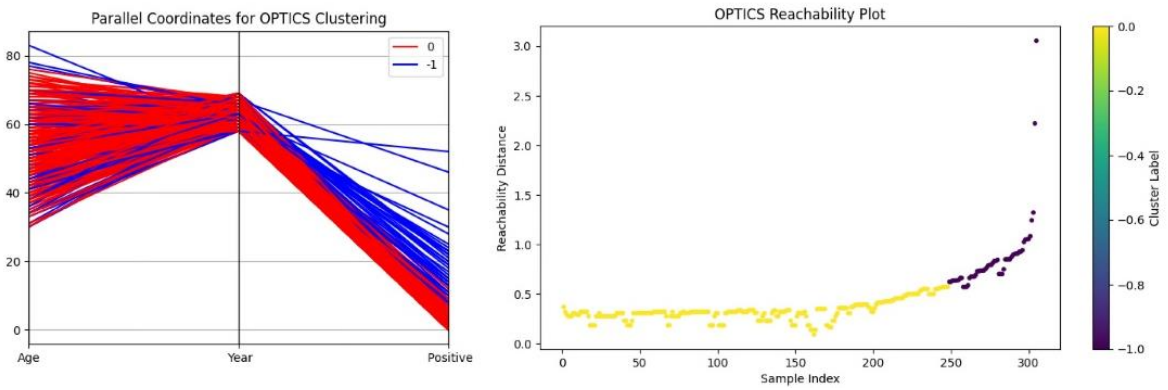
Εικόνα 4.31 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Haberman

Παρόλο που και ο Mean Shift εντοπίζει επικαλυπτόμενες συστάδες πιθανών λόγω δομής των δεδομένων, καταφέρνει να μην εντοπίσει σημεία θορύβου και τοποθετεί τα δείγματα σε τρεις ομάδες. Το cluster 0 μοιάζει να καταλαμβάνει τον περισσότερο χώρο και να είναι το πιο πυκνό όπως φαίνεται και στο διάγραμμα διασποράς αλλά και από τον Πίνακα 5. Τα χαρακτηριστικά Year και Age ενώ σε συνδυασμό με το Positive φαίνεται να λειτουργούν δείχνοντας ευδιάκριτα το διαχωρισμό μεταξύ των cluster, αν συνδυαστούν μεταξύ τους οι συστάδες συγχέονται. Τέλος, βάση του διαγράμματος parallel coordinates, συμπεραίνουμε ότι το χαρακτηριστικό Positive τείνει να είναι το καθοριστικό για τη συσταδοποίηση αφού εκεί τα σημεία κάθε ομάδας παίρνουν τιμές από διαφορετικά εύρη.

OPTICS: optics = OPTICS (min_samples =5, min_cluster_size=0.1)



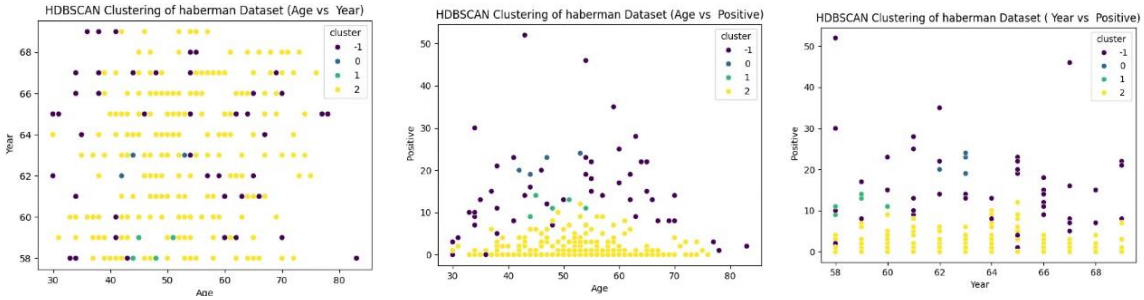
Εικόνα 4.32 Εξαγόμενα γραφήματα Scatter plot του OPTICS για το Haberman



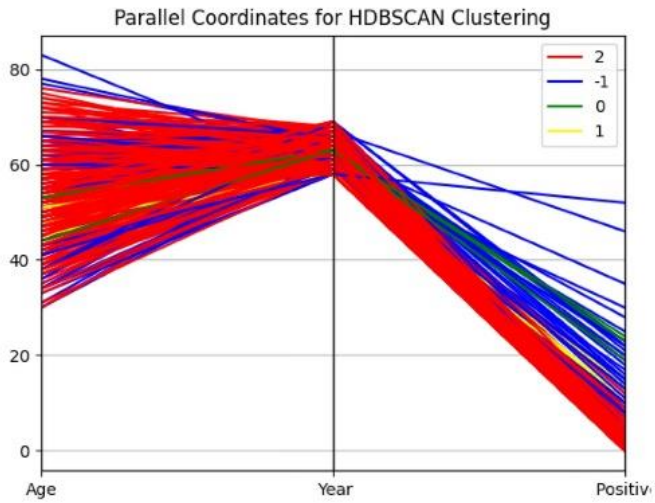
Εικόνα 4.33 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICS για το Haberman

Στον αλγόριθμο OPTICS για το dataset Haberman, η συσταδοποίηση φαίνεται να μην έχει πραγματοποιηθεί ορθά. Τα σημεία της μοναδικής συστάδας που ανιχνεύεται συγχέονται με τα σημεία θορύβου που αναγνωρίστηκαν. Οι outliers καταλαμβάνουν μεγάλο τμήμα του χώρου πιθανών από μη σωστή ενσωμάτωση αυτών στην ήδη υπάρχον συστάδα ή τα χαρακτηριστικά να μην επαρκούν για το σωστό διαχωρισμό. Παρατηρώντας και το διάγραμμα προσβασιμότητας, η συστάδα ακολουθεί μια πορεία κοντινών τιμών αποστάσεων ενώ τη στιγμή που αλγόριθμος όρισε το όριο και τη διακοπή της δεν απέχει ιδιαίτερα για το διαχωρισμό των επερχόμενων σημείων από αυτήν.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size=3)



Εικόνα 4.34 Εξαγόμενα γραφήματα Scatter plot του HDBSCAN για το Haberman



Εικόνα 4.35 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το Haberman

Ομοίως στην εκτέλεση του HDBSCAN, το χαρακτηριστικό Positive χρήζει σημαντικότητας αφού οι τρείς συστάδες που αναγνωρίζονται είναι περισσότερο εμφανής εκεί ενώ τα scatter plot της Εικόνας 4.34 δείχνουν τη σχέση των χαρακτηριστικών Age και Year ξεχωριστά σε σχέση με το Positive όπου το τμήμα που καταλαμβάνει κάθε μια στο χώρο είναι ξεκάθαρο.

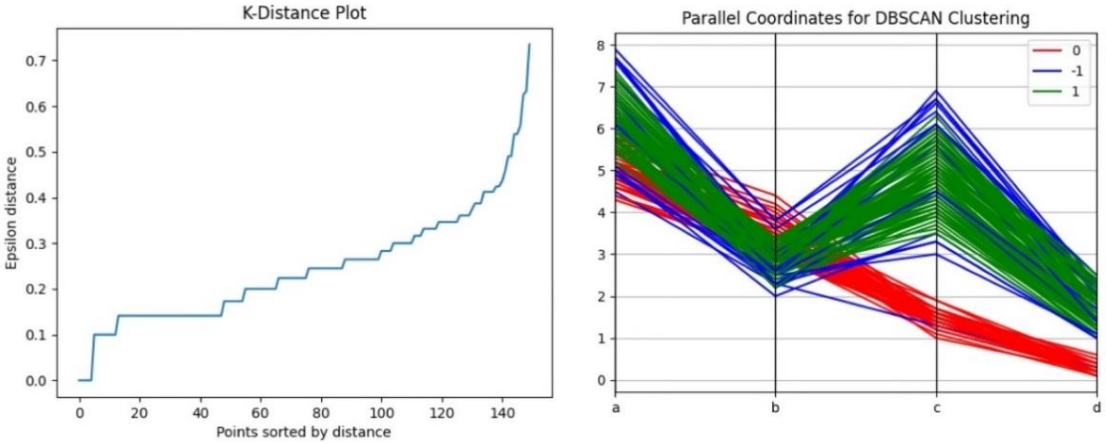
HABERMAN	CL0	CL1	CL2	CL3	CL4
DBSCAN	272	5	5	-	-
DENCLUE	93	35	41	84	53
MEANSHIFT	300	4	2	-	-
OPTICS	249	-	-	-	-
HDBSCAN	4	6	246	-	-

Πίνακας 5. Συνολικά Cluster του Haberman

4.1.6 Iris

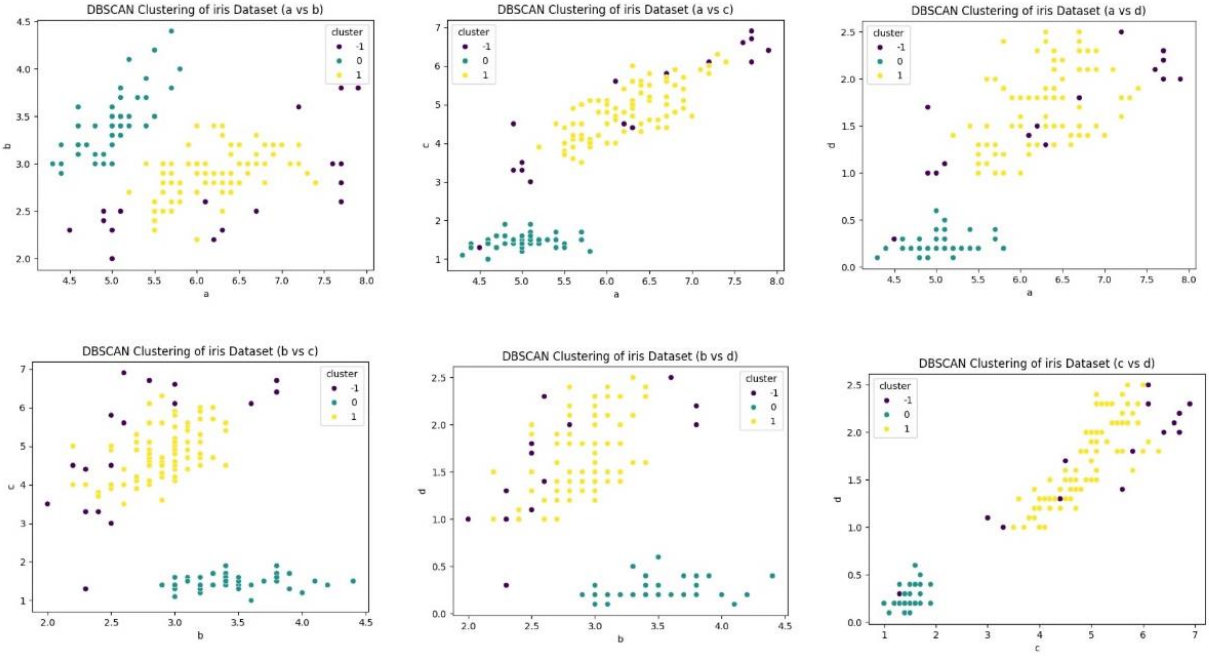
Ένα από τα πιο διαδεδομένα σύνολα δεδομένων στη μηχανική μάθηση είναι το Iris αφού λόγο της απλότητας του περιεχομένου του, χρησιμοποιείται συχνά για εκπαιδευτικούς σκοπούς ώστε να γίνει κατανοητή η χρήση των αλγορίθμων. Περιέχει ένα πλήθος 150 δειγμάτων λουλουδιών Ιρίδων τα οποία ανάλογα τα χαρακτηριστικά τους χωρίζονται σε τρείς κατηγορίες iris-setosa, iris-versicolor και iris-virginica. Κάθε δείγμα κατηγοριοποιείται βάση 4 χαρακτηριστικών που αφορούν τις φυσικές τους διαστάσεις όπως το μήκος και το πλάτος του πετάλου και χρησιμοποιούνται στην συσταδοποίηση ομαδοποιώντας τα κοινά.[20]

DBSCAN: dbscan = DBSCAN (eps= 0.5, min_samples=5)



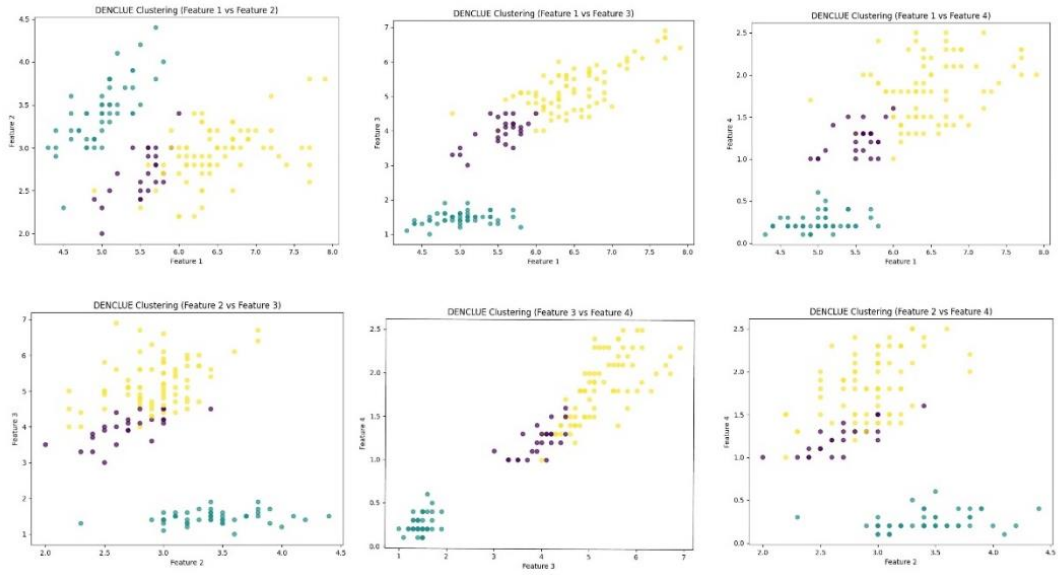
Εικόνα 4.36 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το Iris

Ξεκινώντας από τον αλγόριθμο DBSCAN, στο dataset Iris, φαίνεται πως επιτυγχάνει ορθή συσταδοποίηση αφού σε όλους τους πιθανούς συνδυασμούς χαρακτηριστικών, όπως φαίνεται και στα παραγόμενα scatter plot της Εικόνας 4.37, ο διαχωρισμός των συστάδων είναι ξεκάθαρος. Γεγονός που σημαίνει ότι η επιλογή των παραμέτρων ήταν η κατάλληλη τόσο του Eps=5.0 που εντοπίζεται στο k-distance plot αλλά και του min_samples που επιλέχθηκε. Επιπλέον, το cluster 0 φαίνεται να είναι αυτό με τη μεγαλύτερη ομοιογένεια αφού οι τιμές των σημείων τείνουν να μοιάζουν για κάθε χαρακτηριστικό χωρίς μεγάλες αυξομειώσεις ενώ το cluster 1 δημιουργεί επίσης ένα μοτίβο. Η διαφορά μεταξύ τους είναι εμφανής περισσότερα στα χαρακτηριστικά c και d από όπου και λαμβάνουν τιμές από διαφορετικές περιοχές τιμών.



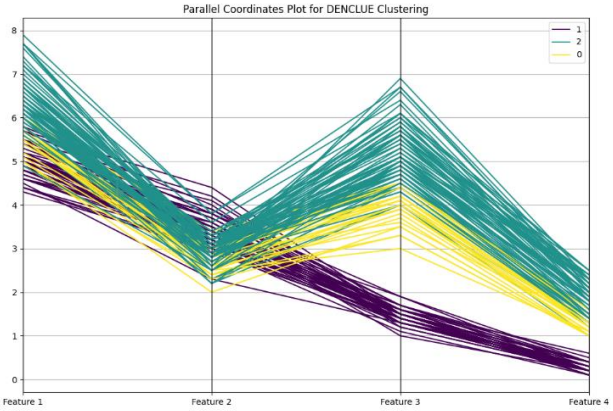
Εικόνα 4.37 Εξαγόμενα γραφήματα Scatter plot του DBSCAN για το Iris

DENCLUE: denclue = DENCLUE (data_scaled, bandwidth=0.5, threshold=0.3)



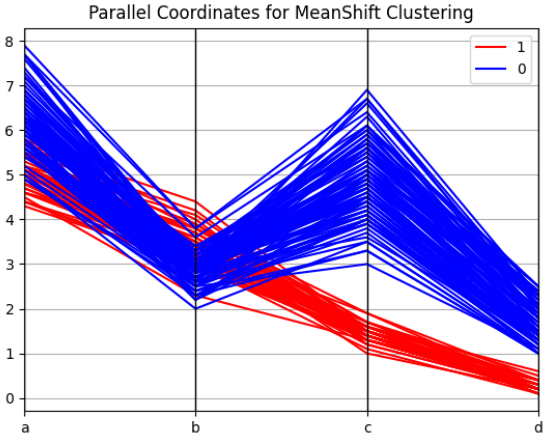
Εικόνα 4.38 Εξαγόμενα γραφήματα Scatter plot του DENCLUE για το Iris

Ταυτόχρονα και ο αλγόριθμος DENCLUE αποδεικνύει πως για το σύνολο δεδομένων αυτό εκτελείται ομαλά χωρίς την ανίχνευση θορύβου και με τον εντοπισμό τριών συστάδων. Σε κάθε τμήμα της ομαδοποίησης που εμφανίζεται στα scatter plot, η διαφορά τουλάχιστον των δύο εξ αυτών είναι ξεκάθαρη χρήζοντας τα χαρακτηριστικά σημαντικά για τη συσταδοποίηση. Ακολούθως, το διάγραμμα παράλληλων συντεταγμένων δείχνει το μονοπάτι καθενός, με μεγαλύτερη ομοιογένεια και χωρίς διακυμάνσεις να έχει το cluster 1 ενώ τα γνωρίσματα Feature 3 και 4 μοιάζουν να είναι αυτά που κάνουν τη διάκριση μεταξύ τους περισσότερο εμφανή.

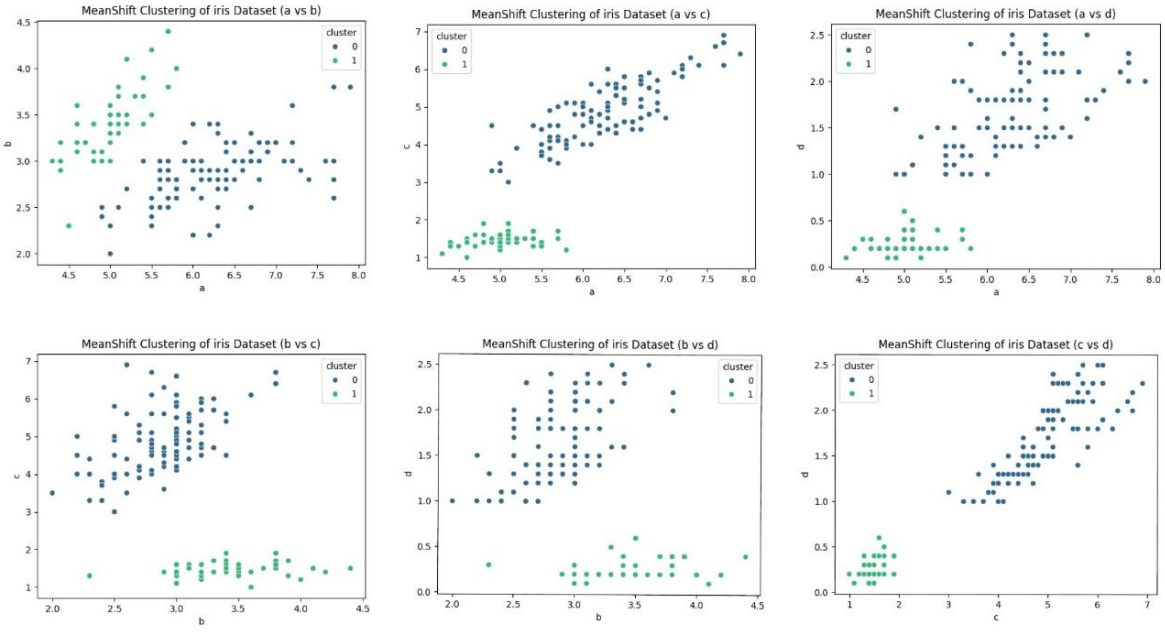


Εικόνα 4.39 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το Iris

Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples=150)
Meanshift = Meanshift (bandwidth=bandwidth)



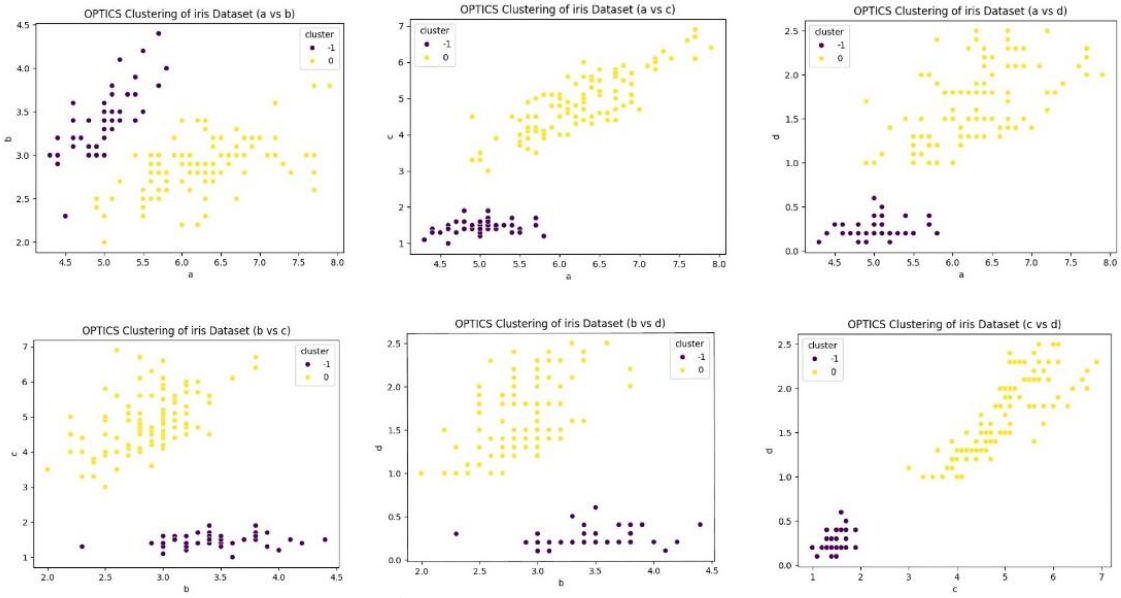
Εικόνα 4.40 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Iris



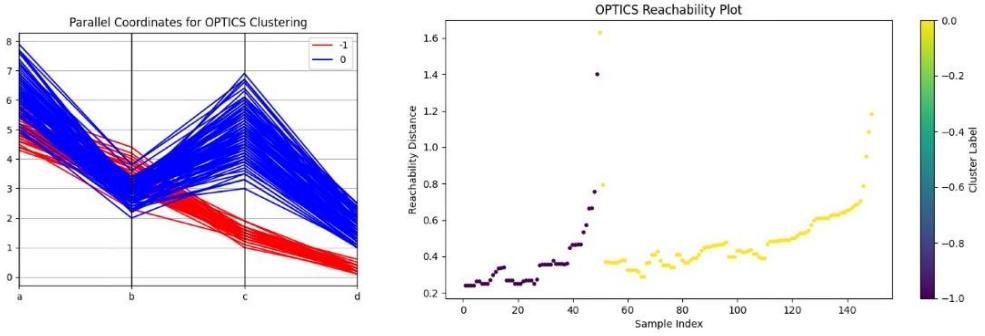
Εικόνα 4.41 Εξαγόμενα γραφήματα Scatter plot του Mean Shift για το Iris

Και αλγόριθμος Mean Shift κατορθώνει με τη σειρά του να ομαδοποιήσει όλα τα δεδομένα σε συστάδες. Από την εκτέλεση του δημιουργούνται δύο συστάδες που η κάθε μία ακολουθεί τη δική της διαδρομή. Τα χαρακτηριστικά που διαθέτει το dataset φαίνεται να είναι επαρκείς και η συσταδοποίηση γίνεται με επιτυχία αφού κάθε συνδυασμός αυτών απεικονίζει δύο συμπαγής και καλά ορισμένες συστάδες με εμφανή απόσταση μεταξύ τους ενώ και σε αυτή την περίπτωση τα δύο πιο σημαντικά χαρακτηριστικά είναι το c και d, όπου κάθε σημείο της συστάδας λαμβάνει τιμές από συγκεκριμένη κλίμακα.

OPTICS: optics = OPTICS (min_samples =5, min_cluster_size=0.6)



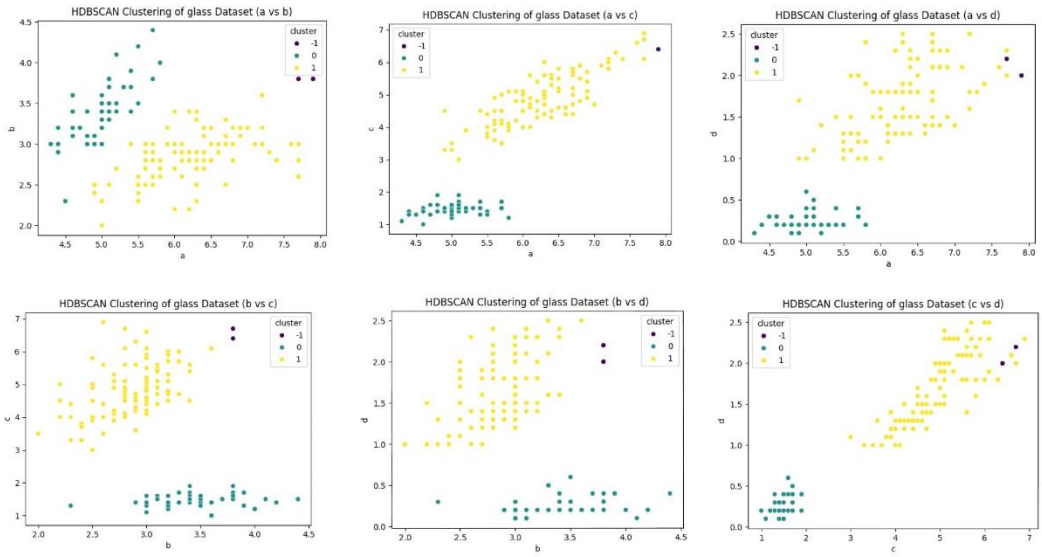
Εικόνα 4.42 Εξαγόμενα γραφήματα Scatter plot του OPTICS για το Iris



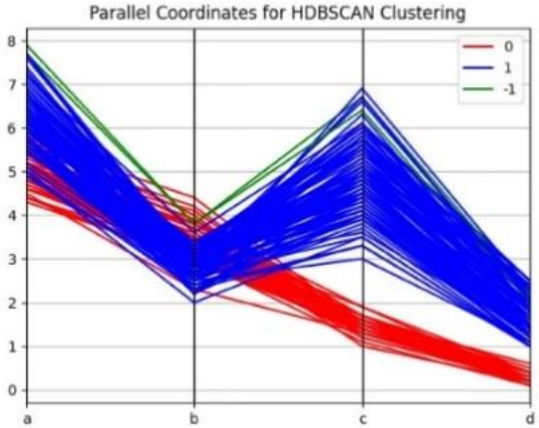
Εικόνα 4.43 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICS για το Iris

Αντίθετα με τους προηγούμενους αλγορίθμους στους οποίους εκτελέστηκε το σύνολο δεδομένων Iris, ο OPTICS εντοπίζει μόνο μία συστάδα η οποία όμως βρίσκεται σε μεγάλη απόσταση από τα απομονωμένα σημεία θορύβου δρώντας επιτυχημένα στη συσταδοποίηση. Ο διαχωρισμός της από το θόρυβο είναι εμφανής και στα διαγράμματα διασποράς που δημιουργήθηκαν. Παρόλα αυτά, ενώ και στο reachability plot γίνεται ξεκάθαρο το όριο από όπου και ξεκινάει το cluster διακρίνονται ακραίες τιμές απόστασης προσβασιμότητας για σημεία που βρίσκονται πιο απομακρυσμένα από το σύνολο και πιθανών να ενσωματώθηκαν λανθασμένα σε αυτό.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size= 4)



Εικόνα 4.44 Εξαγόμενα γραφήματα Scatter plot του HDBSCAN για το Iris



Εικόνα 4.45 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το Iris

Παράλληλα, ο HDBSCAN με τον τρόπου που διαχειρίζεται τα δεδομένα, ανίχνευσε πάνω στο ίδιο σύνολο δύο συστάδες και κατόρθωσε να εξαλείψει σχεδόν τους outliers. Η διάκριση μεταξύ των cluster είναι εμφανής τόσο στα χαρακτηριστικά c και d όσο και σε κάθε συνδυασμό μεταξύ τους δείχνοντας ότι όλα υπήρξαν σημαντικά για τη συσταδοποίηση.

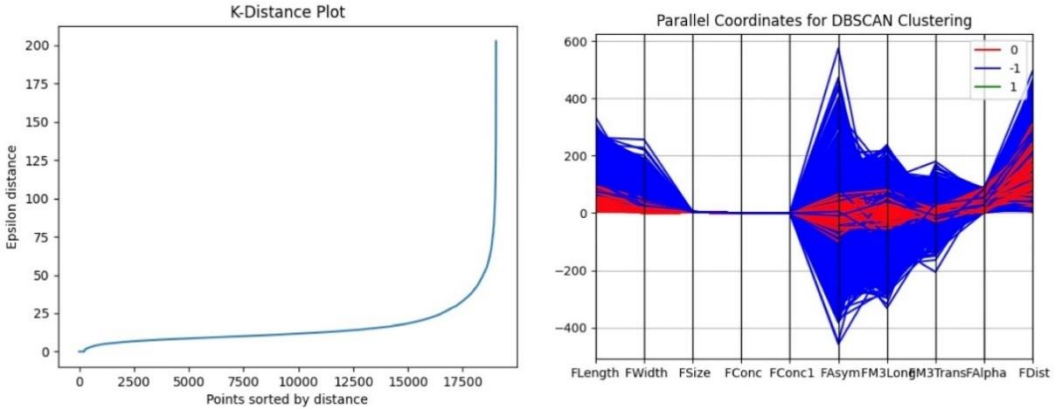
IRIS	CL0	CL1	CL2
DBSCAN	49	84	-
DENCLUE	25	50	75
MEANSHIFT	100	50	-
OPTICS	100	-	-
HDBSCAN	50	98	-

Πίνακας 6. Συνολικά Cluster του Iris

4.1.7 Magic Gamma Telescope

Ως μεσαίου μεγέθους θα μπορούσε να χαρακτηριστεί ένα dataset όπως το Magic Gamma Telescope το οποίο περιλαμβάνει 19.020 παρατηρήσεις από φωτεινά ίχνη που αναγνωρίστηκαν μέσω του τηλεσκοπίου MAGIC (Major Atmospheric Gamma Imaging Cherenkov Telescope) από όπου πήρε και το όνομα του. Τα σήματα αυτά κατηγοριοποιούνται βάσει 10 διαφορετικών χαρακτηριστικών, όπως η συγκέντρωση και η ένταση του φωτός και διακρίνονται σε σήματα που παράγονται από ακτίνες ή άλλα φυσικά αίτια.[20]

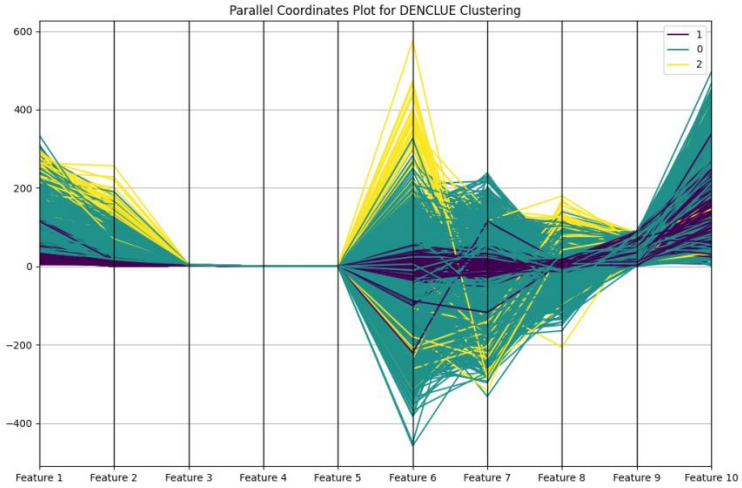
DBSCAN: dbscan = DBSCAN (eps=30.0, min_samples=8)



Εικόνα 4.46 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το Magic

Για το σύνολο δεδομένων MAGIC, ο αλγόριθμος DBSCAN φαίνεται να εντοπίζει δύο συστάδες μη καλά διαχωρισμένες. Παρόλο που η τιμή του Eps επιλέχθηκε από το k-distance plot, η ακτίνα και πάλι φαίνεται να μη δίνει σαφή αποτελέσματα. Τα σημεία θορύβου καλύπτουν το μεγαλύτερο μέρος του γραφήματος γεγονός που σημαίνει ότι ο αλγόριθμος DBSCAN δεν κατάφερε με επιτυχία να αναγνωρίσει τις πυκνές περιοχές ή τα σημεία απέχουν αρκετά μεταξύ τους ώστε αν ενσωματωθούν και το cluster 1 που φέρει το πράσινο χρώμα επικαλύπτεται πλήρως. Ενδεχομένως τα χαρακτηριστικά του dataset να μην επαρκούν για την ορθή συσταδοποίηση ή ο αλγόριθμος να μην είναι ο κατάλληλος για αυτό.

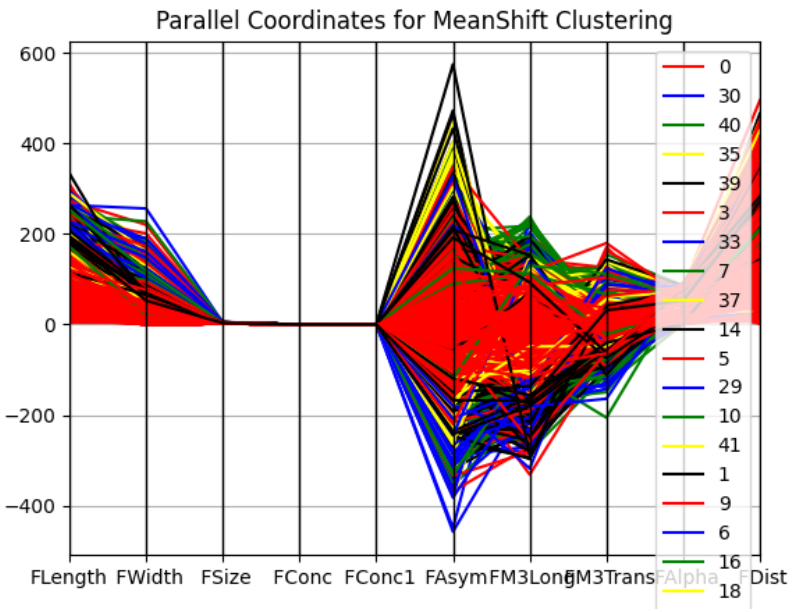
DENCLUE: denclue = DENCLUE (data_scaled, bandwidth= 3.0, threshold=1.2)



Εικόνα 4.47 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Magic

Με τη δική του σειρά, ο DENCLUE ομαδοποιεί τα δεδομένα σε τρείς συστάδες. Το μεγαλύτερο εύρος τιμών εντοπίζεται στο Feature 6 όπου και παίρνει τις υψηλότερες αλλά και χαμηλότερες τιμές και αποτελεί το σημαντικότερο χαρακτηριστικό για το διαχωρισμό των συστάδων μαζί με το Feature 2. Ενώ το cluster 1 φαίνεται να είναι αυτό με τη μεγαλύτερη ομοιογένεια, κανένα από τα τρία δε δημιουργεί κάποιο μοτίβο, με τις διασταυρώσεις μεταξύ τους να είναι αρκετές.

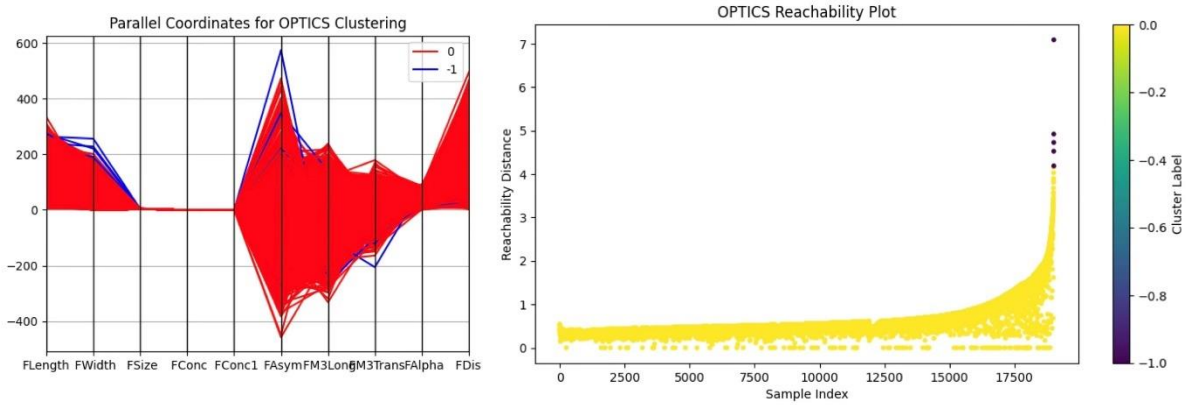
Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples=19020)
Meanshift = Meanshift (bandwidth=bandwidth)



Εικόνα 4.48 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Magic

Ο αλγόριθμος Mean Shift αν και εξασφαλίζει ομαδοποίηση των σημείων χωρίς την εμφάνιση θορύβου, ωστόσο τα cluster που εντοπίζει, φαίνεται να είναι πολλά και υπερβολικά μικρά γεγονός που κάνει τη μέθοδο συσταδοποίησης αυτή ακατάλληλη για το συγκεκριμένο σύνολο. Με την υπερβολική υποδιαίρεση των δεδομένων, δημιουργείται σύγχυση στην αναγνώριση των πραγματικών διαφορών μεταξύ αυτών ενώ η δημιουργία συστάδων με πλήθος 1 όπως φαίνεται από τον Πίνακα 7, δείχνει πως τα σημεία αυτά αποτελούν ακραίες τιμές και λανθασμένα ανιχνεύθηκαν ως συστάδες ενώ θα έπρεπε να οριστούν ως θόρυβος.

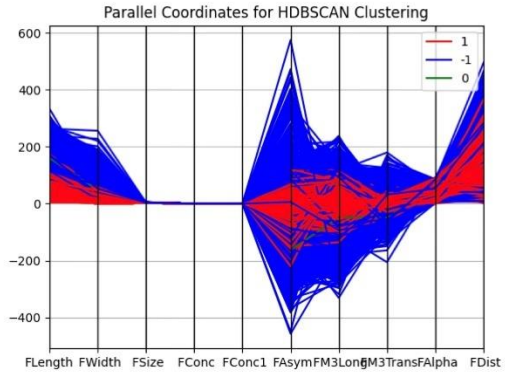
OPTICS: optics = OPTICS (min_samples = 2, min_cluster_size=1.0)



Εικόνα 4.49 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICS για το Magic

Εκτελώντας τον OPTICS στα δεδομένα του MAGIC, τα σημεία εντάσσονται σε μία μόνο συστάδα διαχωρίζοντας μόνο ελάχιστα από αυτά ως θόρυβο. Στο reachability plot δημιουργείται μια αρκετά μεγάλη κοιλάδα που δείχνει τις κοντινές αποστάσεις όμως όταν οι τιμές αρχίζουν να αυξάνονται δεν διαχωρίζονται και συνεχίζουν να παραμένουν εντός της ομάδας. Η κίνηση αυτή πιθανών προήλθε από λανθασμένο ή μη ορισμό ορίου. Όμως, όπως και η ακραία διαίρεση έτσι και η υπερβολική συγχώνευση οδηγεί σε μια μη αντιπροσωπευτική εικόνα των δεδομένων.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size= 8)



Εικόνα 4.50 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το Magic

Η συσταδοποίηση που πραγματοποίησε ο αλγόριθμος HDBSCAN, τείνει να μοιάζει με αυτή του DBSCAN (Εικόνα 4.46), με την μια εκ των δύο συστάδων που δημιουργούνται να μην είναι εμφανής στο διάγραμμα. Το cluster 1 είναι εμφανώς πιο μικρό από το cluster 0, όπως φαίνεται και από τον Πίνακα 7, γεγονός που οδηγεί στην μη εμφάνιση του στο γράφημα. Ωστόσο, η συσταδοποίηση δε πραγματοποιείται με επιτυχία αφού ο θόρυβος καλύπτει το μεγαλύτερο εύρος παραλείποντας πιθανές χρήσιμες πληροφορίες.

MAGIC	DBSCAN	DENCLUE	MEANSHIFT	OPTICS	HDBSCAN
CL0	16086	7519	17751	19015	14
CL1	9	11422	17	-	17349
CL2	-	79	32	-	-
CL3	-	-	128	-	-
CL4	-	-	7	-	-
CL5	-	-	157	-	-
CL6	-	-	42	-	-
CL7	-	-	42	-	-
CL8	-	-	2	-	-
CL9	-	-	26	-	-
CL10	-	-	3	-	-
CL11	-	-	21	-	-
CL12	-	-	1	-	-
CL13	-	-	1	-	-
CL14	-	-	156	-	-
CL15		-	1	-	-
CL16	-	-	43	-	-
CL17	-	-	1	-	-
CL18	-	-	1	-	-
CL19	-	-	11	-	-
CL20	-	-	1	-	-
CL21	-	-	6	-	-
CL22	-	-	1	-	-
CL23	-	-	2	-	-
CL24	-	-	1	-	-
CL25	-	-	18	-	-
CL26	-	-	1	-	-
CL27	-	-	2	-	-

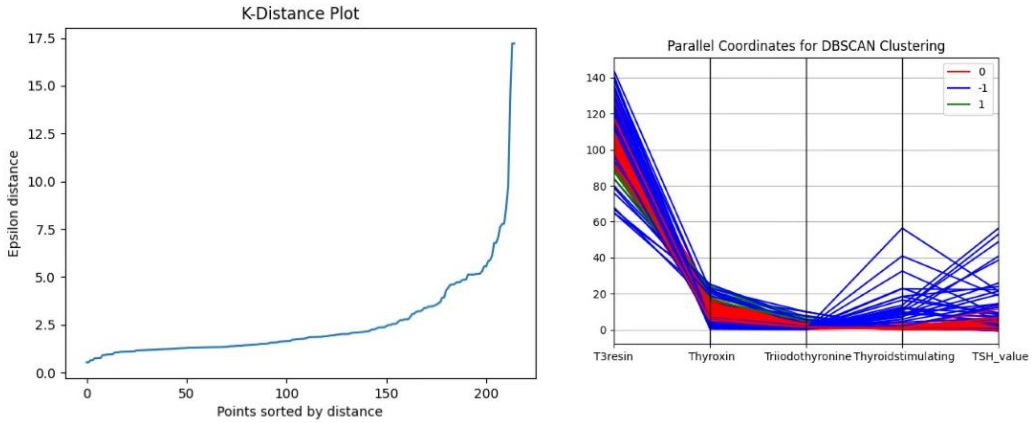
CL28	-	-	12	-	-
CL29	-	-	1	-	-
CL30	-	-	25	-	-
CL31	-	-	1	-	-
CL32	-	-	1	-	-
CL33	-	-	24	-	-
CL34	-	-	57	-	-
CL35	-	-	151	-	-
CL36	-	-	1	-	-
CL37	-	-	9	-	-
CL38	-	-	8	-	-
CL39	-	-	125	-	-
CL40	-	-	45	-	-

Πίνακας 7. Συνολικά Cluster του MAGIC

4.1.8 New Thyroid

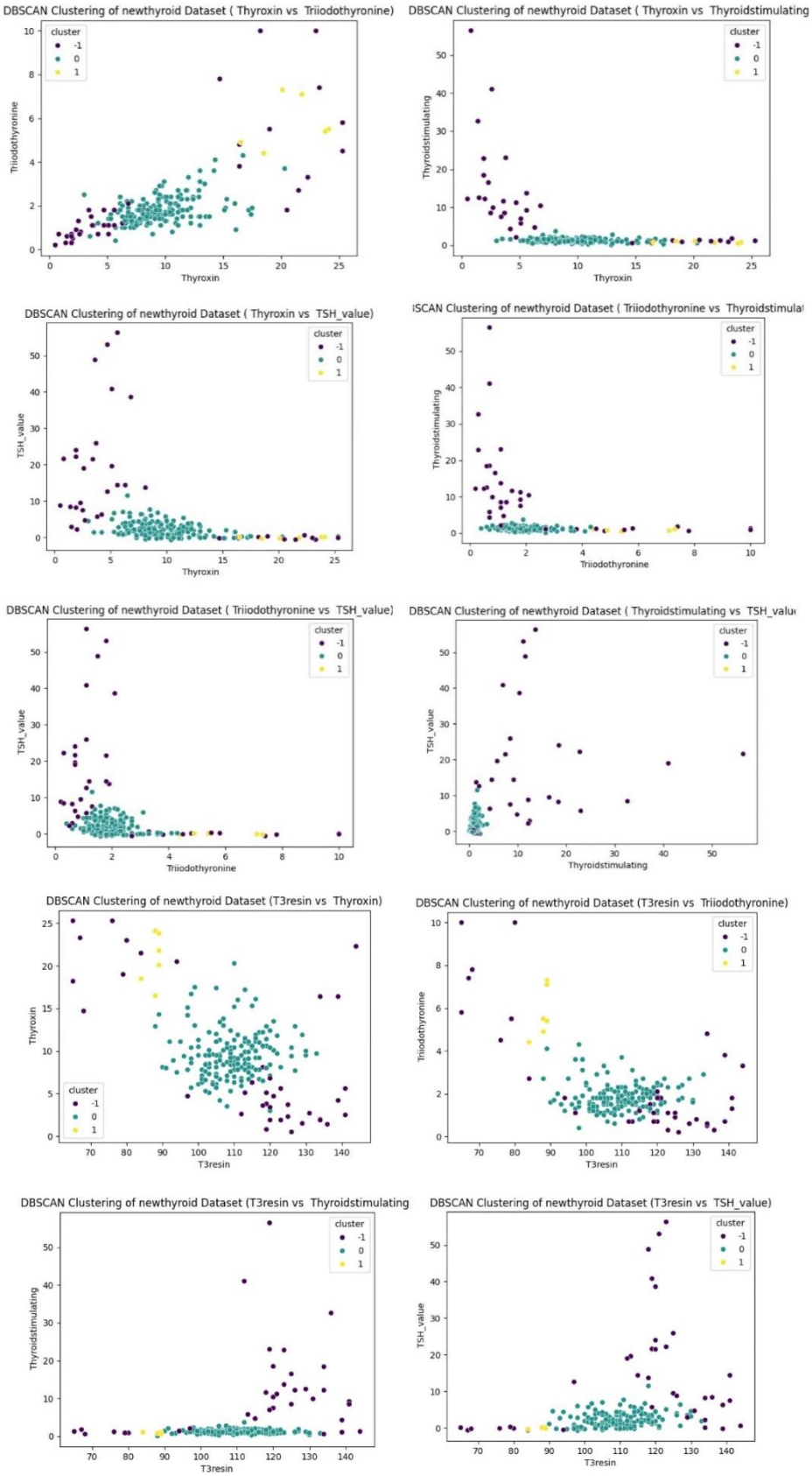
Το σύνολο δεδομένων New Thyroid εξάγεται και αυτό από τον κλάδο της υγείας και αφορά τη διάγνωση παθήσεων που σχετίζονται με το θυρεοειδή. Η κατηγοριοποίηση πραγματοποιείται μέσα από ένα σύνολο 215 δειγμάτων ανθρώπων με 5 διαφορετικά χαρακτηριστικά όπως η ηλικία, το φύλο και άλλα σχετιζόμενα με την κλινική τους εικόνα στοιχεία. Αποτελεί χρήσιμο εργαλείο όταν υπάρχει ανάγκη δημιουργίας ενός μοντέλου για τη διάγνωση τέτοιων προβλημάτων. Στην περίπτωση όμως της συσταδοποίησης τα δεδομένα ομαδοποιούνται με βάση τα κοινά τους στοιχεία και όχι κατηγοριών.[20]

DBSCAN: dbscan = DBSCAN (eps=5.0, min_samples=5)



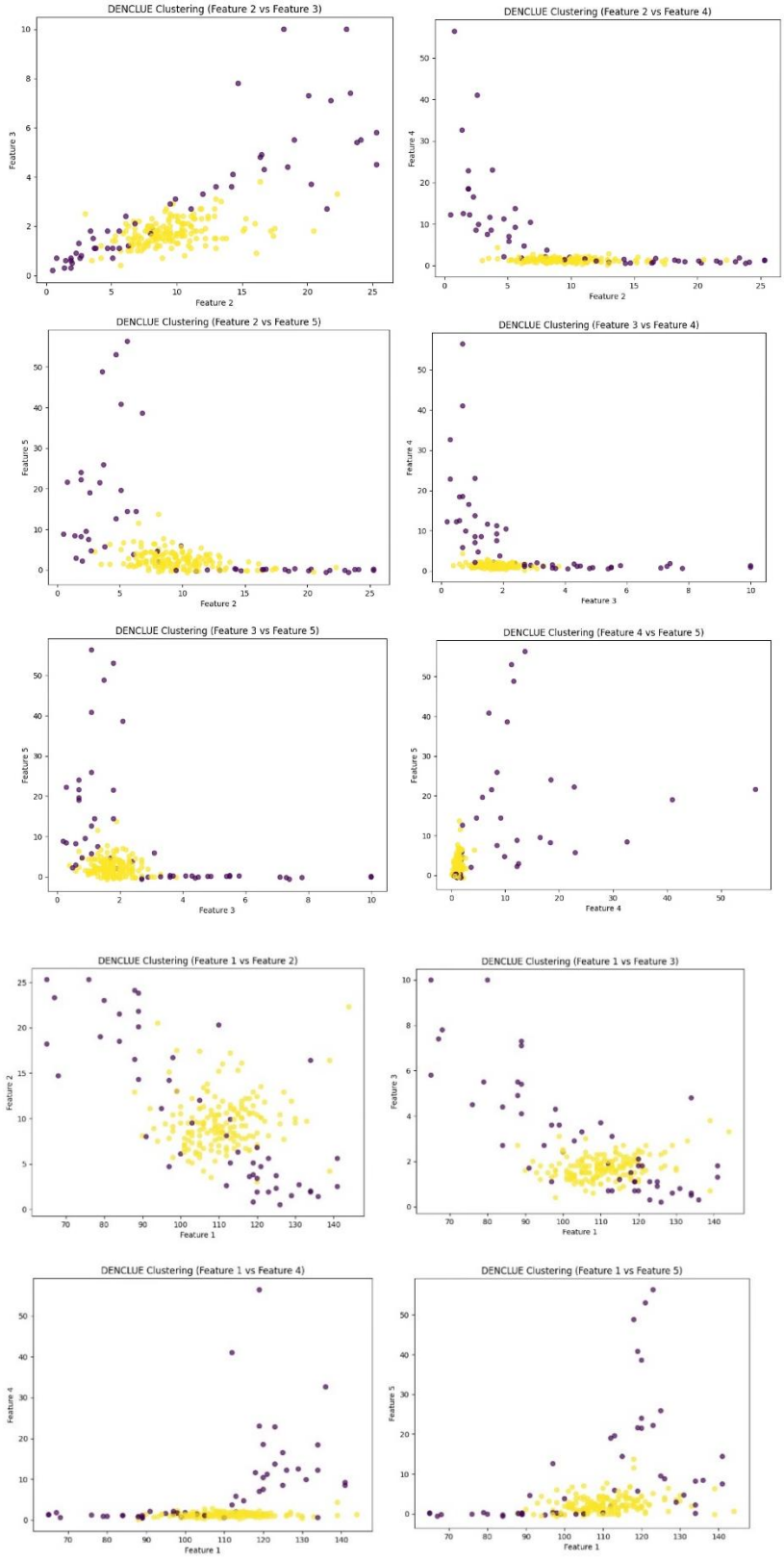
Εικόνα 4.51 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το New Thyroid

Αρχίζοντας την συσταδοποίηση των δεδομένων για το dataset New Thyroid με τη χρήση του αλγορίθμου DBSCAN και εντοπίζοντας την ιδανική τιμή για την ακτίνα Eps στο διάγραμμα k-distance graph, δημιουργούνται δύο συστάδες. Οι υψηλότερες τιμές λαμβάνονται στο πρώτο χαρακτηριστικό T3resin και μειώνονται ταυτόχρονα όλα στο επόμενο. Η πορεία προς τα υπόλοιπα χαρακτηριστικά συνεχίζει με παρόμοιες τιμές σε όλα χωρίς κάποια διακύμανση. Και τα δυο cluster δημιουργούν μοτίβα με το διαχωρισμό τους να είναι εμφανέστερος μόνο στο πρώτο χαρακτηριστικό του parallel coordinates ενώ από τα scatter plot παρακάτω (Εικόνα 4.52) συμπαιρένουμε τη διαφορά αυτών πιο έντονα με μικρή επικάλυψη σε ορισμένα σημεία. Ιδιαίτερα οι σχέσεις T3resin-Thyroxin και T3resin-Triiodothyronine φαίνεται να είναι αυτές που δείχνουν εμφανέστερα την απόσταση των διαφορετικών ομάδων.



Εικόνα 4.52 Εξαγόμενα γραφήματα Scatter plot του DBSCAN για το New Thyroid

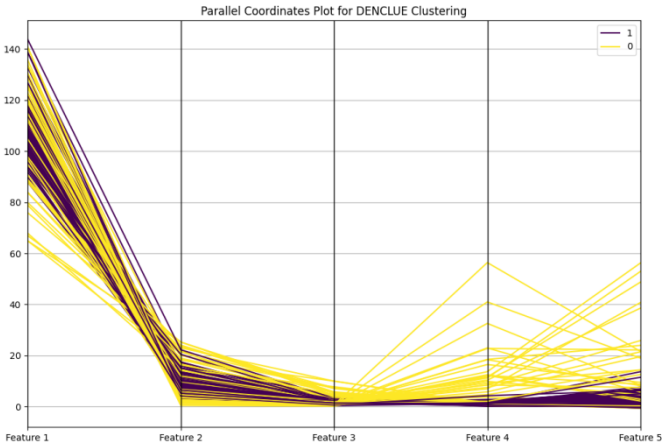
DENCLUE: denclue = DENCLUE (data_scaled, bandwidth=2.0, threshold=0.7)



Εικόνα 4.53 Εξαγόμενα γραφήματα Scatter plot του DENCLUE για το New Thyroid

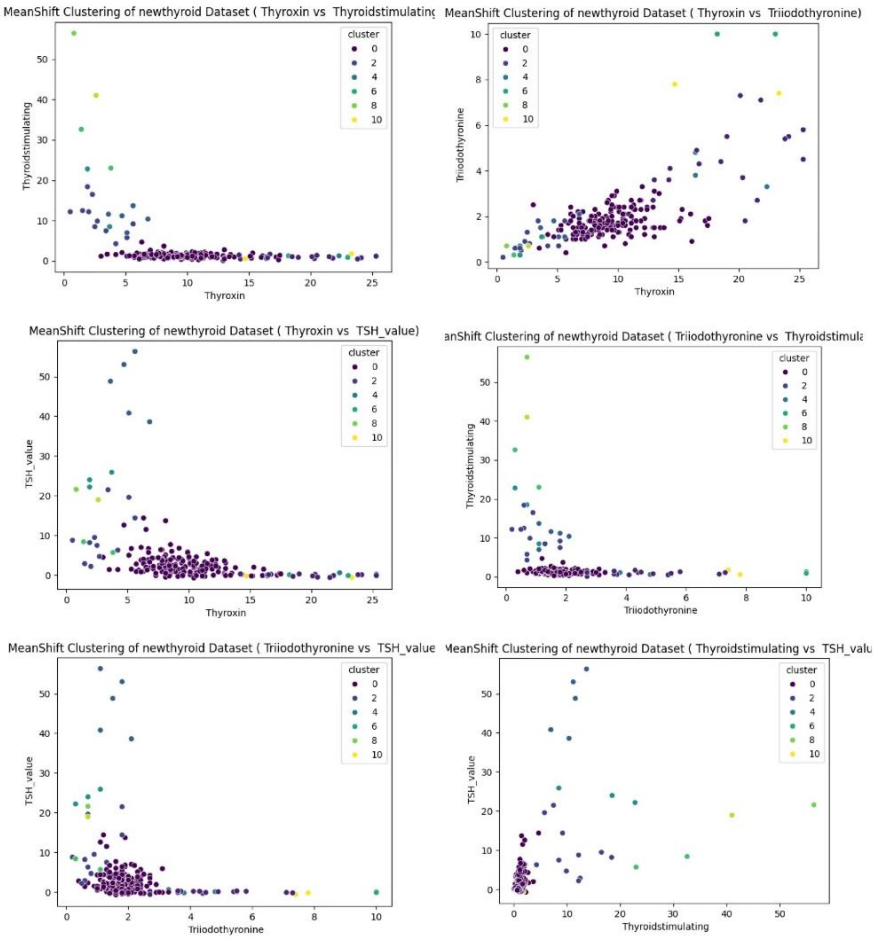
Συνεχίζοντας με τον αλγόριθμο DENCLUE και τις δύο συστάδες οι οποίες ανιχνεύονται γίνεται εύκολα αντιληπτό μέσα από τα scatter plot ότι το cluster 0 είναι πιο συμπαγές από το cluster 1 που

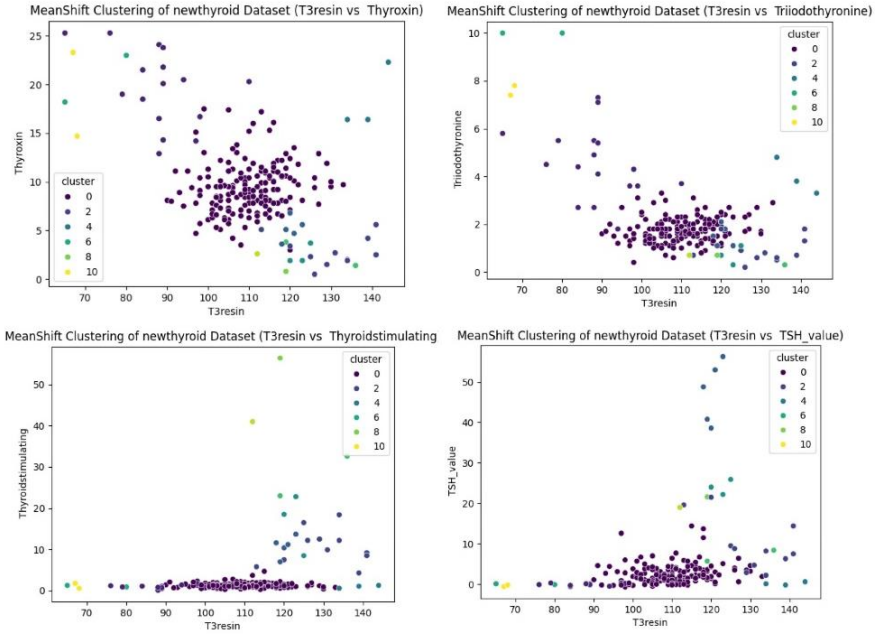
τείνει να είναι αρκετά αραιό. Ο διαχωρισμός τους δεν είναι ιδιαίτερα σαφής αφού τα δυο cluster διασταυρώνονται σε αρκετά σημεία ενώ σύμφωνα και με το διάγραμμα παράλληλων συντεταγμένων η διαφορά τους είναι πιο σαφής στο Feature 4 και 5 όπου και υπάρχει η μεγαλύτερη διακύμανση.



Εικόνα 4.54 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το New Thyroid

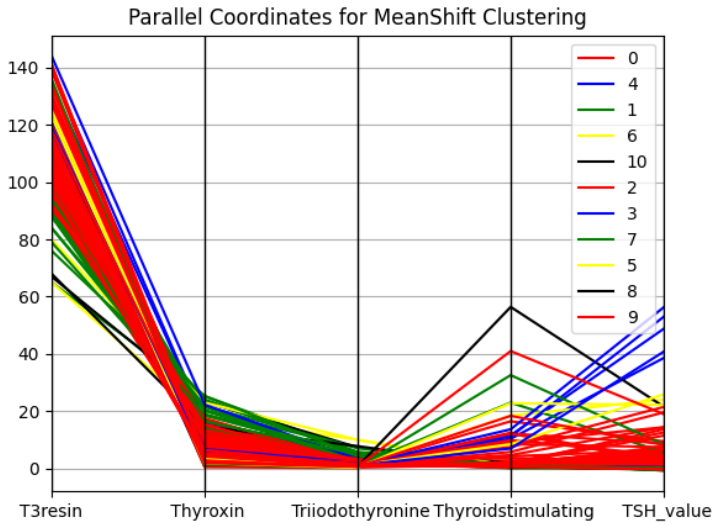
Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples=215)
Meanshift = Meanshift (bandwidth=bandwidth)





Εικόνα 4.55 Εξαγόμενα γραφήματα Scatter plot του Mean Shift για το New Thyroid

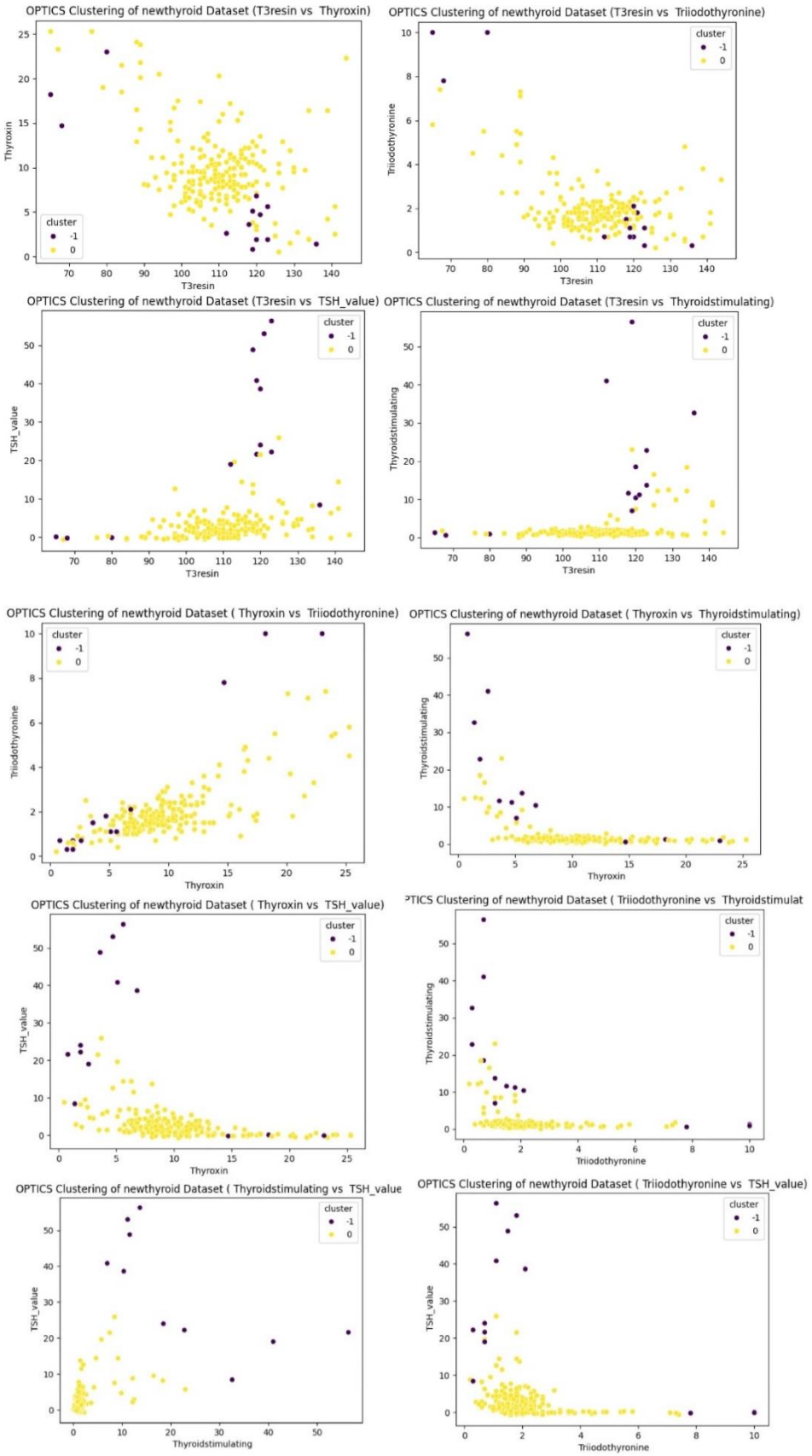
Στο New Thyroid dataset, ο Mean Shift φαίνεται να εντοπίζει συνολικά δέκα συστάδες χωρίς την εμφάνιση θορύβου. Το cluster 0 που απεικονίζεται στα scatter plot με μωβ χρώμα αποτελεί τη μεγαλύτερη και πιο πυκνή συστάδα από όλες με τις υπόλοιπες να σχηματίζονται από λίγα μόλις σημεία. Λανθασμένα και πάλι όπως φαίνεται από τον Πίνακα 8, ο αλγόριθμος δημιούργησε συστάδες με πλήθος σημείων 1, κατάσταση που οδηγεί την συσταδοποίηση ως αποτυχημένη.



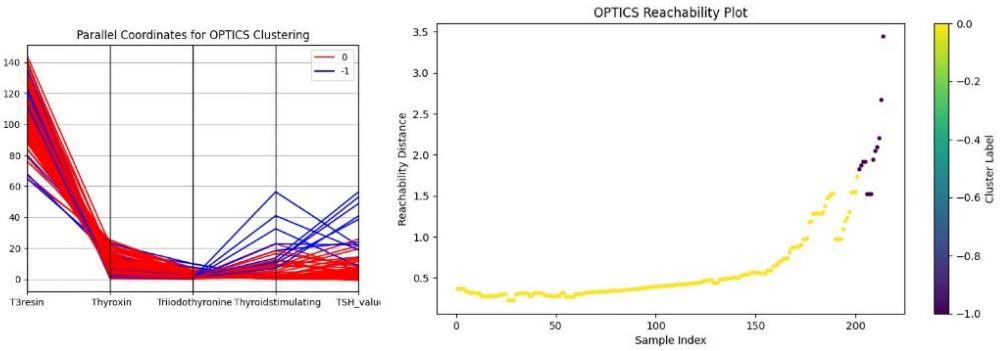
Εικόνα 4.56 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το New Thyroid

OPTICS: optics = OPTICS (min_samples = 5, min_cluster_size=0.9)

Στα διαγράμματα διασποράς που δημιουργήθηκαν για τον αλγόριθμο OPTICS (Εικόνα 4.57) ξεχωρίζει η συστάδα που εντοπίστηκε με τα noise points να καλύπτουν αρκετά μικρό κομμάτι του συνόλου. Όπως φαίνεται και στο parallel coordinates, οι γραμμές που απαρτίζουν την ομάδα ακολουθούν κοινή πορεία δημιουργώντας ένα μοτίβο όμως η συγχώνευση όλων των σημείων σε ένα cluster μπορεί εύκολα να οδηγήσει στην απώλεια χρήσιμων πληροφοριών που μπορεί να διαχωρίζουν τα δεδομένα.

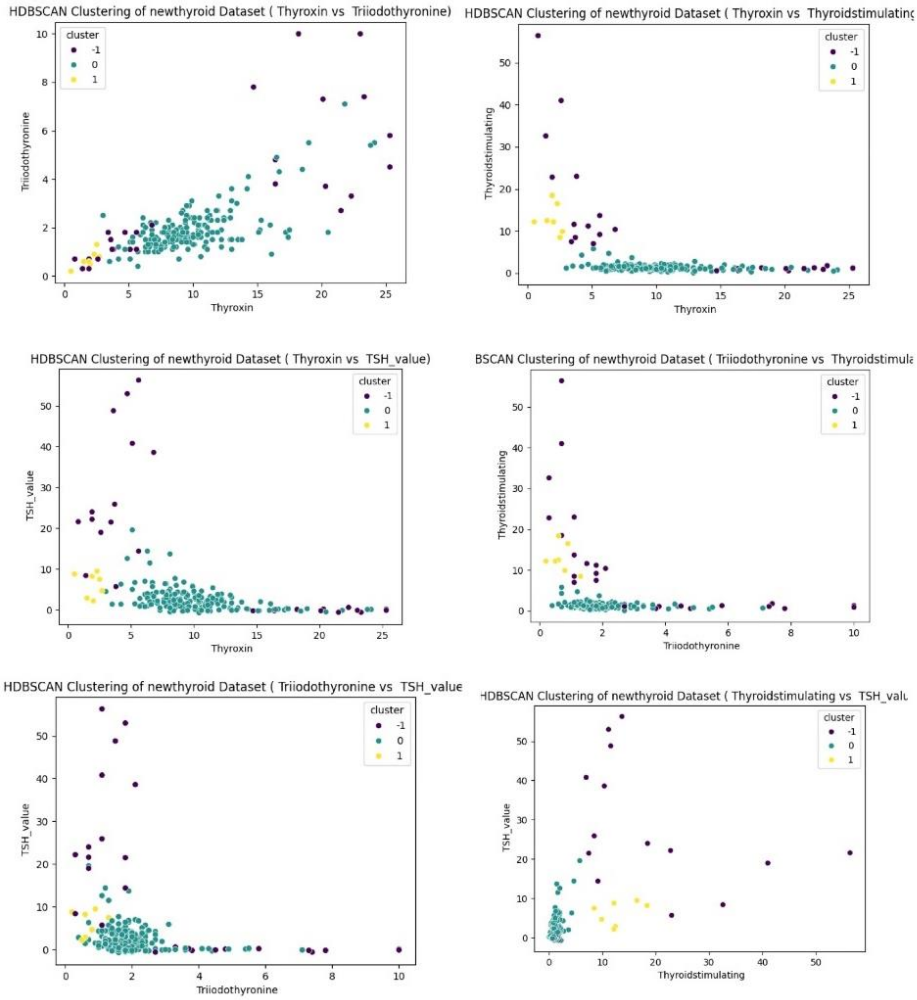


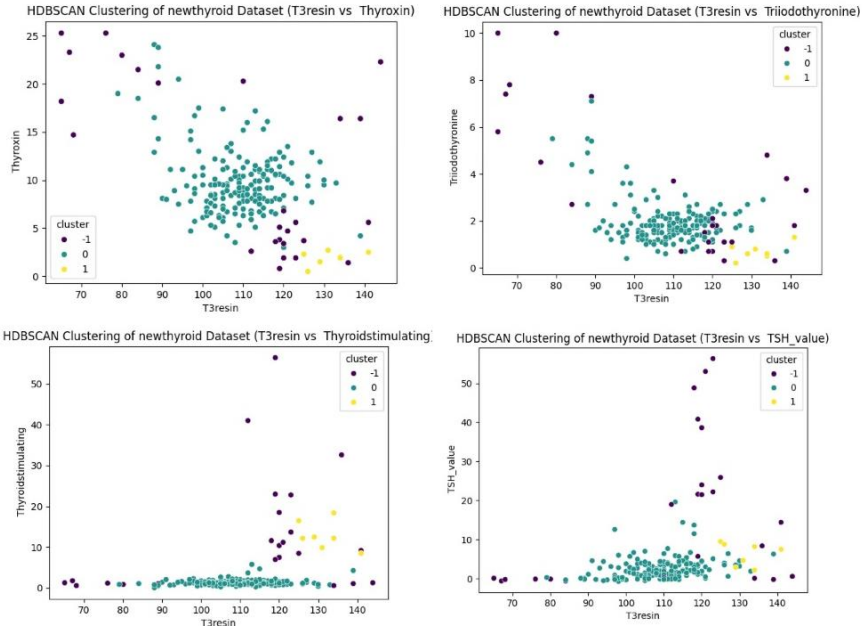
Εικόνα 4.57 Εξαγόμενα γραφήματα Scatter plot του OPTICS για το New Thyroid



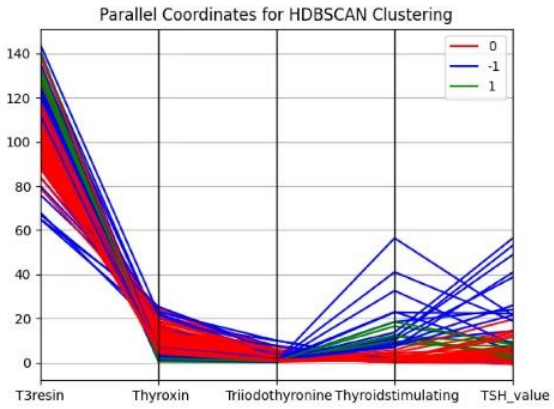
Εικόνα 4.58 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICS για το New Thyroid

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size= 4)





Εικόνα 4.59 Εξαγόμενα γραφήματα Scatter plot του HDBSCAN για το New Thyroid



Εικόνα 4.60 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το New Thyroid

Τελικός αλγόριθμος συσταδοποίησης για το σύνολο δεδομένων New Thyroid αποτέλεσε ο HDBSCAN, ο οποίος μετά την εκτέλεση του τοποθέτησε τα σημεία σε δυο ομάδες. Για κάθε ζεύγος χαρακτηριστικών, οι συστάδες είναι αρκετά ευδιάκριτες και σε απόσταση μεταξύ τους χωρίς επικαλύψεις. Ο διαχωρισμός μεταξύ τους εντοπίζεται εντονότερα στο χαρακτηριστικά Thyroidstimulating όπου κάθε συστάδα παίρνει τιμές από συγκεκριμένο εύρος.

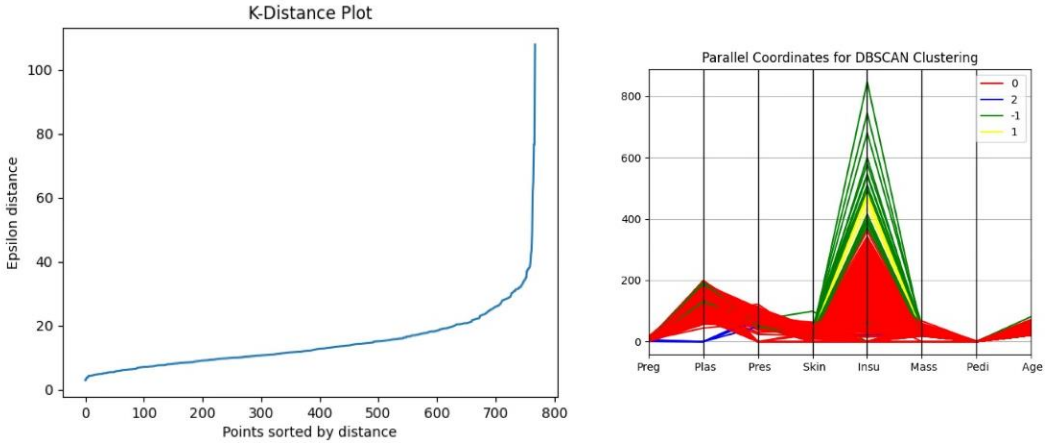
NEWTHYROID	CL0	CL1	CL2	CL3	CL4	CL5	CL6	CL7	CL8	CL9	CL10
DBSCAN	171	6	-	-	-	-	-	-	-	-	-
DENCLUE	51	164	-	-	-	-	-	-	-	-	-
MEANSHIFT	108	17	11	5	3	3	2	2	1	1	2
OPTICS	202	-	-	-	-	-	-	-	-	-	-
HDBSCAN	182	7	-	-	-	-	-	-	-	-	-

Πίνακας 8. Συνολικά Cluster του New Thyroid

4.1.9 Pima Indians Diabetes

Ως ένα επιπλέον γνωστό σύνολο δεδομένων αποτελεί το dataset Pima Indians Diabetes που χρησιμοποιείται για την Ιατρική διάγνωση του διαβήτη και περιέχει 768 δείγματα ασθενών με 8 διαφορετικά χαρακτηριστικά για την παρουσία ή μη αυτού. Τα χαρακτηριστικά αυτά αφορούν στοιχεία της υγείας του εκάστοτε ανθρώπου και ομαδοποιούνται σε συστάδες σύμφωνα με τις κοινές του ιδιότητες.[20]

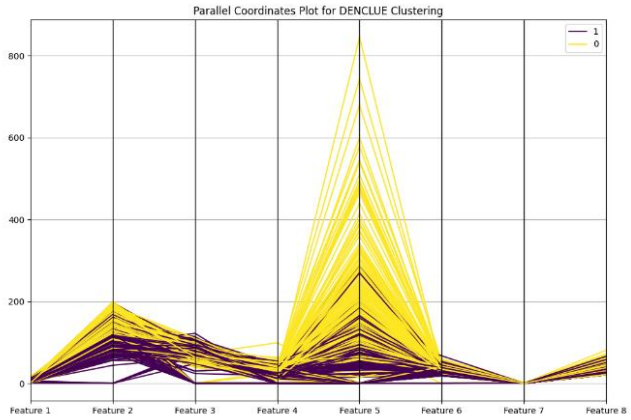
DBSCAN: dbscan = DBSCAN (eps= 37.0, min_samples=5)



Εικόνα 4.61 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το Pima

Εκτελώντας τον αλγόριθμο DBSCAN στο dataset PIMA, το εύρος της γειτονιάς εκτιμάται γύρω στο Eps=37.0. Με την παραμετροποίηση που δόθηκε εντοπίζονται τρείς συστάδες, ο διαχωρισμός των οποίων δεν γίνεται φανερός από το παραγόμενο διάγραμμα παράλληλων συντεταγμένων. Οι υψηλότερες τιμές λαμβάνονται στο χαρακτηριστικό Insu όπου υπάρχει και η μεγαλύτερη διακύμανση. Επιπλέον, η μη εμφανής εικόνα όλων των συστάδων στο γράφημα, οδηγεί στο συμπέρασμα ότι τα χαρακτηριστικά πιθανών να μη διαφέρουν έντονα μεταξύ τους ή να υπάρχει αλληλοεπικάλυψη αυτών. Ο θόρυβος φαίνεται να εξαλείφεται με επιτυχία αφού λαμβάνει και τις πιο ακραίες τιμές.

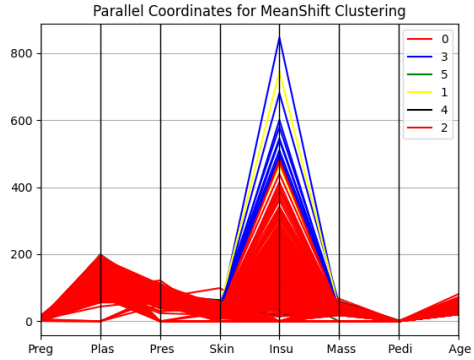
DENCLUE: denclue = DENCLUE (data_scaled, bandwidth= 3.0, threshold=1.0)



Εικόνα 4.62 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το Pima

Από την άλλη μεριά, ο DENCLUE δεν εντοπίζει σημεία θορύβου και ομαδοποιεί τα στοιχεία του χώρου σε δύο cluster. Και τα δυο εξ αυτών φαίνεται να έχουν αρκετές διακυμάνσεις με το χαρακτηριστικό Feature 5 να κάνει πιο αισθητό το διαχωρισμό τους. Από το διάγραμμα παράλληλων συντεταγμένων και σύμφωνα με τον Πίνακα 9, προκύπτει ότι αν και τα δυο cluster είναι χωρισμένα σχεδόν ισάξια, το cluster 1 αποτελεί την πιο πυκνή περιοχή σε σχέση με το cluster 0, που λαμβάνει αρκετά ακραίες τιμές.

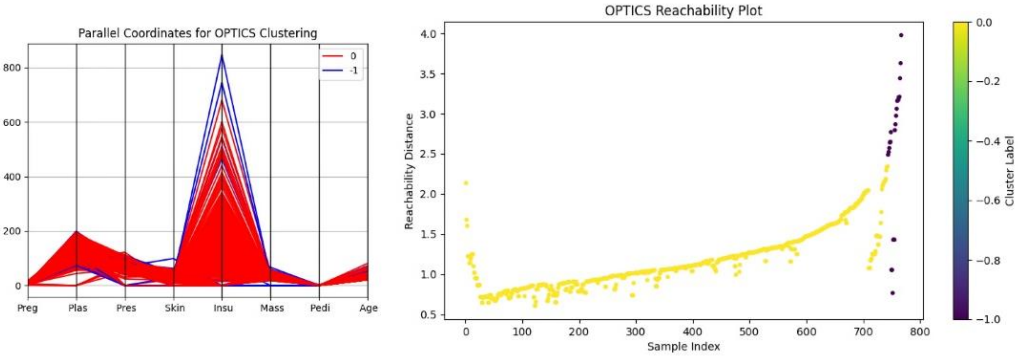
Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples=768)
Meanshift = Meanshift (bandwidth=bandwidth)



Εικόνα 4.63 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Pima

Επόμενη εκτέλεση για το dataset PIMA αποτελεί ο αλγόριθμος Mean Shift κατά τον οποίο ανιχνεύονται έξι διαφορετικές συστάδες χωρίς καμία ένδειξη θορύβου. Αναλύοντας το διάγραμμα παράλληλων συντεταγμένων, τα cluster 0 και 2 συγχέονται χωρίς να είναι εμφανής ο διαχωρισμός τους, ενώ τα cluster 4 και 5 δεν είναι ιδιαίτερα φανερά, κατάσταση κατά την οποία πιθανών να υπάρχει επικάλυψη από άλλες συστάδες. Ακόμη, οι υψηλότερες τιμές λαμβάνονται στο γνώρισμα Insu όπου φαίνεται η διάκριση ορισμένων από αυτών.

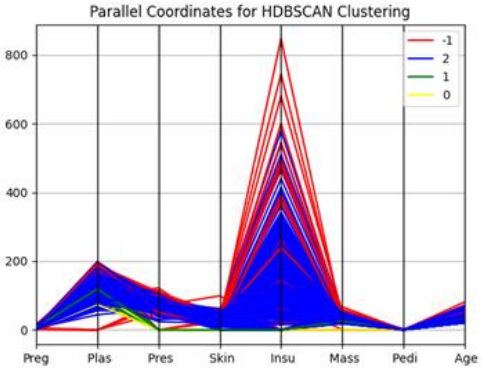
OPTICS: optics = OPTICS (min_samples = 4, min_cluster_size=0.9)



Εικόνα 4.64 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICS για το Pima

Στον OPTICS τα δεδομένα του συνόλου προσδιορίζονται σε μόνο μία ομάδα η οποία δεν παρουσιάζει ομοιογένεια και λαμβάνει αρκετά ακραίες τιμές. Παράλληλα, το διάγραμμα προσβασιμότητας δείχνει την κοιλάδα που ακολουθεί η συστάδα με τη διακοπή να συμβαίνει γύρω στην τιμή 2.5 όπου τα σημεία παίρνουν εξαιρετικά ανοδική πορεία.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size=4)



Εικόνα 4.65 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το Pima

Η συσταδοποίηση που πραγματοποιήθηκε από τον HDBSCAN εμφανίζεται να ανιχνεύει τρία cluster. Το cluster 2 είναι αυτό που καταλαμβάνει το μεγαλύτερο χώρο και αποτελεί την πιο πυκνή περιοχή ενώ τα cluster 1 και 0 κάνουν την εμφάνιση τους μόνο σε λίγα χαρακτηριστικά με τη διαφορά των τριών να είναι πιο έντονη στο χαρακτηριστικό Pres, προσδιορίζοντας το ως και το πιο σημαντικό εξ αυτών.

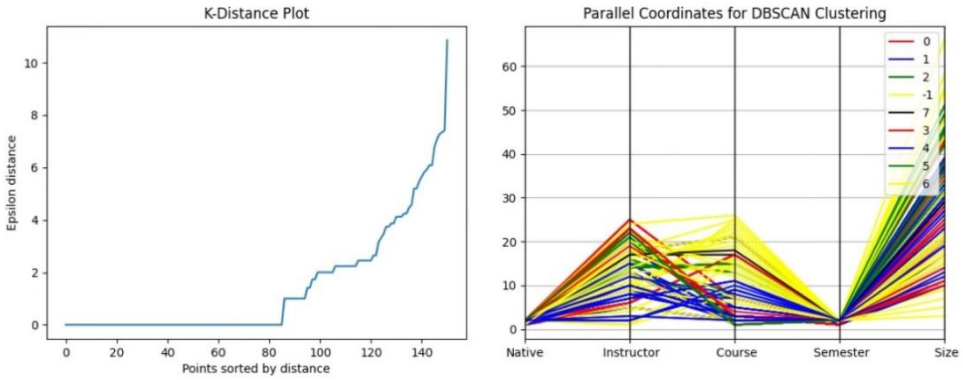
PIMA	CL0	CL1	CL2	CL3	CL4	CL5
DBSCAN	735	7	5	-	-	-
DENCLUE	332	436	-	-	-	-
MEANSHIFT	744	4	1	10	4	5
OPTICS	744	-	-	-	-	-
HDBSCAN	7	22	669	-	-	-

Πίνακας 9. Συνολικά Cluster του PIMA

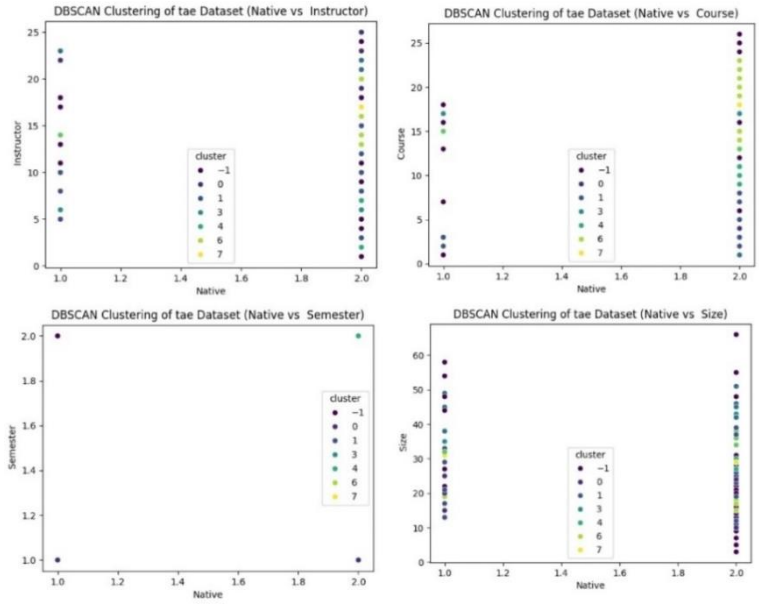
4.1.10 Teaching Assistant Evaluation

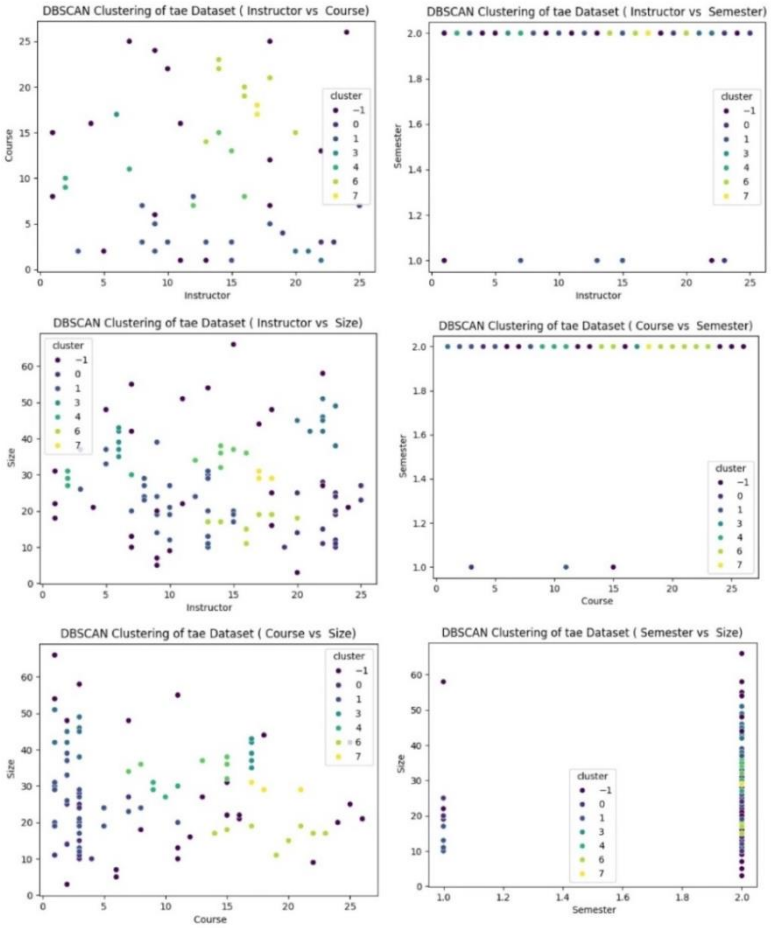
Το dataset TAE (Teaching Assistant Evaluation) αναφέρεται σε ένα σύνολο δειγμάτων που αφορούν την αξιολόγηση της απόδοσης των βοηθών δασκάλων σε εκπαιδευτικούς χώρους και συγκεκριμένα περιλαμβάνει ένα σύνολο από κριτικές που τους έχουν αποδοθεί από μαθητές ή φοιτητές. Οι αξιολογήσεις αυτές που δημιουργούν αυτό το μοντέλο είναι συνολικά 151 και περιέχουν 4 χαρακτηριστικά που αφορούν τον ίδιο τον βοηθό αλλά και το μάθημα που διδάσκει.[20]

DBSCAN: dbscan = DBSCAN (eps=2.5, min_samples=5)



Εικόνα 4.66 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το TAE

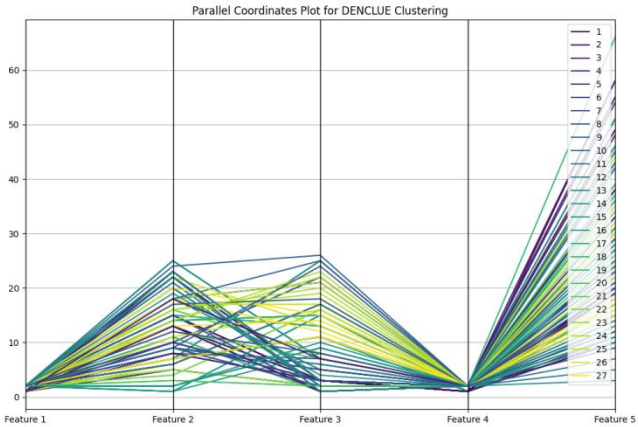




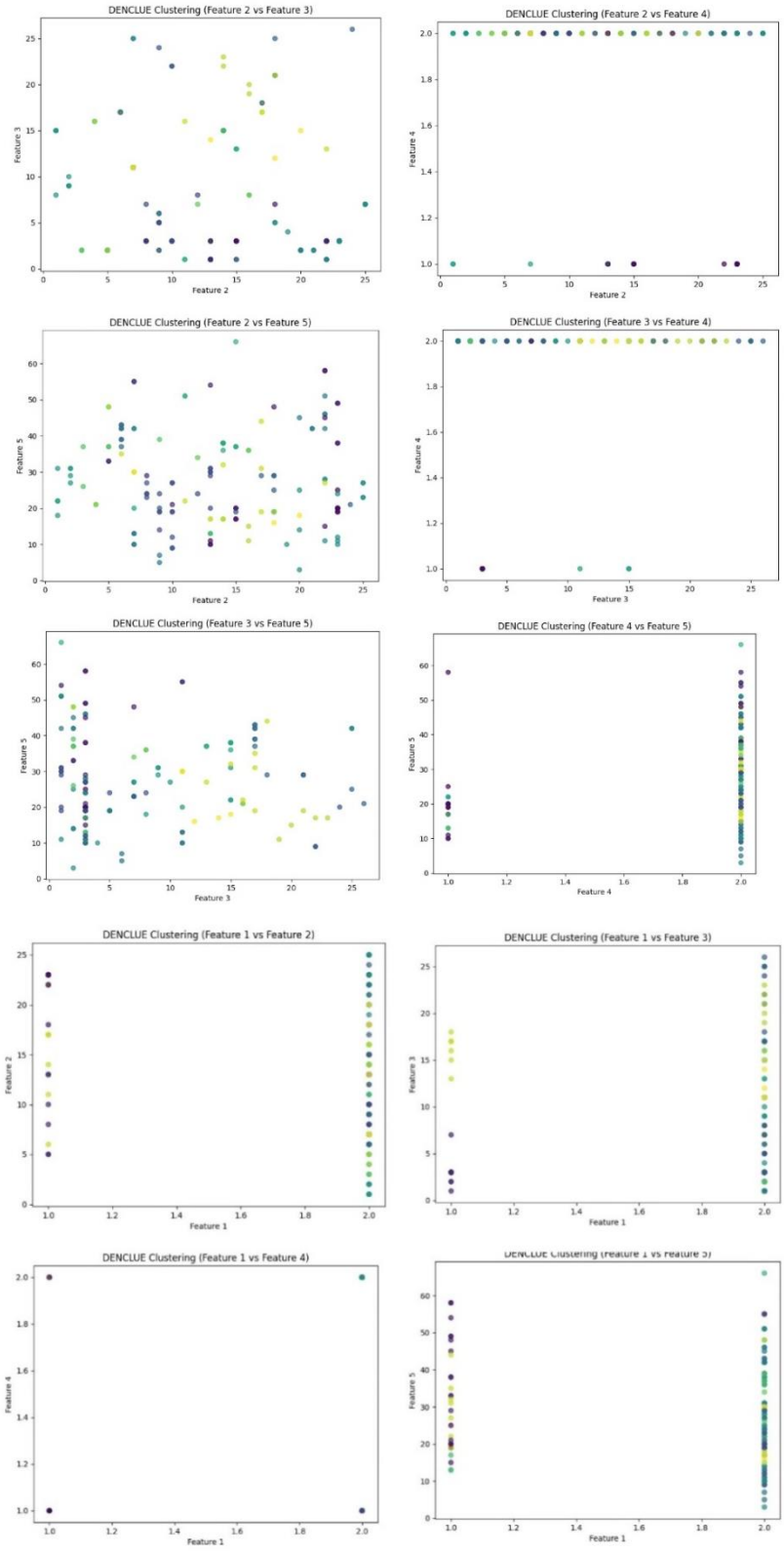
Εικόνα 4.67 Εξαγόμενα γραφήματα Scatter plot του DBSCAN για το TAE

Η συσταδοποίηση που πραγματοποιήθηκε κατά την εκτέλεση του αλγορίθμου DBSCAN για το dataset TAE, ανίχνευσέ οχτώ συστάδες σε ένα σύνολο 151 δειγμάτων με ακτίνα Eps=2.5 που βρέθηκε από το γράφημα k-distance. Ορισμένα από τα διαγράμματα διασποράς που παράχθηκαν δεν εμφανίζουν ευδιάκριτα τις συστάδες λόγω πιθανών της φύσης των δεδομένων, όμως στις σχέσεις Size-Course και Size-Instructor, ο διαχωρισμός αυτών γίνεται πιο ξεκάθαρος με το κάθε cluster να βρίσκεται συγκεντρωμένο σε μια περιοχή με μικρές επικαλύψεις. Αν και ο DBSCAN κατάφερε να ομαδοποιήσει τα δεδομένα, η μεγάλη αλληλοεπικάλυψη και η δομή του dataset αποτελούν μη ιδανική επιλογή για αυτόν τον αλγόριθμο.

DENCLUE: denclue = DENCLUE (data_scaled, bandwidth=0.7, threshold=0.5)



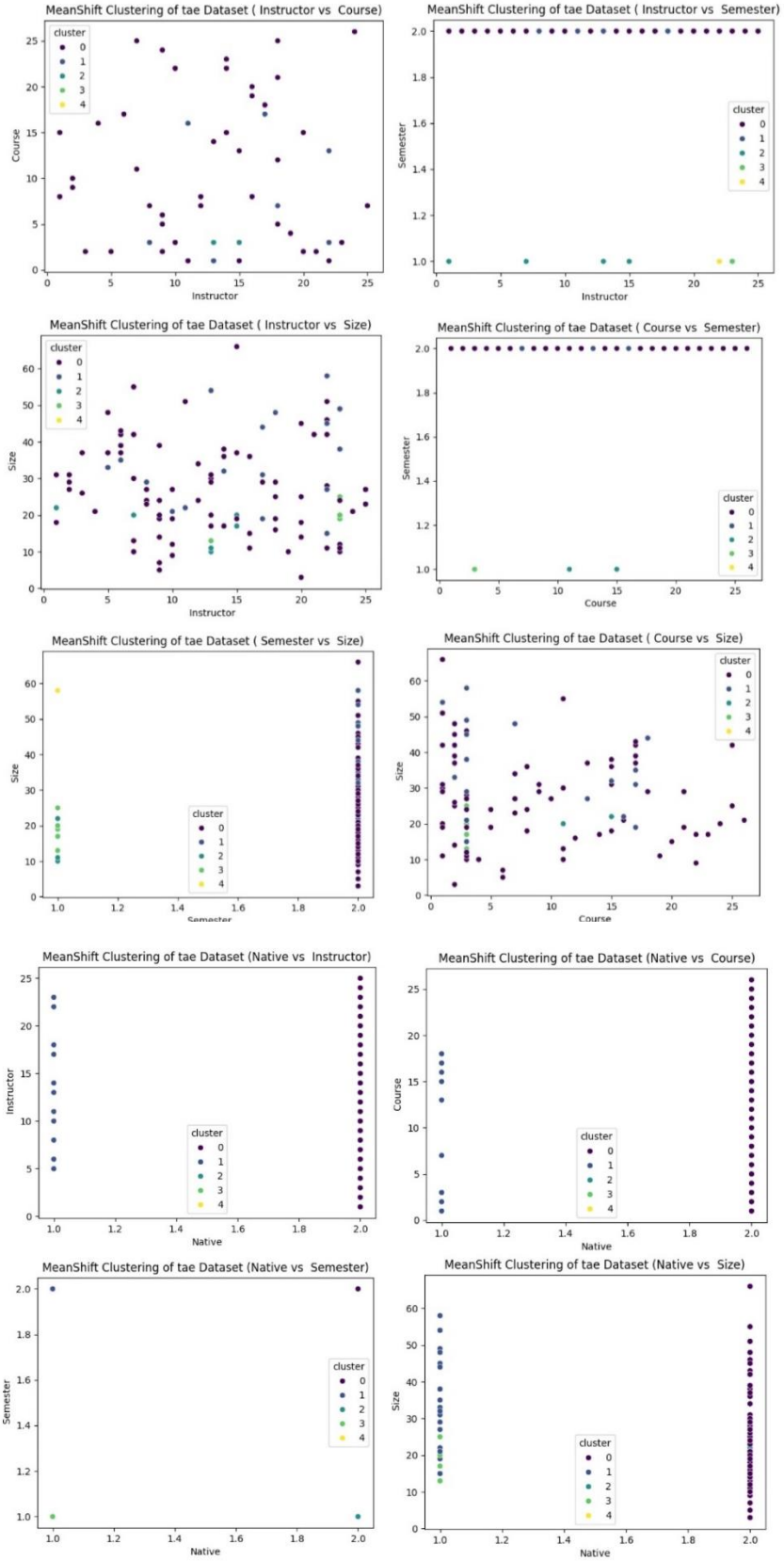
Εικόνα 4.68 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το TAE



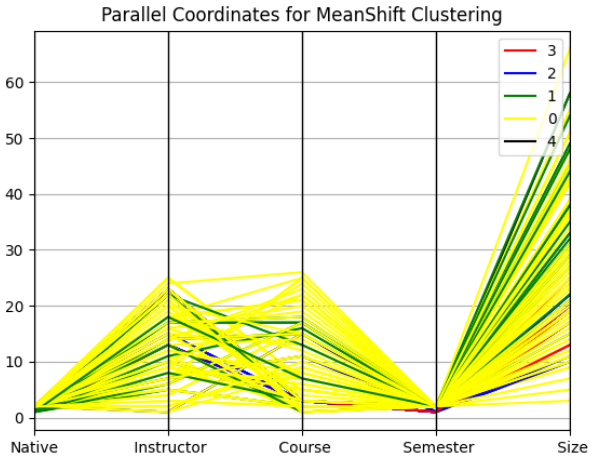
Εικόνα 4.69 Εξαγόμενα γραφήματα Scatter plot του DENCLUE για το TAE

Συνεχίζοντας με τον αλγόριθμο DENCLUE και τη συσταδοποίηση που πραγματοποιήθηκε, τα χαρακτηριστικά και πάλι φαίνονται να μη συσχετίζονται μεταξύ τους ενώ πολλά από τα cluster συμπίπτουν. Ο υπερβολικός διαχωρισμός συστάδων, αν και θα μπορούσε να φανεί χρήσιμος για την ανάλυση και τη διαφοροποίηση μεταξύ των στοιχείων, ωστόσο ο τόσο μεγάλος αριθμός των 27, οδηγεί στο συμπέρασμα ευαισθησίας των δεδομένων με τον αλγόριθμο να μη λειτουργεί ορθά και οι ομάδες να είναι πιθανών αναξιόπιστες.

Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples=151)
Meanshift = Meanshift (bandwidth=bandwidth)



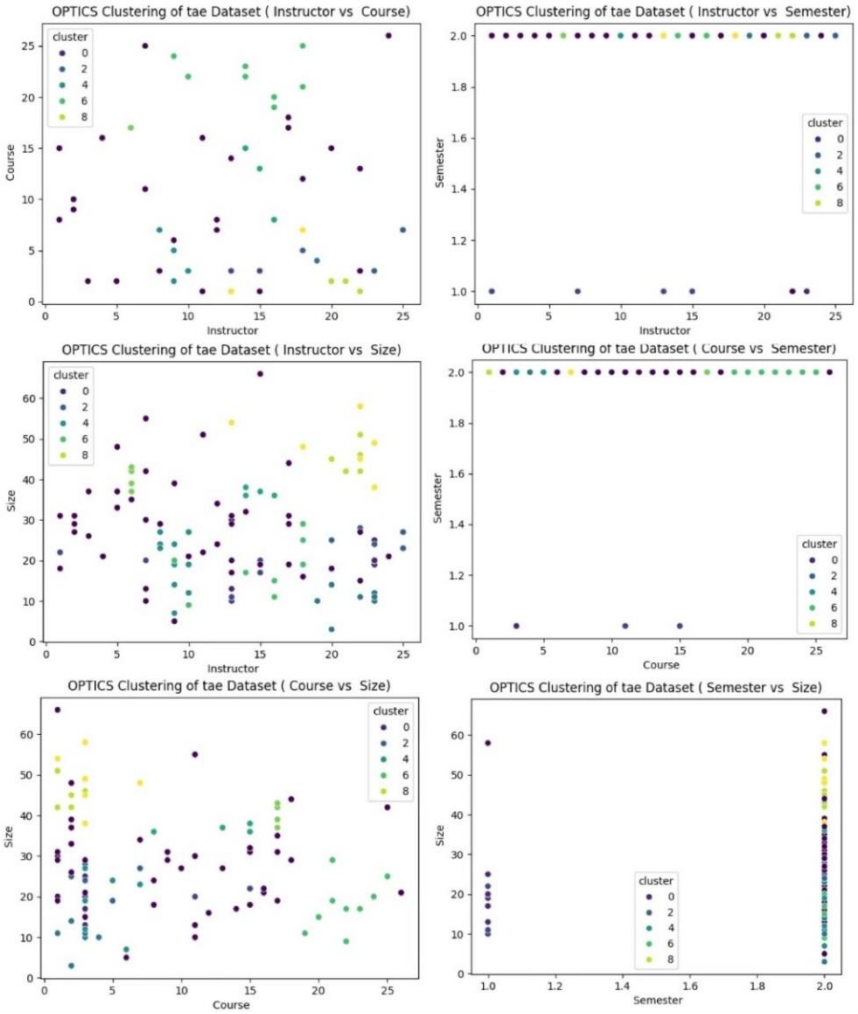
Εικόνα 4.70 Εξαγόμενα γραφήματα Scatter plot του Mean Shift για το TAE

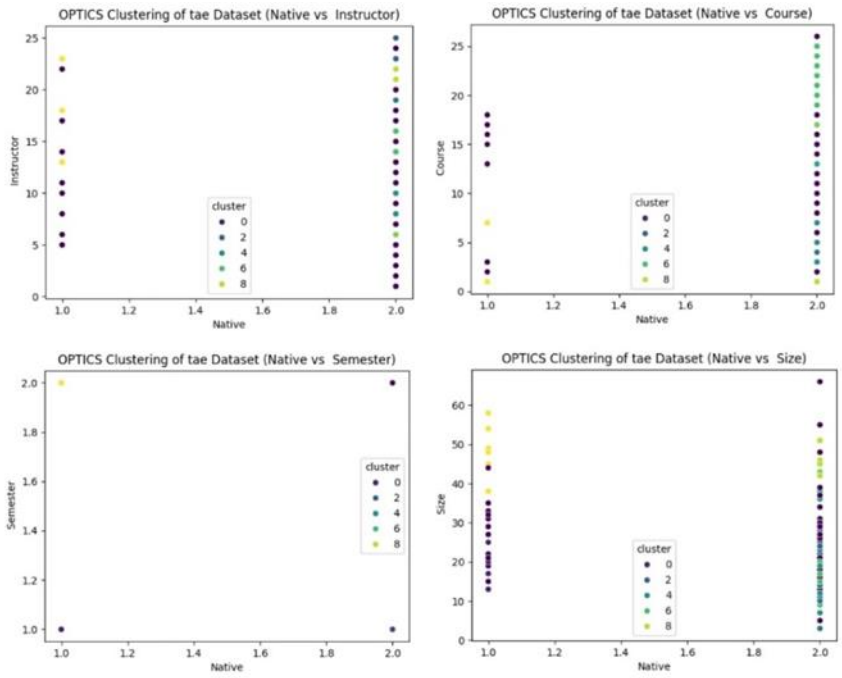


Εικόνα 4.71 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το TAE

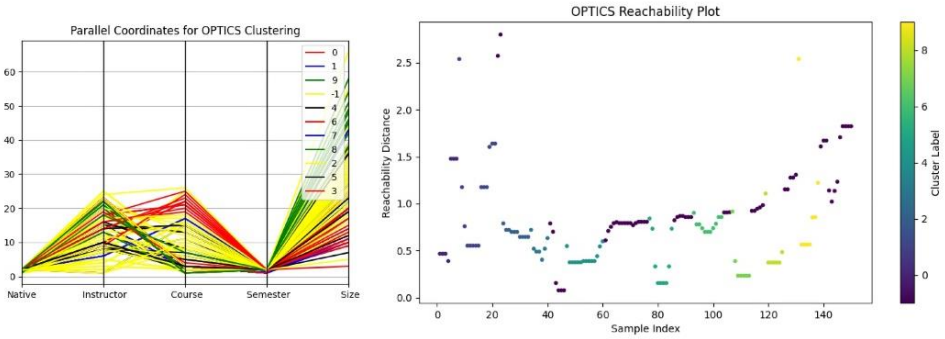
Μετέπειτα, για το dataset TAE με τη βοήθεια του αλγορίθμου Mean Shift, ανιχνεύθηκαν 5 ομάδες χωρίς την ένδειξη σημείων θορύβου με το cluster 0 να αποτελεί την πιο πυκνή περιοχή από όλες. Σύμφωνα με τα διαγράμματα διασποράς της Εικόνας 4.70, η διάσταση Native φαίνεται να αποτελεί σημαντικό κομμάτι για τη συσταδοποίηση αφού σχεδόν σε κάθε σχέση της με τα υπόλοιπα χαρακτηριστικά ο διαχωρισμός των τμημάτων συστάδων που απεικονίζονται είναι ξεκάθαρος ωστόσο. Αντίθετα, αξιολογώντας το διάγραμμα παράλληλων συντεταγμένων, η μεγαλύτερη διαφοροποίηση εντοπίζεται στη διάσταση Size.

OPTICS: optics = OPTICS (min_samples =5, min_cluster_size=7)





Εικόνα 4.72 Εξαγόμενα γραφήματα Scatter plot του OPTICS για το TAE

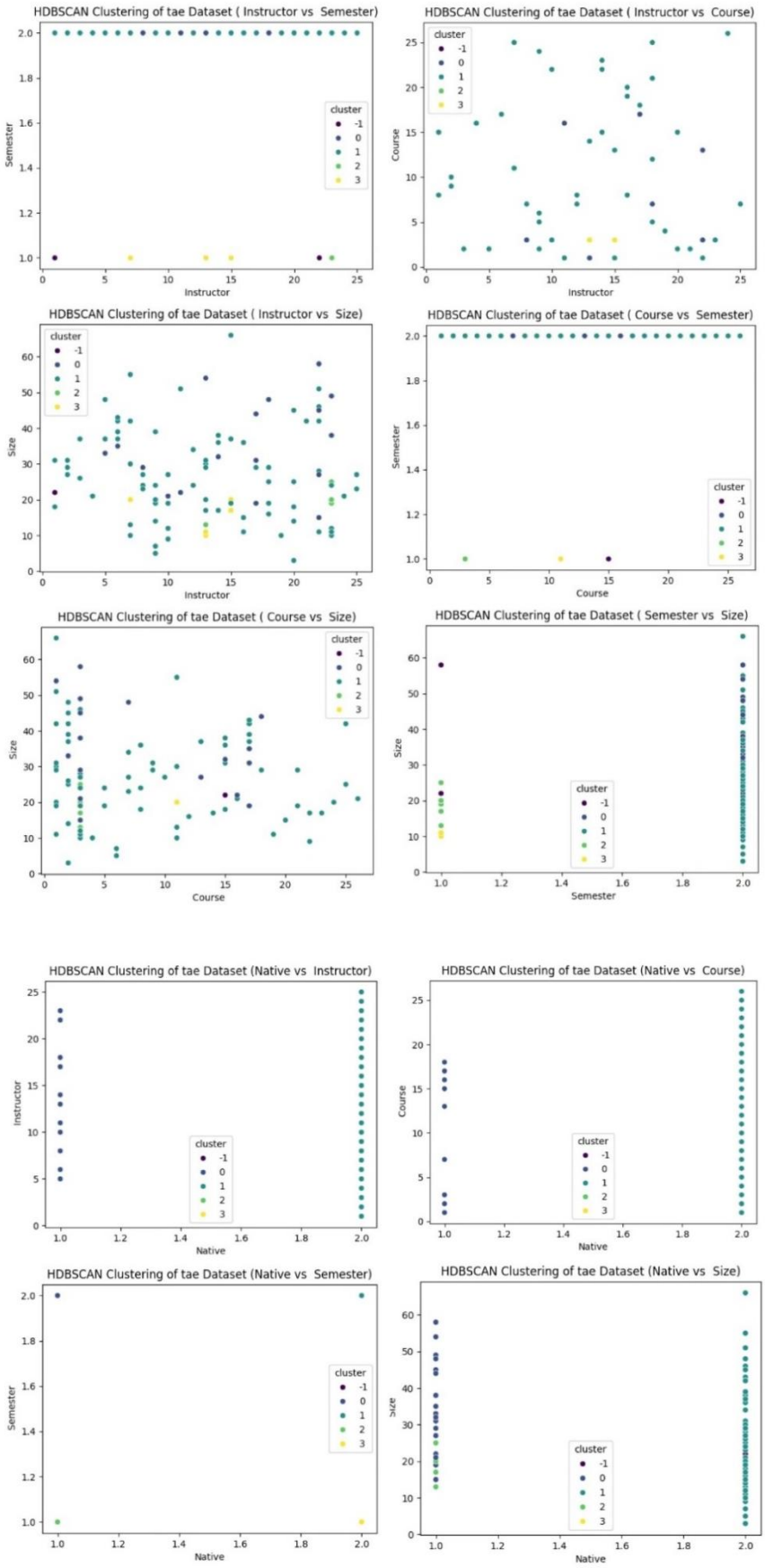


Εικόνα 4.73 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICIS για το TAE

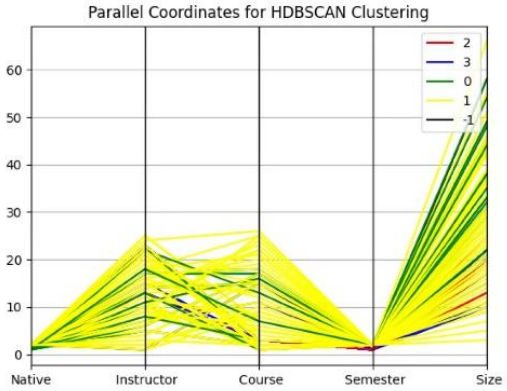
Ο αλγόριθμος OPTICS με τη σειρά του, φαίνεται να διαχωρίζει τα δεδομένα με ομοιόμορφο τρόπο. Κατά την εκτέλεση του ανιχνεύθηκαν δέκα συστάδες οι οποίες όμως σε πολλά σημεία καλύπτουν η μία την άλλη. Ενώ δεν υπάρχει ιδιαίτερη σύνδεση μεταξύ χαρακτηριστικών, το διάγραμμα παράλληλων συντεταγμένων δείχνει τη διάσταση Size να συμβάλει περισσότερο στη συσταδοποίηση. Παράλληλα, στο reachability plot κάθε cluster δημιουργεί τη δική του κοιλάδα τις στιγμές που οι αποστάσεις προσβασιμότητας γίνονται μικρότερες ενώ τα πρώτα από αυτά, φαίνεται να είναι και τα πιο αραιά.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size= 7)

Και στην περίπτωση του HDSCAN η σύγχυση που δημιουργείται μεταξύ των συστάδων είναι μεγάλη. Το γνώρισμα Native και πάλι δείχνει συσχέτιση κυρίως με τα χαρακτηριστικά Instructor, Course και Semester οπού ο διαχωρισμός ορισμένων cluster είναι πιο φανερός. Όμως, δεν φαίνεται ιδιαίτερα σημαντικό γνώρισμα για την συσταδοποίηση με τη διάσταση Size να παίρνει τις υψηλότερες τιμές και να κάνει φανερή τη διαφορά τους.



Εικόνα 4.74 Εξαγόμενα γραφήματα Scatter plot του HDBSCAN για το TAE



Εικόνα 4.75 Εξαγόμενα γραφήματα Stability plot και Parallel Coordinates του HDBSCAN για το TAE

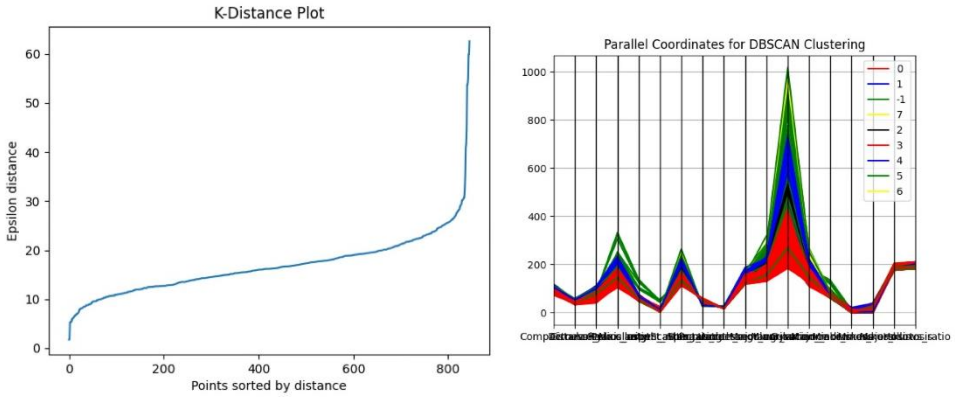
TAE	DBSCAN	DENCLUE	MEANSHIFT	OPTICS	HDBSCAN
CL0	24	1	108	8	20
CL1	43	6	20	14	108
CL2	12	7	14	10	8
CL3	8	8	8	7	12
CL4	6	4	1	14	-
CL5	11	2	-	9	-
CL6	8	4	-	11	-
CL7	4	21	-	7	-
CL8	-	3	-	7	-
CL9	-	5	-	8	-
CL10	-	7	-	-	-
CL11	-	6	-	-	-
CL12	-	7	-	-	-
CL13	-	2	-	-	-
CL14	-	7	-	-	-
CL15	-	6	-	-	-
CL16	-	3	-	-	-
CL17	-	7	-	-	-
CL18	-	3	-	-	-
CL19	-	3	-	-	-
CL20	-	5	-	-	-
CL21	-	3	-	-	-
CL22	-	2	-	-	-
CL23	-	2	-	-	-
CL24	-	5	-	-	-
CL25	-	7	-	-	-
CL26	-	2	-	-	-
CL27	-	3	-	-	-

Πίνακας 10. Συνολικά Cluster του TAE

4.1.11 Vehicle

Το Vehicle αποτελεί ένα σύνολο δεδομένων διαφορετικό από τα υπόλοιπα αφού περιλαμβάνει πληροφορίες σχετικά με διάφορους τύπους αυτοκινήτων. Χρησιμοποιεί 18 χαρακτηριστικά όπως το σχήμα, το μήκος ή το πλάτος ενός αυτοκινήτου, για να τα ξεχωρίσει σε ένα σύνολο 846 δειγμάτων οχημάτων. Ενώ η ταξινόμηση θα χώριζε αυτά σε συγκεκριμένους τύπους, η συσταδοποίηση έρχεται και προσφέρει μια ομαδοποίηση των οχημάτων με κοινά χαρακτηριστικά όπως το ίδιο σχήμα. [20]

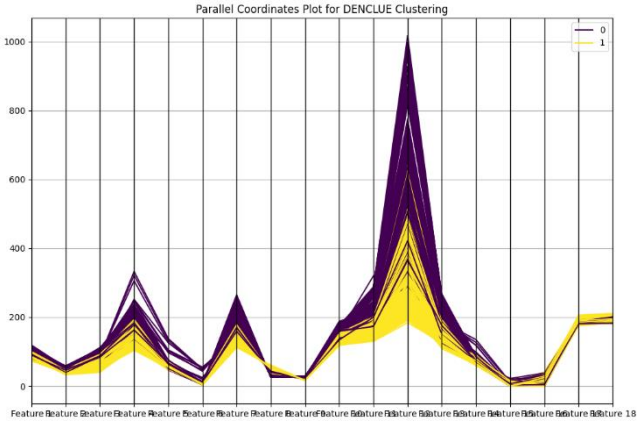
DBSCAN: dbscan = DBSCAN (eps=25.0, min_samples=4)



Εικόνα 4.76 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το Vehicle

Με την εκτέλεση του αλγορίθμου DBSCAN στα δεδομένα του Vehicle, εντοπίστηκαν 8 συστάδες χρησιμοποιώντας ως περιοχή γειτονιάς μιας απόσταση Eps=25.0 που ανακαλύφθηκε από τον αγκώνα που σχηματίζει η πορεία των αποστάσεων στο γράφημα k-distance. Ο θόρυβος φαίνεται να αναγνωρίστηκε με επιτυχία σύμφωνα με τις παραμέτρους ενώ σε αρκετά χαρακτηριστικά βρίσκεται στις υψηλότερες τιμές. Τα cluster προχωράνε στα γνωρίσματα με σχετικό μοτίβο και στα περισσότερα από αυτά όσων δεν επικαλύπτονται ο διαχωρισμός είναι φανερός.

DENCLUE: denclue = DENCLUE (data_scaled, bandwidth=2.0, threshold= 0.8)

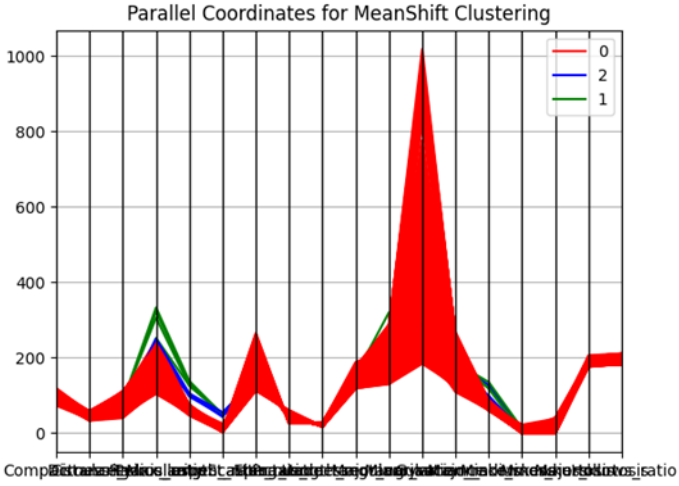


Εικόνα 4.77 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το Vehicle

Συνεχίζοντας στον DENCLUE, αλγόριθμος καταφέρνει να διαχωρίσει τα σημεία του χώρου, σε δύο περίπου ισοσταθμισμένα μέρη, με την διάκριση αυτών σε συστάδες να είναι ξεκάθαρη σχεδόν σε όλα τα χαρακτηριστικά . Το γνώρισμα με τη μεγαλύτερη διακύμανση που αποτελεί και το σημαντικότερο για τη συσταδοποίηση όπως φαίνεται και στην Εικόνα 4.77, είναι το Feature 12 όπου λαμβάνει και τις υψηλότερες τιμές.

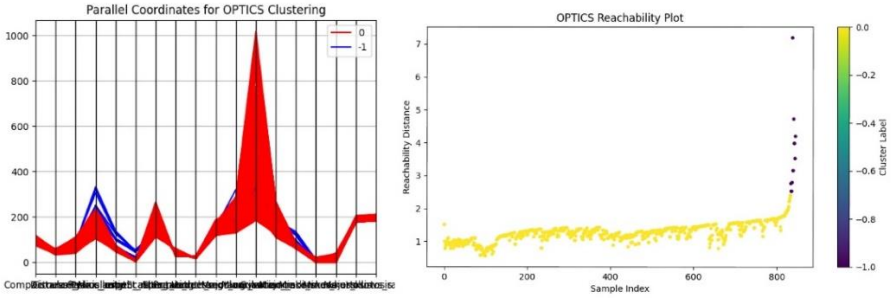
Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples=846)
Meanshift = Meanshift (bandwidth=bandwidth)

Ο αλγόριθμος Mean Shift για το σύνολο των 846 δειγμάτων, ομαδοποίησε τα δεδομένα σε τρείς συστάδες χωρίς την εύρεση θορύβου. Ωστόσο, το cluster 0, που αποτελεί και το μεγαλύτερο και πιο πυκνό από όλα, φαίνεται να παρουσιάζει ακραίες τιμές στο δωδέκατο χαρακτηριστικό γεγονός που σημαίνει πιθανή ένταξη outlier λανθασμένα στη συστάδα. Ο διαχωρισμός και των τριών αυτών ομάδων γίνεται περισσότερο ευδιάκριτος στο 4^{ο} και 5^{ο} γνώρισμα.



Εικόνα 4.78 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Vehicle

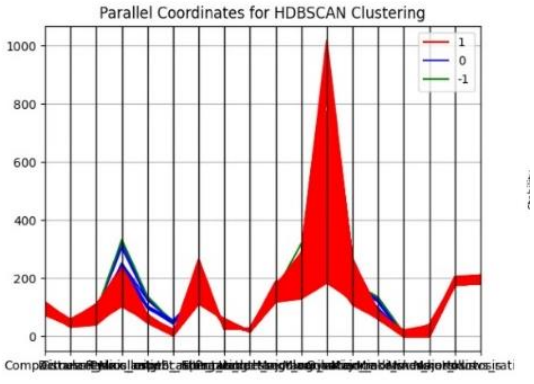
OPTICS: optics = OPTICS (min_samples = 3.0, min_cluster_size=0.1)



Εικόνα 4.79 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICS για το Vehicle

Για τη συσταδοποίηση με τη χρήση του αλγορίθμου OPTICS, το σύστημα εντόπισε μία και μοναδική συστάδα με την απομάκρυνση μόνο ελάχιστων σημείων θορύβων. Στην περίπτωση αυτή αν τα σημεία δεν είναι τόσο πυκνά ή όμοια μεταξύ τους χάνονται χρήσιμες πληροφορίες για την ανάλυση και δεν μπορεί να αξιολογηθεί σωστά η αποτελεσματικότητα του αλγορίθμου. Ωστόσο, το cluster φαίνεται να ακολουθεί μια επίπεδη κοιλάδα στο reachability plot γεγονός που κάνει τη συστάδα να φαίνεται ιδιαίτερα πυκνή.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size= 6)



Εικόνα 4.80 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το Vehicle

Τέλος, ο HDBSCAN τοποθέτησε τα σημεία σε δύο ομάδες αφαιρώντας εκείνα τα οποία δεν ανήκουν σε καμία από αυτές. Η συσταδοποίηση τείνει να μοιάζει με την ομαδοποίηση που πραγματοποίησε ο αλγόριθμος OPTICS με τα σημεία θορύβου που είχαν αναγνωριστεί εκεί, να φαίνεται πως έχουν ενταχθεί σε ένα νέο cluster στον HDBSCAN. Η διαφοροποίηση τους γίνεται πιο ορατή στο 4ο, 5ο και 6ο χαρακτηριστικό όπου κάθε cluster λαμβάνει διαφορετικές τιμές ενώ το cluster 1 αποτελεί την πιο πυκνή περιοχή.

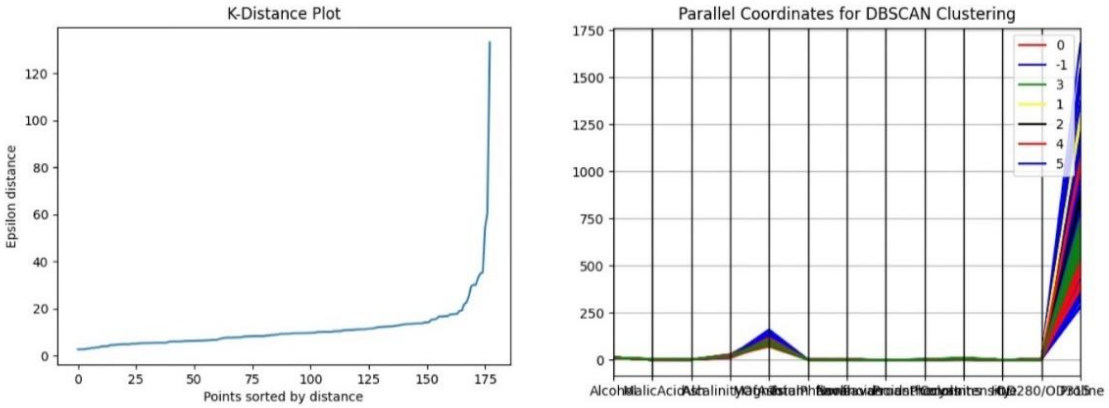
VEHICLE	CL0	CL1	CL2	CL3	CL4	CL5	CL6	CL7
DBSCAN	517	171	35	11	14	4	4	4
DENCLUE	377	469	-	-	-	-	-	-
MEANSHIFT	838	4	4	-	-	-	-	-
OPTICS	834	-	-	-	-	-	-	-
HDBSCAN	7	838	-	-	-	-	-	-

Πίνακας 11. Συνολικά Cluster του Vehicle

4.1.12 Wine

Ένα εξίσου δημοφιλές dataset στο χώρο της μηχανικής μάθησης αποτελεί το σύνολο δεδομένων Wine το οποίο περιέχει στοιχεία από ποικιλίες κρασιών της Ιταλίας και χρησιμοποιείται συχνά σε εκπαιδευτικά περιβάλλοντα για πειραματισμούς. Αποτελείται από 178 δείγματα κρασιών με 13 γνωρίσματα που τα ξεχωρίζουν όπως είναι η αλκοόλη, η χρωματική ένταση και άλλα λοιπά παρεμφερή χαρακτηριστικά.[20]

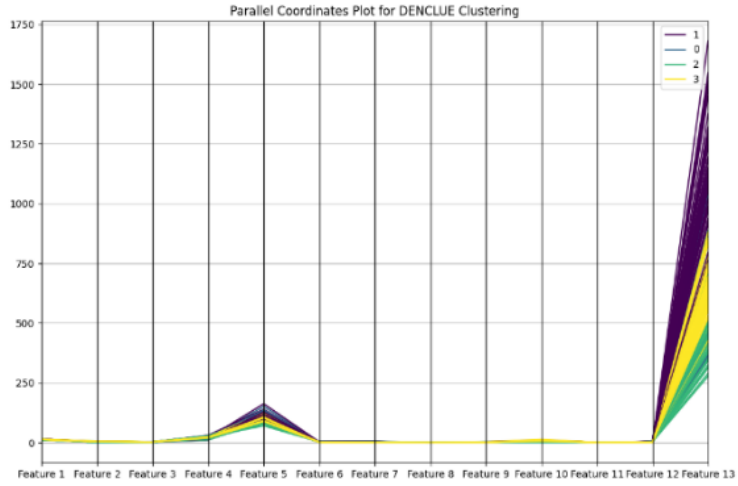
DBSCAN: dbscan = DBSCAN (eps=20.0, min_samples= 4)



Εικόνα 4.81 Εξαγόμενα γραφήματα K-distance plot και Parallel Coordinates του DBSCAN για το Wine

Τα αποτελέσματα που παράγονται από την εκτέλεση του αλγορίθμου DBSCAN στο σύνολο δεδομένων Wine μετά την εύρεση της κατάλληλης ακτίνας Eps=20.0 και τουλάχιστον 4 σημείων για τη δημιουργία ενός cluster, είναι 6 συστάδες. Ο διαχωρισμός μεταξύ τους είναι φανερός στο τελευταίο χαρακτηριστικό Proline, όπου οι τιμές παίρνουν μεγάλη έκταση και αποτελούν κρίσιμες για τη συσταδοποίηση. Επιπλέον, υπάρχει μεγάλη ομοιογένεια στα περισσότερα γνωρίσματα όμως οι τιμές τους είναι αρκετά χαμηλές για να χρήζουν ιδιαίτερης σημασίας με μοναδική διακύμανση στο 5ο χαρακτηριστικό Magnesium.

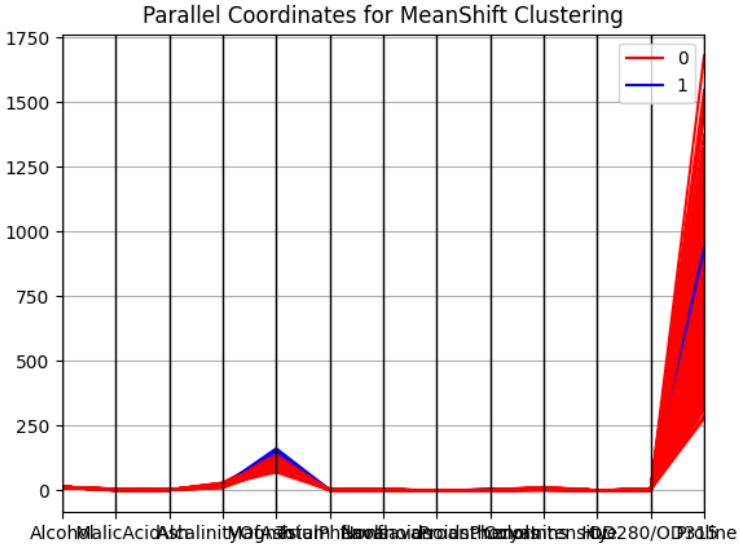
DENCLUE: Denclue = DENCLUE (data_scaled, bandwidth=2.0, threshold=0.7)



Εικόνα 4.82 Εξαγόμενο γράφημα Parallel Coordinates του DENCLUE για το Wine

Συνεχίζοντας, η διαφοροποίηση των πέντε συστάδων που δημιουργήθηκαν από τον αλγόριθμο DENCLUE, δείχνουν αρκετά ξεκάθαρες γεγονός που κάνει την συσταδοποίηση επιτυχημένη. Λόγω της δομής των δεδομένων του Wine η διάσταση Feature 13 (Proline) και 5(Magnesium) είναι και πάλι αυτή που κάνει τη διαφορά εμφανής με τις μεγαλύτερες τιμές λαμβάνοντας διαφορετικό εύρος τιμών για κάθε cluster.

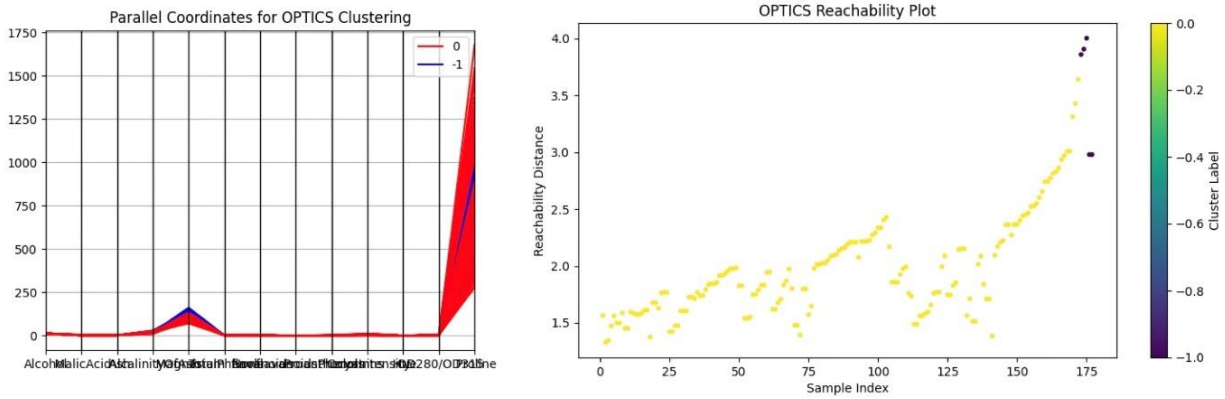
Mean Shift: bandwidth = estimate_bandwidth (data_scaled, n_samples= 178)
Meanshift = Meanshift (bandwidth=bandwidth)



Εικόνα 4.83 Εξαγόμενο γράφημα Parallel Coordinates του Mean Shift για το Wine

Ο Mean Shift αντίστοιχα, αντιλήφθηκε τα σημεία του συνόλου ως δύο διαφορετικές συστάδες με το cluster 0 να καταλαμβάνει τον περισσότερο χώρο. Οι μεγαλύτερες αιχμές εμφανίζονται στο 5ο και 13ο χαρακτηριστικό όπου διακρίνεται και πιο έντονα η διαφορά μεταξύ τους και αποτελούν για ακόμη μια φορά τα πιο σημαντικά γνωρίσματα για τη συσταδοποίηση αν και η επικάλυψη συνολικά είναι μεγάλη.

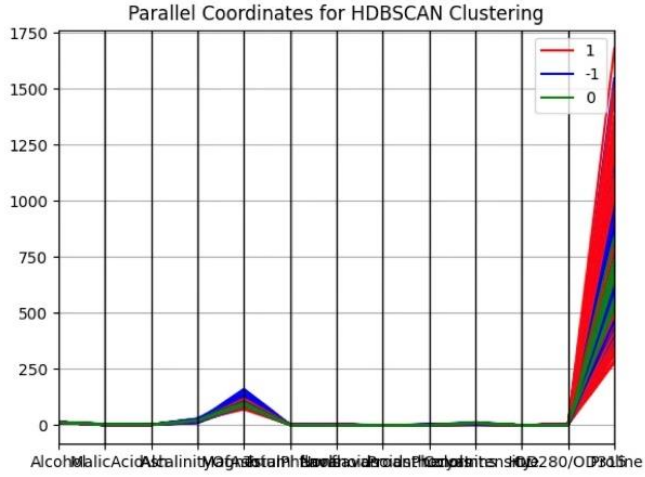
OPTICS: optics = OPTICS (min_samples =3, min_cluster_size=0.3)



Εικόνα 4.84 Εξαγόμενα γραφήματα Reachability plot και Parallel Coordinates του OPTICIS για το Wine

Οι συστάδες που κατάφερε να δημιουργήσει ο αλγόριθμος OPTICS τείνουν να μοιάζουν με τη συσταδοποίηση που πραγματοποίησε ο HDBSCAN αλλά αντί για δύο συστάδες τα δεδομένα της μίας αναγνωρίζονται ως θόρυβος. Στο reachability plot φαίνεται πως το cluster λαμβάνει πολλές τιμές χωρίς να είναι απαραίτητα κοντά, με αρκετά σημεία που απομακρύνονται να παραμένουν στη συστάδα. Η δομή των αποστάσεων προσβασιμότητας δείχνει πως τα δεδομένα θα μπορούσαν να χωριστούν σε περισσότερες από μια ομάδες ενώ οι ακραίες τιμές που εμφανίζονται υποδηλώνουν πιθανό θόρυβο λανθασμένα τοποθετημένο.

HDBSCAN: hdb = hdbscan. HDBSCAN (min_cluster_size=4)



Εικόνα 4.85 Εξαγόμενο γράφημα Parallel Coordinates του HDBSCAN για το Wine

Από την άλλη πλευρά, ο HDBSCAN εντόπισε δύο ομάδες με τη διαφοροποίηση τους να είναι πιο έντονη στο τελευταίο χαρακτηριστικό. Παράλληλα, το cluster 0 ακολουθεί κάποιο μοτίβο σε σχέση με το cluster 1 ενώ και πάλι τα χαρακτηριστικά δεν είναι επαρκή για τη συσταδοποίηση με τις τιμές τους να βρίσκονται σε κοινή κλίμακα σε όλη τη πορεία με ελάχιστες εξαιρέσεις και μεγάλες διασταυρώσεις.

WINE	CL0	CL1	CL2	CL3	CL4	CL5
DBSCAN	16	11	12	51	43	9
DENCLUE	23	55	52	48	-	-
MEANSHIFT	174	4	-	-	-	-
OPTICS	173	-	-	-	-	-
HDBSCAN	34	86	-	-	-	-

Πίνακας 12. Συνολικά Cluster του Wine

	DBSCAN		HDBSCAN	OPTICS		MEANSHIFT	DENCLUE	
	EPS	MIN SAMPLES	MIN CLUSTER SIZE	MIN SAMPLES	MIN CLUSTER SIZE	N SAMPLES	BANDWIDTH	THRESHOLD
APPENDICITIS	0.3	9	7	8	0.05	106	0.3	0.2
BANANA	0.15	6	8	6	0.2	5300	0.15	0.1
BUPA	25.0	5	2	4	0.1	345	1.5	0.6
GLASS	1.0	7	8	5	0.9	214	1.0	0.4
HABERMAN	5.0	5	3	5	0.1	306	0.3	0.2
IRIS	0.5	5	4	5	0.6	150	0.5	0.3
MAGIC	30.0	8	8	2	1.0	19020	3.0	1.2
NEWTHYROID	5.0	5	4	5	0.9	215	2.0	0.7
PIMA	37.0	5	4	4	0.9	768	3.0	1.0
TAE	5.5	5	7	5	7	151	0.7	0.5
VEHICLE	25.0	4	6	3	0.1	846	2.0	0.8
WINE	20.0	4	4	3	0.3	178	2.0	0.7

Πίνακας 13. Πλήρης πίνακας παραμέτρων

4.2 Αξιολόγηση αλγορίθμων με την μετρική Silhouette Score

Για την εύρεση του βέλτιστου αλγορίθμου που θα ικανοποιεί τις απαιτήσεις συσταδοποίησης ενός προβλήματος, σημαντικό ρόλο κατέχουν οι μετρικές αξιολόγησης που αναφέρθηκαν και στο Κεφάλαιο 3ο. Η βοήθεια που προσφέρουν στην εκτίμηση της ποιότητας του αλγορίθμου και των ομάδων που δημιουργεί, χρήζουν ιδιαίτερης σημασίας. Ειδικότερα, οι μετρικές επιτρέπουν στο χρήστη να κατανοεί τον τρόπο λειτουργίας ενός αλγορίθμου και παράλληλα ελέγχουν την απόδοση του στην ομαδοποίηση. Μέσω της αξιολόγησης ο χρήστης εντοπίζει τις αδυναμίες που μπορεί να παρουσιάζει η εκάστοτε συσταδοποίηση, φροντίζοντας να τις διορθώσει ρυθμίζοντας ξανά ενδεχομένως τις απαραίτητες παραμέτρους ώστε οι ομάδες που θα παραχθούν να είναι χρήσιμες για την ανάλυση. Επιπλέον, η δυνατότητα σύγκρισης που παρέχει μεταξύ διαφορετικών μεθόδων διαχωρισμού, διευκολύνει τον εντοπισμό του πιο κατάλληλου αλγορίθμου κάθε φορά ώστε η συσταδοποίηση να πραγματοποιείται με μεγαλύτερη επιτυχία.

Μέρος της αξιολόγησης αποτελούν εξίσου τα γραφήματα και τα σχεδιαγράμματα που παράγονται και παρουσιαστήκαν παραπάνω, δίνοντας έναν πιο απλό και οπτικό τρόπο εκτίμησης των αλγορίθμων ωστόσο δε συνιστώνται σε σύνολα μεγάλων διαστάσεων αφού μπορεί εύκολα να χαθεί η ακρίβεια.

Για τον ποιοτικό έλεγχο των αλγορίθμων υπάρχουν δύο τρόποι, η εσωτερική και η εξωτερική αξιολόγηση. Στην πρώτη περίπτωση, η συσταδοποίηση εκτιμάται με βάση την δομή των συστάδων που παράγει και των δεδομένων που εμπεριέχονται σε κάθε μία αυτές χωρίς να χρειάζεται το σύστημα να γνωρίζει την κατηγοριοποίηση τους σε ετικέτες. Αντίθετα, στην εξωτερική αξιολόγηση, το σύστημα εκμεταλλεύεται τις ήδη υπάρχον κατηγορίες των δεδομένων και τις συγκρίνει με τα παραγόμενα. Επειδή οι συστάδες δεν διαθέτουν συγκεκριμένη κατηγορία αλλά λαμβάνουν σημεία σύμφωνα με την ομοιογένεια τους, η εξωτερική αξιολόγηση λειτουργεί εντοπίζοντας σε αυτά τον τύπο που εμφανίζεται περισσότερα από το πλήθος των σημείων κάθε μιας. Παραδείγματα μετρικών αξιολόγησης για κάθε εκδοχή αποτελούν η F-Measure, η οποία χρησιμοποιεί τις ετικέτες για να διακρίνει αν το μοντέλο που δημιούργησε τις πρόβλεψε σωστά αλλά και η Silhouette score που εκτιμάει την εσωτερική δομή των συστάδων ελέγχοντας τις αποστάσεις των δεδομένων.[25]

Προκειμένου να αξιολογήσουμε την ποιότητα των αλγορίθμων που βασίζονται στην πυκνότητα που παρουσιάστηκαν παραπάνω και οι οποίοι υλοποιήθηκαν για κάθε σύνολο δεδομένων που μελετήθηκε στην παρούσα εργασία, χρησιμοποιήθηκε η μετρική Silhouette Score. Για την ποιότητα της συσταδοποίησης σύμφωνα με το μέτρο αυτό, προσδιορίζεται η απόσταση που εμφανίζουν οι συστάδες ενός συνόλου μεταξύ τους αλλά και η απόσταση μεταξύ των σημείων εντός αυτών.

Συγκεκριμένα, η Silhouette score αναφέρεται σε έναν μαθηματικό τύπο με τον οποίο για κάθε σημείο x του χώρου, υπολογίζεται μια μέση τιμή που αφορά την απόσταση που χρειάζεται να διανύσει κάθε σημείο της συστάδας στην οποία ανήκει για να φτάσει σε αυτό σε σχέση με την μέση απόσταση που χρειάζεται το ίδιο για να φτάσει σε οποιοδήποτε άλλο σημείο τις πιο κοντινής του συστάδας. Όταν δεν υπάρχει δεύτερη ομάδα στο χώρο για να υπολογιστεί το δεύτερο σκέλος του τύπου, τότε η μέση τιμή της απόστασης μεταξύ σημείου και συστάδας παραλείπεται ή ορίζεται ως 0 για να συνεχίσει η διαδικασία. Αφού γίνει αυτή η μέτρηση και βρεθεί ένα αποτέλεσμα για κάθε σημείο του χώρου, ακολουθεί η αξιολόγηση της συνολικής συσταδοποίησης. Το τελικό Silhouette score ορίζεται ως ο μέσος όρος όλων των σκορ σημείων που υπολογίστηκαν για το εκάστοτε dataset και υπάρχουν τρείς περιπτώσεις ερμηνείας του:

-Αν ο μέσος όρος έχει τιμή κοντά στο 1, τότε η συσταδοποίηση θεωρείται επιτυχημένη και υψηλής ποιότητας. Αυτό δηλώνει ότι η συστάδα είναι πυκνή αφού οι αποστάσεις μεταξύ των σημείων είναι μικρές ενώ η διαφορά της σε σχέση με τις υπόλοιπες ομάδες είναι μεγάλη, με τις θέσεις τους να βρίσκονται αρκετά απομακρυσμένες.
-Αν ο μέσος όρος έχει τιμή κοντά στο 0, τότε η συσταδοποίηση δεν θεωρείται απόλυτα επιτυχημένη και η ποιότητα της προσδιορίζεται ως μέτρια. Στην περίπτωση αυτή, τα σημεία εντοπίζονται πιο αραιά εντός του cluster τους με πολλά από αυτά να βρίσκονται στα όρια δύο διαφορετικών, γεγονός που οδηγεί σε ένα μη καλά διαχωρισμένο σύνολο. Ακόμη, η απόσταση τους από το κέντρο της συστάδας στην οποία έχουν τοποθετηθεί, τείνει να μοιάζει με την απόσταση που έχουν από την κοντινότερη προς αυτά συστάδα.
-Αν ο μέσος όρος έχει αρνητική τιμή -1, τότε η συσταδοποίηση θεωρείται αποτυχημένη με τα σημεία των συστάδων να έχουν τοποθετηθεί λανθασμένα. Η περίπτωση αυτή φανερώνει ότι η απόσταση των σημείων εντός των ομάδων είναι μικρότερη σε σχέση με την απόσταση τους από μια άλλη συστάδα του χώρου.

Στην περίπτωση που τα αποτελέσματα δεν αντικατοπτρίζουν μια επιτυχημένη συσταδοποίηση των δεδομένων, οι παράγοντες που μπορεί να επηρεάσουν αυτές τις τιμές και ενδεχομένως να βελτιώσουν το συνολικό σκορ είναι πολλοί. Πρώτη και κυριότερη αλλαγή αποτελεί η διαμόρφωση των παραμέτρων που χρησιμοποιήθηκαν σε κάθε αλγόριθμο και η απομάκρυνση του θορύβου που μπορεί να προσδιορίστηκε λανθασμένα σε συστάδες. Επιπλέον, αν υπάρχουν χαρακτηριστικά που δεν προσφέρουν σημαντικά στην συσταδοποίηση θα μπορούσαν να αφαιρεθούν ενώ αν τίποτα από τα παραπάνω δε συμβάλει στην αύξηση της τιμής του Silhouette score τότε η αλλαγή αλγορίθμου αποτελεί μονόδρομος.[23] Παρακάτω δίνονται οι πίνακες με την τελική αξιολόγηση των dataset που χρησιμοποιήθηκαν για τη μελέτη καθώς και το σύνολο των συστάδων που δημιουργήθηκαν αλλά και των σημείων που προσδιορίστηκαν ως θόρυβος για καθένα από αυτά:

	DBSCAN			DENCLUE			MEAN SHIFT		
	CLUSTER	OUTLIER	SILHOUETTE SCORE	CLUSTER	OUTLIER	SILHOUETTE SCORE	CLUSTER	OUTLIER	SILHOUETTE SCORE
APPENDICITIS	2	14	0.40	3	0	0,09	4	0	0,39
BANANA	1	69	0,33	2	0	0,39	2	0	0,39
BUPA	1	23	0,65	3	0	0,16	12	0	0,35
GLASS	2	46	0,52	4	0	0,32	17	0	0,40
HABERMAN	3	24	0,19	5	0	0,18	3	0	0,52
IRIS	2	17	0,49	3	0	0,37	2	0	0,58
MAGIC	2	2925	0,17	3	0	0,27	40	0	0,37
NEWTHYROID	2	38	0,38	2	0	0,45	10	0	0,49
PIMA	3	21	0,20	2	0	0,12	5	0	0,26
TAE	8	35	0,16	27	0	0,53	5	0	0,36
VEHICLE	7	86	0,23	2	0	0,21	3	0	0,56
WINE	5	36	0,30	4	0	0,25	2	0	0,27
AVG SCORE			0,33			0,28			0,41

Πίνακας 14. Πλήρης πίνακας Cluster, Outlier και Silhouette Score για τους αλγορίθμους DBSCAN, DENCLUE, Mean Shift

	OPTICS			HDBSCAN		
	CLUSTER	OUTLIER	SILHOUETTE SCORE	CLUSTER	OUTLIER	SILHOUETTE SCORE
APPENDICITIS	2	70	0,06	2	33	0,28
BANANA	1	18	0,39	2	332	0,17
BUPA	1	20	0,54	3	6	0,50
GLASS	1	12	0,56	2	40	0,42
HABERMAN	1	57	0,33	3	50	0,11
IRIS	1	50	0,58	2	2	0,49
MAGIC	1	5	0,74	2	1657	0,44
NEWTHYROID	1	13	0,65	2	26	0,45
PIMA	1	24	0,42	3	70	0,26
TAE	10	56	0,23	4	3	0,36
VEHICLE	1	12	0,52	2	1	0,60
WINE	1	5	0,24	2	58	0,14
AVG SCORE			0,44			0,35

Πίνακας 15.Πλήρης πίνακας Cluster, Outlier και Silhouette Score για τους αλγορίθμους OPTICS, HDBSCAN

Εξετάζοντας κάθε σύνολο δεδομένων ξεχωριστά για κάθε αλγόριθμο και έπειτα από τους πειραματισμούς που πραγματοποιήθηκαν, οι εκτιμήσεις που προκύπτουν είναι οι εξής: Για το Appendicitis, οι DBSCAN και Mean Shift φαίνεται να είναι οι αλγόριθμοι που λειτούργησαν καλύτερα στο συγκεκριμένο σύνολο δεδομένων όμως και οι πέντε προσφέρουν μια μέτριας ποιότητας συσταδοποίησης αφού τα σκορ τους είναι αρκετά μικρά και πολύ κοντά στο 0. Συνεχίζοντας στο Banana, κανένας αλγόριθμος δε φαίνεται να έχει κάνει τέλεια δουλεία με όλους να παρέχουν μια μέτρια συσταδοποίηση στα δεδομένα του. Οι DENCLUE, OPTICS και Mean Shift αποτελούν ίσως την καλύτερη επιλογή για το dataset με κοινή τιμή. Από την άλλη πλευρά το Bupa, εμφανίζεται ως ένα σύνολο δεδομένων που εκτελεί μια υψηλής ποιότητας συσταδοποίηση όταν χρησιμοποιούνται οι αλγόριθμοι DBSCAN, OPTICS και HDBSCAN με τις συγκεκριμένες παραμέτρους, γεγονός που τους χρήζει κατάλληλους για το συγκεκριμένο dataset. Προχωρώντας στο Glass, διαπιστώνουμε ότι για την ιδανικότερη συσταδοποίηση συνιστώνται οι DBSCAN και OPTICS με τα σκορ τους να κυμαίνονται πιο κοντά στο 1 ενώ για το αρχείο Haberman, ο Mean Shift είναι η πιο επωφελής επιλογή. Ακολούθως ο Iris, εξασφαλίζει σε πολλές περιπτώσεις μια καλή ομαδοποίηση των δεδομένων με τιμές να ξεπερνούν το 0.50 όπως στον OPTICS και Mean Shift αλλά και σκορ που βρίσκονται στο όριο μιας καλής συσταδοποίησης όπως ο DBSCAN και ο HDBSCAN. Ο MAGIC με ένα σύνολο 19.020 δειγμάτων φαίνεται να προσφέρει μια εξαιρετική ομαδοποίηση με την μέγιστη τιμή σκορ 0.74 για τον αλγόριθμο OPTICS ενώ εξίσου δυνατή στο συγκεκριμένο αλγόριθμο είναι και η συσταδοποίηση που πραγματοποιήθηκε για το dataset New Thyroid με τιμή 0.65 με κοντά στο όριο να βρίσκονται και οι υπόλοιποι αλγόριθμοι. Στο αντίθετο πλαίσιο, οι δοκιμές που πραγματοποιήθηκαν στο dataset PIMA πάνω σε όλους τους αλγορίθμους που βασίζονται στην πυκνότητα δε φαίνεται να ευδοκίμησαν, προσφέροντας μια μέτριας ποιότητας συσταδοποίηση στα σημεία με καλύτερη αυτών να αποτελεί η διάκριση που πραγματοποίησε ο OPTICS. Επιπλέον, η μοναδική περίπτωση όπου ο DENCLUE φαίνεται να λειτούργησε επιτυχημένα είναι αυτή του dataset TAE με ένα σκορ της τάξης του 0.53 ενώ για το Vehicle τα κατάφεραν τρείς από αυτούς, οι OPTICS, Mean Shift και HDBSCAN. Τελικό σύνολο δεδομένων αποτελεί το Wine με καμία πλήρης επιτυχία αλλά με τον DBSCAN να είναι ο μόνος που παρείχε την μέγιστη δυνατή ποιότητας συσταδοποίησης με τιμή 0.30. Συνολικά με μια ματιά καθώς και βάση του μέσου όρου κάθε αλγορίθμου προς όλα τα σύνολα δεδομένων που χρησιμοποιήθηκαν, φτάνουμε στο συμπέρασμα ότι με μικρή διαφορά ο αλγόριθμος OPTICS, είναι αυτός που επικρατεί με τη μεγαλύτερη τιμή σκορ αλλά και με τις περισσότερες επιτυχημένες υψηλής ποιότητας ομαδοποιήσεις.

Εν κατακλείδι, τα αποτελέσματα που παρουσιάστηκαν, παρείχαν χρήσιμες πληροφορίες για την κατανόηση της πρακτικής χρήσης του Silhouette score ως τρόπο αξιολόγησης των αλγορίθμων σε διάφορα σύνολα δεδομένων. Η σύγκριση των τιμών σκορ που πραγματοποιήθηκε μεταξύ των πέντε αλγορίθμων και σε κάθε dataset ανέδειξε τις περιπτώσεις όπου η συσταδοποίηση πραγματοποιήθηκε

με επιτυχία ή άγγιζε τα όρια αλλά και τις καταστάσεις όπου οι δοκιμές δεν έφεραν τα αναμενόμενα αποτελέσματα και ίσως μια περαιτέρω πειραματική διερεύνηση να ήταν αναγκαία.

Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

Σε μια εποχή όπου η γνώση που εξάγεται από την ανάλυση των δεδομένων αποκτά συνεχώς μεγαλύτερη αξία, αποτελεί πρόκληση να μπορεί κανείς να κατανοεί και να διαχειρίζεται τις διάφορες τεχνικές συσταδοποίησης που υφίστανται. Η τεχνολογία εξελίσσεται και μαζί της πορεύεται και ο διαχωρισμός των δεδομένων με μεγάλη μελλοντική ανάπτυξη. Η μελέτη που πραγματοποιήθηκε ανέδειξε τη σημασία της συσταδοποίησης τουλάχιστον από την πλευρά της πυκνότητας και απέδειξε πως ο κάθε αλγόριθμος έχει τις δικές του ιδιότητες, με κάθε αποτέλεσμα που παράγεται να έχει πάντα να προσφέρει κάποια πληροφορία. Όμως όποια τεχνική και αν αποφασιστεί εν τέλει, η εύρεση του καταλληλότερου αλγορίθμου δε θα είναι ποτέ μια «τυχαία» επιλογή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΕΙΚΟΝΑ ΕΞΩΦΥΛΛΟΥ

https://medium.com/@ashutoshkumbhare/3-powerful-clustering-algorithms-in-machine-learning-12a90d5d43e

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

[1] Ράπτης Σάββας "*Τι είναι η μηχανική μάθηση (machine learning); – Μέρος Α: "Εισαγωγή*" 2' science, 10. Οκτ. 2021, https://2science.gr/machine-learning-1/

[2] Sara Brown "Machine learning, explained" MIT Management Sloan School, 21. Apr. 2021, https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

[3] "*What is clustering?*" IBM, 21. Feb. 2024, https://www.ibm.com/think/topics/clustering

[4] Gopi Gandhi and Rohit Srivastava "*A Comparative Study on Partitioning Techniques of Clustering Algorithms*" ResearchGate, Feb. 2014, https://www.researchgate.net/publication/263028213_Review_Paper_A_Comparative_Study_on_Partitioning_Techniques_of_Clustering_Algorithms

[5] Yogita Rani and Dr. Harish Rohil "*A Study of Hierarchical Clustering Algorithm*" Research India Publications, 11. Nov. 2013, https://www.ripublication.com/irph/ijict_spl/14_ijictv3n11spl.pdf

[6] Rupanka Bhuyan and Samarjeet Borah "*A Survey of Some Density Based Clustering Techniques*" Cornell University, https://arxiv.org/pdf/2306.09256

[7] Rajesh Kumar "*A Guide to the DBSCAN Clustering Algorithm*" datacamp, 29. Sep. 2024, https://www.datacamp.com/tutorial/dbscan-clustering-algorithm

[8] Martin Ester, et al. "*A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*" Ludwig Maximilian University of Munich (LMU), https://www.dbs.ifi.lmu.de/Publikationen/Papers/KDD-96.final.frame.pdf

[9] Lifeng Yin, et al. "*Improvement of DBSCAN Algorithm Based on K-Dist Graph for Adaptive Determining Parameters"* MDPI, 25. Jul. 2023, https://www.mdpi.com/2079-9292/12/15/3213

[10] "*Understanding DENCLUE: Density-Based Clustering Algorithm In Distribution Functions*" JanBask Training, https://www.janbasktraining.com/tutorials/density-based-clustering/

[11] Alexander Hinneburg and Hans-Henning Gabriel "*DENCLUE 2.0: Fast Clustering based on Kernel Density Estimation*" Universität Halle, https://users.informatik.uni-halle.de/~hinnebur/PS_Files/denclue2_0_ida2007.pdf

[12] "*Mean Shift*" Machine Learning Explained, 30. Nov. 2020, https://ml-explained.com/blog/mean-shift-explained

[13] Dorin Comaniciu and Peter Meer "*Mean Shift Analysis and Applications*" Dorin Comaniciu, https://comaniciu.net/Papers/MsAnalysis.pdf

[14] Mihael Ankerst, et al. "*OPTICS: Ordering Points To Identify the Clustering Structure*" Ludwig Maximilian University of Munich (LMU), 1. Jun. 1999, https://www.dbs.ifi.lmu.de/Publikationen/Papers/OPTICS.pdf

[15] "*OPTICS algorithm*" WIKIPEDIA https://en.wikipedia.org/wiki/OPTICS_algorithm

[16] Leland McInnes, et al. "*How HDBSCAN Works*" The hdbscan Clustering Library, 2023, https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

[17] "*Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)"* GeeksforGeeks, 14. Mar. 2024, https://www.geeksforgeeks.org/hdbscan/
[18] "*Scikit-Learn vs TensorFlow: Which One to Choose?"* Analytics Vidhya, 14. Aug. 2023, https://www.analyticsvidhya.com/blog/2023/08/scikit-learn-and-tensorflow/
[19] "*scikit-learn. Machine Learning in Python"* Scikitlearn, https://scikit-learn.org/stable/
[20] "*Browse Datasets*" UC Irvine Machine Learning Repository, https://archive.ics.uci.edu/datasets
[21] Mike Yi "*A complete guide to scatter plots*" ATLASSIAN, https://www.atlassian.com/data/charts/what-is-a-scatter-plot
[22] "*What it a Parallel Coordinate Plot?"* JASPERSOFT, https://www.jaspersoft.com/articles/what-is-a-parallel-coordinate-plot
[23] "*What is Silhouette Score?"* educative, https://www.educative.io/answers/what-is-silhouette-score
[24] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar 2η Έκδοση (2018) "*Εισαγωγή στην Εξόρυξη Δεδομένων*" Εκδόσεις ΤΖΙΟΛΑ – Κεφάλαιο 7ο και 8ο
[25] Mohammed J.Zaki Wagner Meira Jr. (2017) "*Εξόρυξη και ανάλυση δεδομένων. Βασικές έννοιες και αλγόριθμοι*" Εκδόσεις Κλειδάριθμος – Μέρος 3ο
[26] Μ. Χαλκίδη – Μ. Βαζιργιάννης 2η Έκδοση (2005) "*Εξόρυξη γνώσης από βάσεις δεδομένων και τον παγκόσμιο ιστό*" Εκδόσεις Τυπωθήτω / Δαρδάνος – 3ο Κεφάλαιο