ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ : ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΝΙΩΝ

ΤΜΗΜΑ: ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	Αλγόριθμοι Ομαδοποίησης Χρηστών σε Συστήματα Σύστασης. User Clustering Algorithms in Recommendation Systems.
Ονοματεπώνυμο	Φώτης Σιούζιος
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	Π17121
Επιβλέπων	Σωτηρόπουλος Διονύσιος

Ημερομηνία Παράδοσης Σεπτέμβριος 2024

Διμελής Εξεταστική Επιτροπή

Δ. Σωτηρόπουλος (Επίκουρος Καθηγητής) Γ. Τσιχριντζής (Καθηγητής)

Copyright ©

Ευχαριστίες:

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου, Διονύσιο Σωτηρόπουλο, για την καθοριστική του συμβολή στην επιτυχή ολοκλήρωση αυτής της πτυχιακής εργασίας. Η πολύτιμη καθοδήγησή του, η υπομονή και οι συμβουλές του σε κάθε στάδιο της έρευνας υπήρξαν ανεκτίμητες. Η στήριξη του με βοήθησε να ξεπεράσω τις προκλήσεις και να εμβαθύνω στο αντικείμενο της μελέτης.

Πίνακας περιεχομένων

Περίληψη :

Η εργασία αυτή εξετάζει τα συστήματα σύστασης, εστιάζοντας σε διάφορες μεθόδους και αλγόριθμους που χρησιμοποιούνται για την ομαδοποίηση και σύσταση δεδομένων. Αρχικά, παρουσιάζεται μια επισκόπηση των υπαρχόντων συστημάτων σύστασης, περιγράφοντας τις διάφορες κατηγορίες και τις λειτουργίες τους. Στη συνέχεια, αναλύονται συγκεκριμένοι αλγόριθμοι ομαδοποίησης, οι οποίοι αποτελούν βασικά εργαλεία για τη δημιουργία αποτελεσματικών συστημάτων σύστασης.

Ένα σημαντικό μέρος της εργασίας αφορά την προεπεξεργασία των δεδομένων, όπου αναλύεται πώς καθαρίζονται, μετασχηματίζονται και προετοιμάζονται τα δεδομένα για να εισαχθούν στους αλγόριθμους ομαδοποίησης. Τέλος, παρουσιάζονται τα δικά πειράματα σε αυτούς τους αλγορίθμους, όπου εφαρμόζεις τις μετρικές Pearson correlation, cosine similarity και euclidean distance στον αλγόριθμο K-means. Μέσα από αυτή την εφαρμογή, επιδεικνύεται η πρακτική χρήση αυτών των μετρικών για τη βελτίωση της απόδοσης του αλγορίθμου και την επίτευξη καλύτερων αποτελεσμάτων ομαδοποίησης.

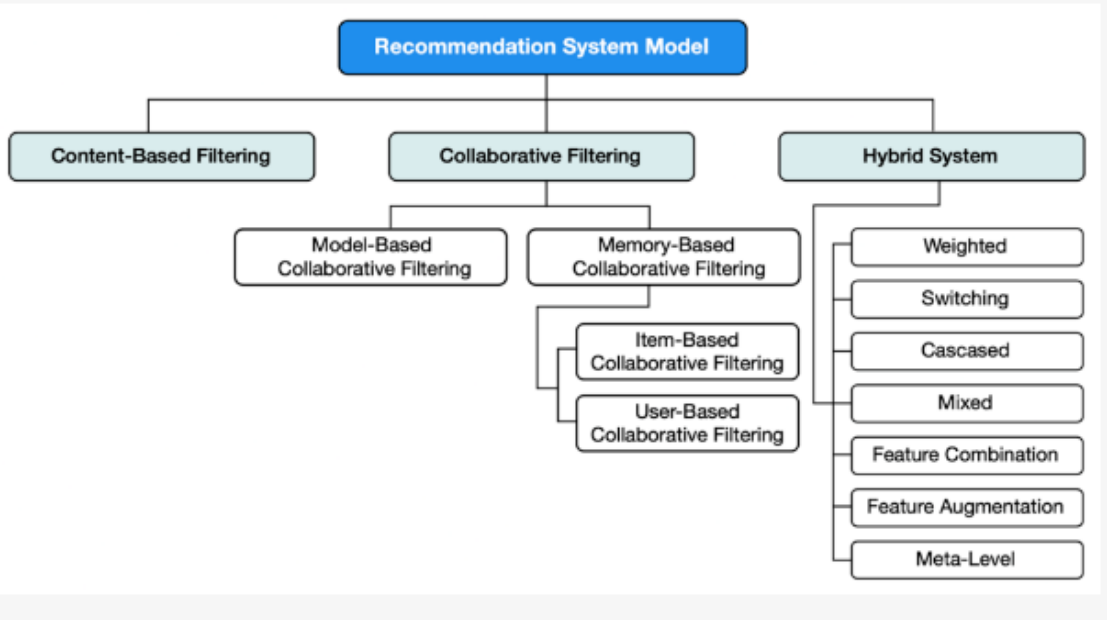
Abstract :

My thesis examines recommender systems, focusing on various methods and algorithms used to cluster and recommend data. First, an overview of existing recommender systems is presented, describing their various categories and functions. Then, specific clustering algorithms, which are key tools for creating effective recommender systems, are discussed.
An important part of the paper deals with data preprocessing, where it is discussed how data is cleaned, transformed and prepared for input to the clustering algorithms. Finally, we present our own experiments on these algorithms, where we apply the Pearson correlation, cosine similarity and euclidean distance metrics to the K-means algorithm. Through this application, we demonstrate the practical use of these metrics to improve the performance of the algorithm and achieve better clustering results.

Κεφάλαιο 1 : Συστήματα Σύστασης Μεθοδολογίες Υπάρχουσες

Εισαγωγή

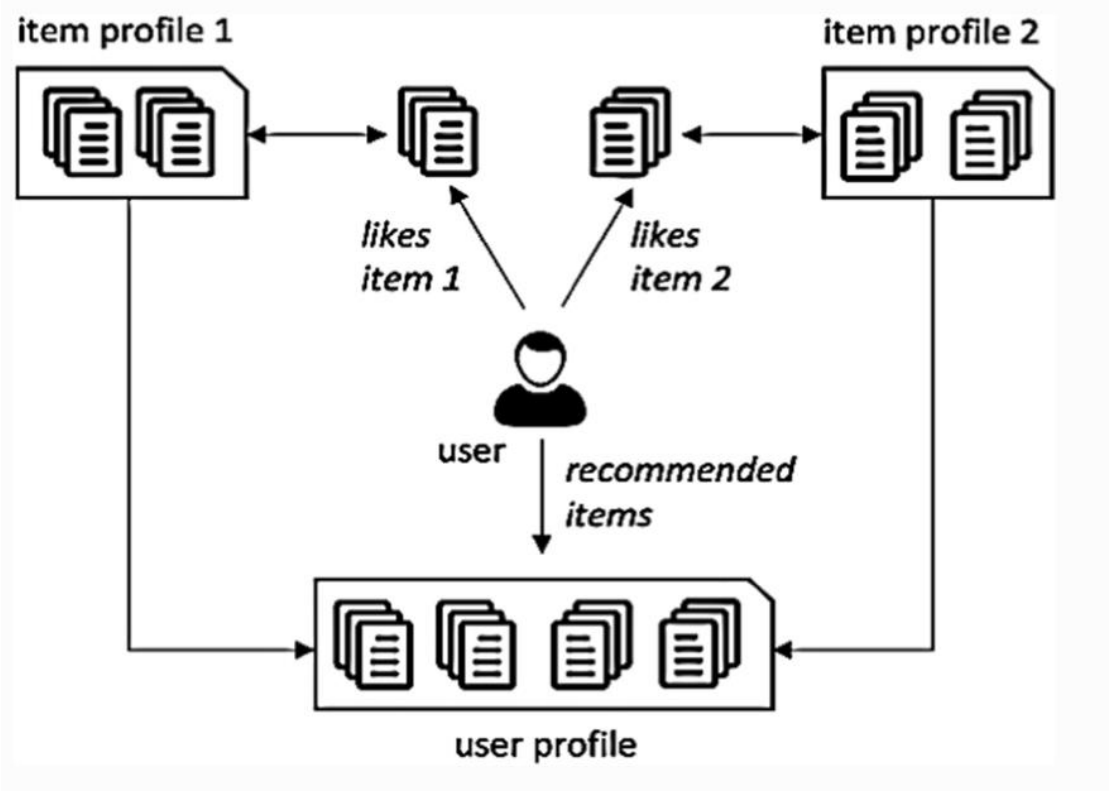
Το διαδίκτυο και οι σύγχρονες υπηρεσίες ιστού έχουν αυξηθεί τις τελευταίες δεκαετίες- ένα πλεόνασμα πληροφοριών είναι πλέον προσβάσιμο σε όλους . Μπορεί να είναι δύσκολο για τους χρήστες να φιλτράρουν όλες αυτές τις πληροφορίες και να παίρνουν τις ουσιαστικές πτυχές. Πολλές εταιρείες ηλεκτρονικού εμπορίου στο διαδίκτυο προτείνουν προϊόντα στους χρήστες τους, πουλώντας εκατομμύρια προϊόντα σε μία πλατφόρμα. Για έναν καθημερινό χρήστη, η περιήγηση σε όλες τις δυνατότητες μπορεί να είναι συντριπτική- αυτό μπορεί να προκαλέσει υπερφόρτωση πληροφοριών[3].Τα συστήματα συστάσεων είναι συστήματα φιλτραρίσματος πληροφοριών που παρέχουν μια εξατομικευμένη σύσταση στοιχείων σε έναν χρήστη, σε ένα περιβάλλον υπηρεσιών που μπορεί να κατέχει ή να συλλέγει διάφορα δεδομένα. Είναι μια χρήσιμη τεχνολογία που μπορεί να ανακουφίσει το πρόβλημα της υπερφόρτωσης των πληροφοριών που παρέχονται στους χρήστες. Προβλέπει τον βαθμό των στοιχείων που θα προταθούν στον χρήστη, δημιουργεί έναν κατάλογο κατάταξης συστάσεων για κάθε χρήστη και καθιστά δυνατή τη σύσταση στοιχείων που σχετίζονται με τον χρήστη. Αρκετές υπηρεσίες πλατφόρμας προτείνουν ενεργά εξατομικευμένα αντικείμενα που ανταποκρίνονται στις ανάγκες των χρηστών εισάγοντας ένα σύστημα συστάσεων. Προκειμένου να βελτιωθεί η απόδοση αυτών των συστάσεων, διεξάγονται μελέτες σχετικά με διάφορα μοντέλα φιλτραρίσματος συστάσεων και τεχνικές εξόρυξης δεδομένων [1]. Πρόσφατα, έχουν αναπτυχθεί διάφορες προσεγγίσεις για τη δημιουργία συστημάτων συστάσεων, οι οποίες μπορούν να χρησιμοποιήσουν είτε συνεργατικό φιλτράρισμα, είτε φιλτράρισμα βάσει περιεχομένου είτε υβριδικό φιλτράρισμα [2]



Εικονα 1 : Επισκόπηση των μοντέλων σύστασης.

1.1 Φιλτράρισμα βάση περιεχομένου (Content-Based Filtering)

Το 1992, ξεκινώντας από τη μελέτη των Loeb, εμφανίστηκαν διάφορα μοντέλα φιλτραρίσματος πληροφοριών. Το φιλτράρισμα βάσει περιεχομένου είναι μια μέθοδος για τη σύσταση στοιχείων με χαρακτηριστικά παρόμοια με αυτά που αρέσουν στους χρήστες και τα συστήνει με βάση τις πληροφορίες των στοιχείων. Το μοντέλο φιλτραρίσματος βάσει περιεχομένου είναι το πιο βασικό μοντέλο στο πλαίσιο του συνολικού μοντέλου συστήματος συστάσεων και χρησιμοποιήθηκε κυρίως στα πρώτα συστήματα συστάσεων [1]. Η τεχνική που βασίζεται στο περιεχόμενο είναι ένας αλγόριθμος που εξαρτάται από τον εκάστοτε τομέα και δίνει μεγαλύτερη έμφαση στην ανάλυση των χαρακτηριστικών των στοιχείων, προκειμένου να δημιουργήσει προβλέψεις. Όταν πρόκειται να προταθούν έγγραφα όπως ιστοσελίδες, δημοσιεύσεις και ειδήσεις, η τεχνική φιλτραρίσματος βάσει περιεχομένου είναι η πιο επιτυχημένη. Στην τεχνική φιλτραρίσματος βάσει περιεχομένου η σύσταση γίνεται με βάση τα προφίλ του χρήστη χρησιμοποιώντας χαρακτηριστικά που εξάγονται από το περιεχόμενο των στοιχείων που ο χρήστης έχει αξιολογήσει στο παρελθόν [2]. Ωστόσο, ως αποτέλεσμα της μελέτης των Salter κ.ά. , το μοντέλο φιλτραρίσματος βάσει περιεχομένου συνιστά μόνο δεδομένα που σχετίζονται με στοιχεία που έχουν αξιολογηθεί προηγουμένως από τον χρήστη, οπότε το σύστημα είναι γνωστό για τον περιορισμό του στο ότι δεν μπορεί να συστήσει νέα στοιχεία [1].



Εικονα 2 : Content-Based recommender system.

1.1.1 Θετικά και Αρνητικά του μοντέλου σύστασης βάση περιεχομένου.

Καθώς οι προτιμήσεις των χρηστών τείνουν να αλλάζουν με τον χρόνο, η προσέγγιση αυτή έχει τη γρήγορη ικανότητα να προσαρμόζεται δυναμικά στις μεταβαλλόμενες προτιμήσεις των χρηστών. Δεδομένου ότι ένα προφίλ χρήστη είναι συγκεκριμένο μόνο για τον συγκεκριμένο χρήστη, ο αλγόριθμος αυτός δεν απαιτεί τα στοιχεία του προφίλ άλλων χρηστών, επειδή δεν παρέχουν καμία επιρροή στη διαδικασία σύστασης. Αυτό διασφαλίζει την ασφάλεια και την ιδιωτικότητα των δεδομένων των χρηστών. Εάν τα νέα στοιχεία έχουν επαρκή περιγραφή, οι τεχνικές που βασίζονται στο περιεχόμενο μπορούν να ξεπεράσουν το πρόβλημα της ψυχρής εκκίνησης (cold start), δηλαδή η τεχνική αυτή μπορεί να συστήσει ένα στοιχείο ακόμη και όταν το στοιχείο αυτό δεν έχει βαθμολογηθεί προηγουμένως από κανέναν χρήστη. Οι προσεγγίσεις φιλτραρίσματος με βάση το περιεχόμενο είναι πιο συνηθισμένες σε συστήματα σύστασης εξατομικευμένων ειδήσεων, δημοσιεύσεων, συστημάτων σύστασης ιστοσελίδων κ.λπ. [4].

Αναφορικά με τους περιορισμούς της σύστασης βάση περιεχομένου, αντιμετωπίζουμε το πρόβλημα έλλειψης πληροφοριών. Δηλαδή όταν ένας νέος χρήστης εισάγεται στο σύστημα και δεν έχουμε επαρκή δεδομένα για το προφίλ και

τις προτιμήσεις του, η σύσταση που θα κάνει το σύστημα θα είναι αδύναμη και ελλιπής. Ένα ακόμα ζήτημα με το οποίο μπορούμε να έρθουμε αντιμέτωποι με την σύσταση βάση περιεχομένου είναι το πρόβλημα της υπερεξειδίκευσης. Επειδή τα συστήματα συστάσεων συνιστούν μόνο τα στοιχεία ή προϊόντα που έχουν αξιολογηθεί στο παρελθόν. Συνεπώς το σύστημα δεν συνιστά αυτά τα αντικείμενα που είναι διαφορετικά από οτιδήποτε έχει αξιολογήσει ο χρήστης στο παρελθόν, γεγονός που μπορεί να αποτελέσει πρόβλημα επειδή ο χρήστης μπορεί να θέλει να δοκιμάσει κάτι νέο και το σύστημα δεν έχει τα δεδομένα για να του το προτείνει [5] .

Συνοψίζοντας τα κύρια πλεονεκτήματα ενός συστήματος συστάσεων με βάση το περιεχόμενο περιλαμβάνουν τη διαφάνεια, την ανεξαρτησία και τις συστάσεις για μη ταξινομημένες οντότητες. Στα μειονεκτήματα περιλαμβάνονται η τυχαία επιλογή, η μερική ανάλυση περιεχομένου και η υπερεξειδίκευση [6].

1.1.2 Μέθοδοι που χρησιμοποιούνται στο μοντέλο σύστασης βάση περιεχομένου.

TF-IDF :

Είναι μια αριθμητική στατιστική που έχει σκοπό να αντικατοπτρίζει πόσο σημαντική είναι μια λέξη σε ένα έγγραφο ή σε μια συλλογή. TF-IDF (term frequency - inverse document frequency) χρησιμοποιείται για να δώσει βαρύτητα σε συγκεκριμένες λέξεις στην ανάκτηση πληροφοριών, η οποία έχει ως εξής: Έστω ότι N είναι ο συνολικός αριθμός των εγγράφων που μπορούν να συνιστώνται στους χρήστες και ότι η λέξη-κλειδί kj εμφανίζεται στα ni των αυτά. Επιπλέον, υποθέστε ότι fi,j είναι ο αριθμός των φορών που η λέξη-κλειδί ki εμφανίζεται στο έγγραφο di. Έτσι, η συχνότητα του όρου (ή κανονικοποιημένη συχνότητα) TFi,j της λέξης-κλειδί ki στο έγγραφο dj, είναι ορίζεται ως εξής:

$$TF_{i,j} = \frac{f_{i,j}}{max_z f_{z,j}}$$

κατά τον υπολογισμό του TF, όλοι οι όροι θεωρούνται εξίσου σημαντικοί. Ωστόσο, είναι γνωστό ότι ορισμένοι όροι, όπως συνδετικές λέξεις, μπορεί να εμφανίζονται πολλές φορές αλλά έχουν μικρή σημασία. Συνεπώς, πρέπει να σταθμίσουμε τους συχνούς όρους ενώ να αναβαθμίζει τους σπάνιους, υπολογίζοντας τον αντίστροφο συχνότητας εγγράφων ($IDFi$) για τη λέξη-κλειδί ki ορίζεται ως εξής:

$$IDF_i = \log \frac{N}{n_i}$$

τότε, το βάρος TF-IDF για τη λέξη-κλειδί ki στο έγγραφο dj είναι ορίζεται ως εξής:

$$w_{i,j} = TF_{i,j} \times IDF_i$$

Το περιεχόμενο του εγγράφου dj ορίζεται ως εξής:

$$Content(d_j) = (w_{1j}, \ldots., w_{kj})$$

Naïve Bayes:

Εκτός από την προηγούμενη παραδοσιακή ευρετική προσέγγιση, το φιλτράρισμα με βάση το περιεχόμενο μπορεί να γίνει μέσω ταξινομητών Bayes και διάφορες τεχνικές μηχανικής μάθησης, συμπεριλαμβανομένης της ομαδοποίησης, δέντρα αποφάσεων και τεχνητά νευρωνικά δίκτυα. Ο ταξινομητής Bayesian χρησιμοποιείται για την εκτίμηση της πιθανότητας ότι ένα έγγραφο είναι αρεστό ή όχι. Ο ταξινομητής Naïve Bayes είναι ο ταξινομητής μηχανικής μάθησης, οικογένεια απλών πιθανοτικών ταξινομητών που βασίζονται στην εφαρμογή του θεωρήματος Bayes με ισχυρές υποθέσεις ανεξαρτησίας. μεταξύ των χαρακτηριστικών. Ο ταξινομητής Naive Bayes χρησιμοποιείται για να να εκτιμήσει την ακόλουθη πιθανότητα ότι η ιστοσελίδα pj ανήκει σε μια συγκεκριμένη κλάση Ci (π.χ., σχετική ή άσχετη) δεδομένου του συνόλου των λέξεων-κλειδιών $k1,j$; . . . ; kn,j στην εν λόγω ιστοσελίδα ως:

$$P(C_i|k_{1j} \,\&\, \ldots. \,\&\, k_{nj)}$$

Εδώ υποθέτουμε ότι οι λέξεις-κλειδιά είναι ανεξάρτητες και, ως εκ τούτου, η παραπάνω πιθανότητα είναι ανάλογη με:

$$P(C_i)\prod_{x} P(k_{x,j}|C_i)$$

Επιπλέον, τόσο το P (kx,i |Ci) και P (Ci) μπορούν να εκτιμηθούν από τα υποκείμενα δεδομένα εκπαίδευσης. Επομένως, για κάθε σελίδα pj, η πιθανότητα P (Ci|k1,j & . . .& xn,i) υπολογίζεται για κάθε κλάση. Οι μέθοδοι CF μπορούν να υποδιαιρεθούν περαιτέρω σε προσεγγίσεις που βασίζονται στη γειτονιά και σε προσεγγίσεις που βασίζονται σε μοντέλα. Με βάση τη γειτονιά μέθοδοι αναφέρονται επίσης συνήθως ως μνήμες που βασίζονται στη μνήμη προσέγγιση.

1.2 Συνεργατικό φιλτράρισμα (Collaborative Filtering)

Το σύστημα συστάσεων με βάση το συνεργατικό φιλτράρισμα, εξαρτάται από τη συλλογή και την ανάλυση των δεδομένων με βάση τις προτιμήσεις ενός χρήστη ώστε να κάνει μια πρόβλεψη αντικειμένων βάση ομοιότητάς με άλλους χρήστες, αυτή η ομοιότητα υπολογίζεται συγκρίνοντας τις αξιολογήσεις τους με τις αξιολογήσεις άλλων χρηστών για το ίδιο αντικείμενο .

Υπάρχουν δύο τύποι προσεγγίσεις συνεργατικού φιλτραρίσματος - με βάση τα αντικείμενα και με βάση τους χρήστες.

α) Σύστημα συνεργατικού φιλτραρίσματος με βάση τα αντικείμενα. Σε αυτό το σύστημα, οι σχέσεις αντικειμένων προσδιορίζονται χρησιμοποιώντας τον πίνακα user item και τον χρησιμοποιούν για τον υπολογισμό της σύστασης για τους χρήστες έμμεσα. Παρέχει στους χρήστες, ένα στοιχείο ως σύσταση, βασισμένο στα άλλα αντικείμενα με υψηλές συσχετίσεις. Πρωτίστως για κάθε στοιχείο φτιάχνετε ένα σύνολο με παρόμοια αντικείμενα « το σύνολο των γειτόνων ». Το σύστημα αυτό προβλέπει τις αξιολογήσεις του χρήστη για ένα συγκεκριμένο στοιχείο ανάλογα με τη βαθμολογία που έχει δοθεί στα παρόμοια στοιχείο από τον ίδιο χρήστη.

β) Σύστημα συνεργατικού φιλτραρίσματος με βάση τον χρήστη. Το σύστημα συνεργασίας με βάση τον χρήστη συνιστά ένα στοιχείο στο χρήστη με βάση τη γνώμη άλλων χρηστών με παρόμοιο πνεύμα για αυτό το στοιχείο. Βρίσκει τους χρήστες με κοινά ενδιαφέροντα και τους θεωρεί ως τους πλησιέστερους γείτονες. Στη συνέχεια, η πρόβλεψη αυτού του χρήστη για το στοιχείο θα γίνει με βάση τη βαθμολογία που οι γείτονες του χρήστη για το συγκεκριμένο στοιχείο.[7]

1.2.1 Θετικά και Αρνητικά του συνεργατικού φιλτραρίσματος.

Το συνεργατικό φιλτράρισμα έχει ορισμένα σημαντικά πλεονεκτήματα σε σχέση με το μοντέλο σύστασης βάση περιεχομένου, καθώς μπορεί να αποδώσει σε τομείς όπου δεν υπάρχει πολύ περιεχόμενο που σχετίζεται με τα στοιχεία και όπου το περιεχόμενο είναι δύσκολο να αναλυθεί από ένα υπολογιστικό σύστημα (όπως οι απόψεις και το ιδανικό). Επίσης, η τεχνική συνεργατικού φιλτραρίσματος έχει τη δυνατότητα να παρέχει τυχαίες συστάσεις, πράγμα που σημαίνει ότι μπορεί να προτείνει αντικείμενα που είναι σχετικά με τον χρήστη ακόμη και χωρίς το περιεχόμενο να βρίσκεται στο προφίλ του χρήστη. [2]

Παρά την επιτυχία των τεχνικών συνεργατικού φιλτραρίσματος, η ευρεία χρήση τους έχει αποκαλύψει ορισμένα πιθανά προβλήματα, όπως τα ακόλουθα:

Πρόβλημα ψυχρής εκκίνησης (Cold-start problem). Αυτό αναφέρεται σε μια κατάσταση όπου μια σύσταση δεν έχει επαρκείς πληροφορίες για έναν χρήστη ή ένα στοιχείο προκειμένου να κάνει σχετικές προβλέψεις. Πρόκειται για ένα από τα

σημαντικότερα προβλήματα που μειώνουν την απόδοση του συστήματος συστάσεων. Το προφίλ ενός τέτοιου νέου χρήστη ή στοιχείου θα είναι κενό, αφού δεν έχει αξιολογήσει κανένα στοιχείο- συνεπώς, το γούστο του δεν είναι γνωστό στο σύστημα.

Πρόβλημα αραιών δεδομένων (Data sparsity problem). Πρόκειται για το πρόβλημα που εμφανίζεται ως αποτέλεσμα της έλλειψης επαρκούς πληροφορίας, δηλαδή όταν μόνο λίγα από τον συνολικό αριθμό των στοιχείων που υπάρχουν σε μια βάση δεδομένων αξιολογούνται από τους χρήστες. Αυτό οδηγεί πάντα σε έναν αραιό πίνακα χρήστη-αντικειμένου, σε αδυναμία εντοπισμού επιτυχημένων γειτόνων και, τέλος, στη δημιουργία αδύναμων συστάσεων. Επίσης, η σπανιότητα των δεδομένων οδηγεί πάντα σε προβλήματα κάλυψης, το οποίο είναι το ποσοστό των στοιχείων του συστήματος για τα οποία μπορούν να γίνουν συστάσεις.

Κλιμάκωση (Scalability). Αυτό είναι ένα άλλο πρόβλημα που σχετίζεται με τους αλγορίθμους συστάσεων, επειδή ο υπολογισμός συνήθως αυξάνεται γραμμικά με τον αριθμό των χρηστών και των αντικειμένων. Μια τεχνική σύστασης που είναι αποτελεσματική όταν ο αριθμός των δεδομένων είναι περιορισμένος μπορεί να μην είναι σε θέση να παράγει ικανοποιητικό αριθμό συστάσεων όταν αυξάνεται ο όγκος των δεδομένων. Συνεπώς, είναι ζωτικής σημασίας να εφαρμόζονται τεχνικές συστάσεων που είναι ικανές να κλιμακώνονται με επιτυχία καθώς αυξάνεται ο αριθμός των συνόλων δεδομένων σε μια βάση δεδομένων. Οι μέθοδοι που χρησιμοποιούνται για την επίλυση του προβλήματος κλιμάκωσης και την επιτάχυνση της παραγωγής συστάσεων βασίζονται σε τεχνικές μείωσης των διαστάσεων, όπως η μέθοδος Singular Value Decomposition (SVD), η οποία έχει την ικανότητα να παράγει αξιόπιστες και αποτελεσματικές συστάσεις.

Συνωνυμία (Synonymy). Η συνωνυμία είναι η τάση των πολύ παρόμοιων αντικειμένων να έχουν διαφορετικά ονόματα ή καταχωρήσεις. Τα περισσότερα συστήματα συστάσεων δυσκολεύονται να κάνουν διάκριση μεταξύ των στενά συνδεδεμένων αντικειμένων, όπως η διαφορά μεταξύ π.χ. του baby wear και του baby cloth. Τα συστήματα συνεργατικού φιλτραρίσματος συνήθως δεν βρίσκουν καμία αντιστοιχία μεταξύ των δύο όρων για να μπορέσουν να υπολογίσουν την ομοιότητά τους. Διαφορετικές μέθοδοι, όπως η αυτόματη επέκταση όρων, η κατασκευή θησαυρού και η Singular Value Decomposition (SVD), ιδίως η Latent Semantic Indexing, είναι ικανές να επιλύσουν το πρόβλημα της συνωνυμίας. Το μειονέκτημα αυτών των μεθόδων είναι ότι ορισμένοι προστιθέμενοι όροι μπορεί να έχουν διαφορετική σημασία από την προβλεπόμενη, γεγονός που οδηγεί μερικές φορές σε ταχεία υποβάθμιση της απόδοσης των συστάσεων.

1.2.2 Μέθοδοι που χρησιμοποιούνται στο μοντέλο του συνεργατικού φιλτραρίσματος.

Τεχνικές βασισμένες στη μνήμη (Memory based techniques):

Τα αντικείμενα που έχουν ήδη αξιολογηθεί από τον χρήστη στο παρελθόν παίζουν σημαντικό ρόλο στην αναζήτηση ενός γείτονα που μοιράζεται την ίδια εκτίμηση με αυτόν. Μόλις βρεθεί ένας γείτονας ενός χρήστη, μπορούν να χρησιμοποιηθούν διάφοροι αλγόριθμοι για να συνδυάσουν τις προτιμήσεις των γειτόνων για τη δημιουργία συστάσεων. Λόγω της αποτελεσματικότητας αυτών των τεχνικών, έχουν ευρεία επιτυχία σε εφαρμογές της πραγματικής ζωής. Η τεχνική συνεργατικού φιλτραρίσματος με βάση τη μνήμη μπορεί να επιτευχθεί με δύο τρόπους, μέσω τεχνικών με βάση τον χρήστη ή με βάση τα στοιχεία. Η τεχνική συνεργατικού φιλτραρίσματος με βάση τον χρήστη υπολογίζει την ομοιότητα μεταξύ των χρηστών συγκρίνοντας τις αξιολογήσεις τους για το ίδιο στοιχείο και στη συνέχεια υπολογίζει την προβλεπόμενη αξιολόγηση για ένα στοιχείο από τον ενεργό χρήστη ως σταθμισμένο μέσο όρο των αξιολογήσεων του στοιχείου από χρήστες παρόμοιους με τον ενεργό χρήστη, όπου τα βάρη είναι οι ομοιότητες αυτών των χρηστών με το αντικείμενο-στόχο. Οι τεχνικές φιλτραρίσματος με βάση το αντικείμενο υπολογίζουν τις προβλέψεις χρησιμοποιώντας την ομοιότητα μεταξύ των αντικειμένων και όχι την ομοιότητα μεταξύ των χρηστών. Δημιουργεί ένα μοντέλο ομοιότητας αντικειμένων ανακτώντας όλα τα αντικείμενα που έχουν βαθμολογηθεί από έναν ενεργό χρήστη από τον πίνακα χρήστη-αντικειμένου, προσδιορίζει πόσο παρόμοια είναι τα ανακτηθέντα αντικείμενα με το αντικείμενο-στόχο, στη συνέχεια επιλέγει τα k πιο παρόμοια αντικείμενα και προσδιορίζονται επίσης οι αντίστοιχες ομοιότητές τους. Η πρόβλεψη γίνεται με τη λήψη ενός σταθμισμένου μέσου όρου των αξιολογήσεων των ενεργών χρηστών για τα παρόμοια αντικείμενα k. Χρησιμοποιούνται διάφοροι τύποι μέτρων ομοιότητας για τον υπολογισμό της ομοιότητας μεταξύ αντικειμένου/χρήστη. Τα δύο πιο δημοφιλή μέτρα ομοιότητας είναι τα μέτρα που βασίζονται στη συσχέτιση και τα μέτρα που βασίζονται στο συνημίτονο. Ο συντελεστής συσχέτισης Pearson χρησιμοποιείται για τη μέτρηση του βαθμού στον οποίο δύο μεταβλητές σχετίζονται γραμμικά μεταξύ τους και ορίζεται ως εξής.

$$s(a,u) = \frac{\sum_{i=1}^{n}(r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i=1}^{n}(r_{a,i} - \overline{r_a})^2}\sqrt{\sum_{i=1}^{n}(r_{u,i} - \overline{r_u})^2}}$$

Από την παραπάνω εξίσωση, s (a , u) δηλώνει την ομοιότητα μεταξύ δύο χρηστών a και u, $r\ a$, i είναι η βαθμολογία που δίνει ο χρήστης a στο αντικείμενο i, $r\ a\ \bar{}$ είναι η μέση βαθμολογία που δίνει ο χρήστης a ενώ n είναι ο συνολικός αριθμός των αντικειμένων στο χώρο χρήστη-αντικειμένου. Επίσης, η πρόβλεψη για ένα στοιχείο γίνεται από τον σταθμισμένο συνδυασμό των αξιολογήσεων των επιλεγμένων

γειτόνων, ο οποίος υπολογίζεται ως η σταθμισμένη απόκλιση από τον μέσο όρο των γειτόνων. Ο γενικός τύπος πρόβλεψης είναι

$$p(a,i) = \overline{r_a} + \frac{\sum_{i=1}^{n}\left(r_{u,i} - \overline{r_u}\right) \times s(a,u)}{\sum_{i=1}^{n} s(a,u)}$$

Η ομοιότητα συνημίτονου διαφέρει από το μέτρο με βάση τον Pearson στο ότι είναι ένα μοντέλο διανυσματικού χώρου που βασίζεται στη γραμμική άλγεβρα και όχι στη στατιστική προσέγγιση. Μετρά την ομοιότητα μεταξύ δύο διανυσμάτων n-διάστασης με βάση τη μεταξύ τους γωνία. Το μέτρο με βάση το συνημίτονο χρησιμοποιείται ευρέως στους τομείς της ανάκτησης πληροφοριών και της εξόρυξης κειμένων για τη σύγκριση δύο εγγράφων κειμένου, στην περίπτωση αυτή, τα έγγραφα αναπαρίστανται ως διανύσματα όρων. Η ομοιότητα μεταξύ δύο στοιχείων u και v μπορεί να οριστεί ως εξής :

$$s(\vec{u},\vec{v}) = \frac{\vec{u}\cdot\vec{v}}{|\vec{u}|*|\vec{v}|} = \frac{\sum_i r_{u,i} r_{v,i}}{\sqrt{\sum_i {r^2}_{u,i}} \times \sqrt{\sum_i {r^2}_{v,i}}}$$

Το μέτρο ομοιότητας αναφέρεται επίσης ως μετρική ομοιότητας και είναι μέθοδοι που χρησιμοποιούνται για τον υπολογισμό των βαθμολογιών που εκφράζουν πόσο όμοιοι είναι οι χρήστες ή τα στοιχεία μεταξύ τους. Αυτές οι βαθμολογίες μπορούν στη συνέχεια να χρησιμοποιηθούν ως βάση για τη δημιουργία συστάσεων βάσει χρηστών ή στοιχείων. Ανάλογα με το πλαίσιο χρήσης, οι μετρικές ομοιότητας μπορούν επίσης να αναφέρονται ως μετρικές συσχέτισης ή μετρικές απόστασης .

Τεχνικές βασισμένες σε μοντέλα (Model-based techniques):

Αυτή η τεχνική χρησιμοποιεί τις προηγούμενες αξιολογήσεις για την εκμάθηση ενός μοντέλου, προκειμένου να βελτιώσει την απόδοση της Τεχνικής Συνεργατικού Φιλτραρίσματος. Η διαδικασία δημιουργίας μοντέλου μπορεί να γίνει με τη χρήση τεχνικών μηχανικής μάθησης ή εξόρυξης δεδομένων. Αυτές οι τεχνικές μπορούν να συστήσουν γρήγορα ένα σύνολο στοιχείων για το γεγονός ότι χρησιμοποιούν προ-υπολογισμένο μοντέλο και έχει αποδειχθεί ότι παράγουν αποτελέσματα συστάσεων που είναι παρόμοια με τις τεχνικές συστάσεων που βασίζονται στη γειτονιά. Παραδείγματα αυτών των τεχνικών είναι η τεχνική μείωσης διαστάσεων, όπως η Singular Value Decomposition (SVD), η τεχνική συμπλήρωσης πινάκων, οι μέθοδοι Latent Semantic, η παλινδρόμηση και η ομαδοποίηση. Οι τεχνικές που βασίζονται σε μοντέλα αναλύουν τον πίνακα χρήστη-αντικειμένου για να εντοπίσουν σχέσεις μεταξύ των αντικειμένων χρησιμοποιούν αυτές τις σχέσεις για να συγκρίνουν τον

κατάλογο των κορυφαίων Ν συστάσεων. Οι τεχνικές που βασίζονται σε μοντέλα επιλύουν τα προβλήματα σπανιότητας που σχετίζονται με τα συστήματα συστάσεων. Η χρήση αλγορίθμων μάθησης έχει επίσης αλλάξει τον τρόπο των συστάσεων από τη σύσταση του τι να καταναλώσουν οι χρήστες, στη σύσταση του πότε να καταναλώσουν πραγματικά ένα προϊόν.

1.3 Υβριδικός τρόπος φιλτραρίσματος (Hybrid Filtering).

Για καλύτερα αποτελέσματα, ορισμένα συστήματα συστάσεων συνδυάζουν διαφορετικές τεχνικές συνεργατικών προσεγγίσεων και προσεγγίσεων που βασίζονται στο περιεχόμενο. Με την υβριδική προσέγγιση, μπορούν να αποφευχθούν οι περιορισμοί της προσέγγισης που βασίζεται στο περιεχόμενο και της συνεργατικής προσέγγισης. Ο συνδυασμός αυτών των δύο προσεγγίσεων γίνεται με διαφορετικούς τρόπους που μπορούν να ταξινομηθούν ως εξής:

- Εφαρμογή συνεργατικών και βασισμένων στο περιεχόμενο μεθόδων ξεχωριστά και συνδυασμός των προβλέψεών τους.
- Ενσωμάτωση ορισμένων χαρακτηριστικών βασισμένων στο περιεχόμενο σε μια συνεργατική προσέγγιση.
- Ενσωμάτωση ορισμένων συνεργατικών χαρακτηριστικών σε προσέγγιση βασισμένη στο περιεχόμενο.
- Κατασκευή γενικού ενοποιημένου μοντέλου που ενσωματώνει και με τις δύο προσεγγίσεις περιεχομένου καθώς και μια συνεργατική.

1.3.1 Μέθοδοι που χρησιμοποιούνται στο Υβριδικό μοντέλο φιλτραρίσματος.

<u>Σταθμισμένος υβριδισμός (Weighted hybridization):</u>

Ο σταθμισμένος υβριδισμός συνδυάζει τα αποτελέσματα διαφορετικών συστάσεων για τη δημιουργία ενός καταλόγου συστάσεων ή μιας πρόβλεψης, ενσωματώνοντας τις βαθμολογίες από κάθε μία από τις χρησιμοποιούμενες τεχνικές με έναν γραμμικό τύπο. Ένα παράδειγμα σταθμισμένου υβριδικού συστήματος συστάσεων είναι το P-tango . Το σύστημα αποτελείται από έναν συστήνοντα που βασίζεται σε περιεχόμενο και έναν συνεργατικό συστήνοντα. Αρχικά τους δίνονται ίσα βάρη, αλλά τα βάρη

προσαρμόζονται καθώς οι προβλέψεις επιβεβαιώνονται ή όχι. Το πλεονέκτημα ενός σταθμισμένου υβριδικού συστήματος είναι ότι όλα τα πλεονεκτήματα του συστήματος συστάσεων αξιοποιούνται κατά τη διαδικασία σύστασης με απλό τρόπο.

Εναλλαγή υβριδισμού (Switching hybridization):

Το σύστημα αλλάζει σε μία από τις τεχνικές σύστασης σύμφωνα με ένα εύρημα που αντανακλά την ικανότητα του συστήνοντος να παράγει μια καλή αξιολόγηση. Ο υβριδισμός εναλλαγής έχει τη δυνατότητα να αποφεύγει προβλήματα που αφορούν ειδικά μια μέθοδο, π.χ. το πρόβλημα του νέου χρήστη της σύστασης βάσει περιεχομένου, μεταβαίνοντας σε ένα συνεργατικό σύστημα σύστασης. Το πλεονέκτημα αυτής της στρατηγικής είναι ότι το σύστημα είναι ευαίσθητο στα δυνατά και αδύνατα σημεία των συστατικών μερών των συνιστώντων που το αποτελούν. Το κύριο μειονέκτημα της εναλλαγής υβριδικών συστημάτων είναι ότι συνήθως εισάγει μεγαλύτερη πολυπλοκότητα στη διαδικασία σύστασης, επειδή πρέπει να καθοριστεί το κριτήριο εναλλαγής, το οποίο συνήθως αυξάνει τον αριθμό των παραμέτρων του συστήματος σύστασης. Παράδειγμα ενός υβριδικού συστήματος με εναλλαγή είναι το DailyLearner που χρησιμοποιεί τόσο υβριδικό σύστημα βασισμένο στο περιεχόμενο όσο και συνεργατικό, όπου χρησιμοποιείται πρώτα σύσταση βασισμένη στο περιεχόμενο πριν από τη συνεργατική σύσταση σε μια κατάσταση όπου το σύστημα βασισμένο στο περιεχόμενο δεν μπορεί να κάνει συστάσεις με αρκετά αποδεικτικά στοιχεία.

Υβριδισμός καταρράκτη (Cascade hybridization):

Η τεχνική της κλιμακωτής υβριδοποίησης εφαρμόζει μια επαναληπτική διαδικασία βελτίωσης για την κατασκευή μιας σειράς προτίμησης μεταξύ διαφορετικών στοιχείων. Οι συστάσεις μιας τεχνικής βελτιώνονται από μια άλλη τεχνική συστάσεων. Η πρώτη τεχνική συστάσεων παράγει έναν χονδροειδή κατάλογο συστάσεων, ο οποίος με τη σειρά του βελτιώνεται από την επόμενη τεχνική συστάσεων. Η τεχνική υβριδισμού είναι πολύ αποτελεσματική και ανεκτική στο θόρυβο λόγω της χονδροειδούς προς λεπτότερη φύση της επανάληψης. EntreeC είναι ένα παράδειγμα μεθόδου υβριδισμού καταρράκτη που χρησιμοποίησε επανάληψη βασισμένη στη γνώση και στη συνεργατική σύσταση.

Μικτός υβριδισμός (Mixed hybridization):

Τα μικτά υβρίδια συνδυάζουν ταυτόχρονα τα αποτελέσματα συστάσεων διαφορετικών τεχνικών αντί να έχουν μόνο μία σύσταση ανά στοιχείο. Κάθε στοιχείο έχει πολλαπλές συστάσεις που συνδέονται με αυτό από διαφορετικές τεχνικές

σύστασης. Στον μικτό υβριδισμό, οι επιμέρους επιδόσεις δεν επηρεάζουν πάντα τη γενική απόδοση μιας τοπικής περιοχής. Παράδειγμα συστήματος συστάσεων αυτής της κατηγορίας που χρησιμοποιεί τη μικτή υβριδοποίηση είναι το σύστημα PTV το οποίο συνιστά ένα πρόγραμμα παρακολούθησης τηλεόρασης για έναν χρήστη συνδυάζοντας συστάσεις από συστήματα βασισμένα στο περιεχόμενο και συνεργατικά συστήματα για να σχηματίσουν ένα πρόγραμμα. Profinder και PickAFlick αποτελούν επίσης παραδείγματα μικτών υβριδικών συστημάτων.

Συνδυασμός χαρακτηριστικών (Feature-combination):

Τα χαρακτηριστικά που παράγονται από μια συγκεκριμένη τεχνική σύστασης τροφοδοτούν μια άλλη τεχνική σύστασης. Για παράδειγμα, η βαθμολογία των παρόμοιων χρηστών, η οποία αποτελεί χαρακτηριστικό γνώρισμα του συνεργατικού φιλτραρίσματος, χρησιμοποιείται σε μια τεχνική συστάσεων που βασίζεται στην επιχειρηματολογία περίπτωσης ως ένα από τα χαρακτηριστικά γνωρίσματα για τον προσδιορισμό της ομοιότητας μεταξύ των αντικειμένων. Το Pipper είναι ένα παράδειγμα τεχνικής συνδυασμού χαρακτηριστικών που χρησιμοποιεί τις αξιολογήσεις του συνεργατικού φίλτρου σε ένα σύστημα βασισμένο στο περιεχόμενο ως χαρακτηριστικό για τη σύσταση ταινιών. Το πλεονέκτημα αυτής της τεχνικής είναι ότι, δεν βασίζεται πάντα αποκλειστικά στα συνεργατικά δεδομένα.

Βελτίωση χαρακτηριστικών (Feature-augmentation):

Η τεχνική χρησιμοποιεί τις αξιολογήσεις και άλλες πληροφορίες που παράγονται από την προηγούμενη σύσταση και απαιτεί επίσης πρόσθετη λειτουργικότητα από τα συστήματα συστάσεων. Για παράδειγμα, το σύστημα Libra κάνει συστάσεις βιβλίων βάσει περιεχομένου σε δεδομένα που βρίσκονται στο Amazon.com χρησιμοποιώντας έναν ταξινομητή κειμένου naïve Bayes. Τα υβριδικά συστήματα ενίσχυσης χαρακτηριστικών υπερτερούν έναντι των μεθόδων συνδυασμού χαρακτηριστικών, καθώς προσθέτουν έναν μικρό αριθμό χαρακτηριστικών στην κύρια σύσταση.

Meta-level:

Το εσωτερικό μοντέλο που παράγεται από μια τεχνική σύστασης χρησιμοποιείται ως είσοδος για μια άλλη. Το μοντέλο που δημιουργείται είναι πάντα πιο πλούσιο σε πληροφορίες σε σύγκριση με μια απλή αξιολόγηση. Μετα-επίπεδο τα υβρίδια είναι σε θέση να επιλύσουν το πρόβλημα σπανιότητας των τεχνικών συνεργατικού φιλτραρίσματος χρησιμοποιώντας ολόκληρο το μοντέλο που μαθαίνεται από την πρώτη τεχνική ως είσοδο για τη δεύτερη τεχνική. Παράδειγμα τεχνικής μετα-επιπέδου είναι το LaboUr η οποία χρησιμοποιεί άμεση μάθηση για τη δημιουργία

προφίλ χρήστη με βάση το περιεχόμενο, το οποίο στη συνέχεια συγκρίνεται με συνεργατικό τρόπο.

Κεφάλαιο 2 : Αλγόριθμοι Ομαδοποίησης

Η ομαδοποίηση είναι μια προσέγγιση μάθησης χωρίς επίβλεψη που λαμβάνει ένα σύνολο δεδομένων ως είσοδο, ενώ τα αναμενόμενα αποτελέσματα είναι άγνωστα. Ο στόχος είναι να μάθουμε τη δομή των δεδομένων και τις σχέσεις μεταξύ των συστατικών τους. Με άλλα λόγια, η ομαδοποίηση έχει ως στόχο να ανακαλύψει τα μοτίβα και τις ομάδες στα δεδομένα χωρίς να προσδιορίζει αναμενόμενες πληροφορίες ή τιμές. Το αποτέλεσμα της διαδικασίας ομαδοποίησης θα είναι διάφορες ομάδες που περιγράφουν τη δομή των δεδομένων, όπως οι ομάδες χρηστών με παρόμοια αγοραστική συμπεριφορά και οι ομάδες χρηστών με παρόμοια αναγνωστικά ενδιαφέροντα. Τα συστήματα συστάσεων, συγκεκριμένα τα συστήματα συνεργατικού φιλτραρίσματος, χρησιμοποιούν αυτές τις συστάδες για να παρέχουν στους χρήστες ακριβείς και γρήγορες συστάσεις από μια ποικιλία στοιχείων. Υπάρχουν πολλοί αλγόριθμοι κατασκευής συστάδων που προτείνονται στη βιβλιογραφία και οι οποίοι κατατάσσονται σε διάφορες κατηγορίες ανάλογα με διάφορους παράγοντες, όπως το πρόβλημα που πρέπει να επιλυθεί, η δομή του συνόλου δεδομένων, η μέθοδος που χρησιμοποιείται και η αναμενόμενη μορφή της εξόδου. Ορισμένες από τις γνωστές κατηγορίες αλγορίθμων είναι η κατασκευή συστάδων με βάση το κεντροειδές, η ιεραρχική και η κατασκευή συστάδων με βάση την πυκνότητα.

Πρώτον, οι μέθοδοι συσταδοποίησης με βάση το κεντροειδές Αλγόριθμοι ομαδοποίησης κατάτμησης δημιουργούν συστάδες από τα αντικείμενα των δεδομένων υπολογίζοντας την απόσταση μεταξύ τους με βάση άλλα αντικείμενα, που ονομάζονται κεντροειδή, τα οποία δεν ανήκουν απαραίτητα στο σύνολο δεδομένων. Οι αλγόριθμοι K-Means, K-Medoids και Fuzzy C-Means είναι κοινοί αλγόριθμοι βασισμένοι σε κεντροειδή.

Η ιεραρχική κατασκευή συστάδων συνίσταται στη σταδιακή δημιουργία συστάδων είτε είναι από κάτω προς τα πάνω (συσσωρευτική) είτε από πάνω προς τα κάτω (διαχωριστική). Ο αλγόριθμος συσσωρευτικής συσταδοποίησης δημιουργεί τόσες συστάδες όσες είναι ο αριθμός των στοιχείων δεδομένων, όπου κάθε συστάδα παίρνει αρχικά ένα στοιχείο, στη συνέχεια να υπολογίζει τις αποστάσεις μεταξύ των συστάδων που δημιουργούνται και συγχωνεύει τις πλησιέστερες συστάδες στην απόσταση μέχρις ότου όλες να παραμείνουν αμετάβλητες ή να ικανοποιηθεί μια

συγκεκριμένη συνθήκη. Η διαιρετική προσέγγιση λειτουργεί με τον αντίθετο τρόπο, ξεκινώντας από μία μόνο συστάδα που περιέχει όλα τα στοιχεία δεδομένων. Αυτή η συστάδα συνεχίζει να διασπάται με βάση την απόσταση μεταξύ των στοιχείων μέχρι να μην αλλάξει καμία συστάδα ή να ικανοποιηθεί μια συγκεκριμένη συνθήκη διακοπής.

Τέλος, η συσταδοποίηση με βάση την πυκνότητα χρησιμοποιείται λιγότερο στα συστήματα συστάσεων σε σχέση με τις προηγούμενες κατηγορίες αλγορίθμων συσταδοποίησης. Ο DBSCAN (Density-Based Spatial Clustering of Applications with Noise) είναι ο πιο συνηθισμένος αλγόριθμος αυτής της κατηγορίας. Συνοψίζοντας, πολλές προσεγγίσεις ομαδοποίησης μπορούν να χρησιμοποιηθούν σε συστήματα συστάσεων και κάθε προσέγγιση έχει τις προδιαγραφές της και επιλύει ένα συγκεκριμένο πρόβλημα. [8]

2.1 Αλγόριθμοι ομαδοποίησης κατάτμησης (Partitional Clustering).

Οι αλγόριθμοι ομαδοποίησης με κατάτμηση είναι μια μη ιεραρχική ομαδοποίηση που συνήθως ασχολείται με στατικά σύνολα. Ο στόχος της είναι να ανακαλύψει τις ομαδοποιήσεις που υπάρχουν στα δεδομένα μέσω τεχνικών βελτιστοποίησης της αντικειμενικής συνάρτησης οι οποίες βελτιώνουν επαναληπτικά την ποιότητα των διαμερισμάτων. Σε αυτές τις μεθόδους, ο επιθυμητός αριθμός συστάδων, k, θα παρέχεται από τον χρήστη, οι οποίες στη συνέχεια βελτιώνονται επαναληπτικά. Η ομαδοποίηση με κατάτμηση παραμένει μια από τις πιο δημοφιλείς και εφαρμοσμένες τεχνικές λόγω της απλότητας, της αποδοτικότητας, της πολύ εύκολης εφαρμογής της και της εμπειρικής της επιτυχίας. Δεν επιβάλλει ιεραρχική δομή και υπολογίζει όλες τις εφικτές συστάδες ταυτόχρονα.[9]

K-Means

Η τεχνική K-Means λαμβάνει ως είσοδο τον αρχικό αριθμό των κεντροειδών των συστάδων και το σύνολο δεδομένων. Η πρώτη φάση είναι η εκτίμηση του αριθμού k των κεντροειδών. Μπορεί είτε να δημιουργηθεί τυχαία είτε να επιλεγεί από το σύνολο δεδομένων. Δεύτερον, αντιστοιχίζεται κάθε ένα από τα στοιχεία στο πλησιέστερο κεντροειδές. Αυτό γίνεται με τον υπολογισμό της απόστασης μεταξύ κάθε στοιχείου και των κεντροειδών χρησιμοποιώντας ένα μέτρο απόστασης όπως η ευκλείδεια απόσταση. Η τρίτη και τελευταία φάση είναι ο υπολογισμός του μέσου όρου όλων των στοιχείων που ανατίθενται σε κάθε κεντροειδές και η ενημέρωση της προηγούμενης τιμής του. Τα δύο τελευταία στάδια επαναλαμβάνονται μέχρι να ικανοποιηθεί μια συνθήκη τερματισμού. Η συνθήκη τερματισμού μπορεί να είναι, για παράδειγμα, αν το περιεχόμενο μιας συστάδας παραμένει αμετάβλητο και ο βρόχος

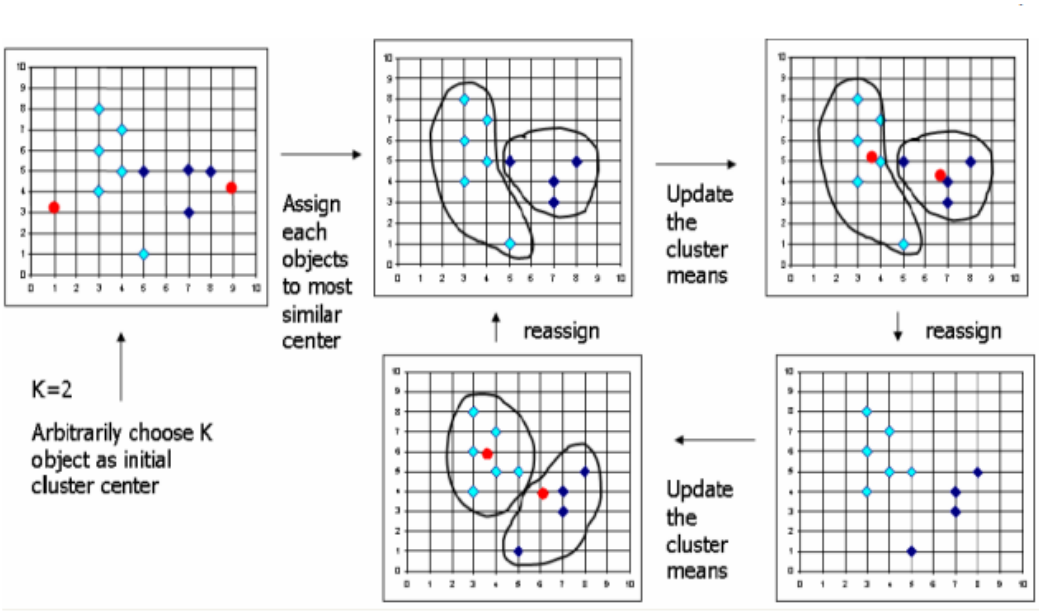
φτάσει σε ένα ελάχιστο άθροισμα αποστάσεων ή σε έναν μέγιστο αριθμό επαναλήψεων που καθορίστηκε αρχικά [10]. Παρά την ευρεία δημοτικότητά του, ο k-means είναι πολύ ευαίσθητος στο « θόρυβο » και τις ακραίες τιμές, δεδομένου ότι ένας μικρός αριθμός τέτοιων δεδομένων μπορεί να επηρεάσει σημαντικά τα κεντροειδή. Άλλες αδυναμίες είναι η ευαισθησία σε αρχικοποίηση, ο εγκλωβισμός σε τοπικά βέλτιστα, οι φτωχές περιγραφές συστάδων, η αδυναμία αντιμετώπισης συστάδων αυθαίρετου σχήματος, μεγέθους πυκνότητας και εξάρτηση από τον χρήστη για τον καθορισμό του αριθμού των συστάδων. Τέλος, αυτός ο αλγόριθμος στοχεύει στην ελαχιστοποίηση μιας αντικειμενικής συνάρτησης- στην προκειμένη περίπτωση μιας συνάρτησης τετραγωνικού σφάλματος. Η αντικειμενική συνάρτηση όπου || xi(j) - cj ||2 είναι ένα επιλεγμένο μέτρο απόστασης μεταξύ ενός σημείου δεδομένων xi(j)και του κέντρου συστάδας Cj, είναι ένας δείκτης της απόστασης των n σημείων δεδομένων από τα αντίστοιχα κέντρα των συστάδων τους.[11]

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Επιπλέον, δεν υπάρχει αποτελεσματικός τρόπος προσδιορισμού του αρχικού k αριθμού συστάδων που πρέπει να δημιουργηθούν. Επίσης, η εργασία στο πρότεινε την προσέγγιση LeaderRank εμπνευσμένη από τη σχέση ηγέτη-οπαδού στις ομάδες των κοινωνικών δικτύων για την επίλυση της ευαισθησίας του k-means της αρχικής επιλογής των κεντροειδών και τη βελτίωση της ποιότητας της συσταδοποίησης. Ο Yang, 2018 [30] πρότεινε μια βελτίωση για την ομαδοποίηση με βάση τον k-means για την επίλυση των προβλημάτων χαμηλής ακρίβειας και επεκτασιμότητας των συστάσεων. Η κύρια ιδέα τους ήταν να κανονικοποιήσουν πρώτα τα δεδομένα εισόδου, να επιλέξουν πέντε αρχικά κεντροειδή με την ίδια επέκταση, στη συνέχεια να τροφοδοτήσουν αυτά τα δεδομένα στον αλγόριθμο και τέλος να τον εκτελέσουν μέχρι τα κέντρα να γίνουν αμετάβλητα.[10]

Τα βήματα του αλγορίθμου είναι:

- Επιλογή του αριθμού των συστάδων, k.
- Δημιουργία k συστάδων με τυχαίο τρόπο και προσδιορισμός των κέντρων των συστάδων ή απευθείας δημιουργία k τυχαίων σημείων ως κέντρα συστάδων.
- Ανάθεση κάθε σημείου στο πλησιέστερο κέντρο συστάδας.
- Υπολογίστε εκ νέου τα νέα κέντρα συστάδων.
- Επαναλάβετε τα δύο προηγούμενα βήματα μέχρι να ικανοποιηθεί κάποιο κριτήριο σύγκλισης (συνήθως ότι η ανάθεση δεν έχει αλλάξει)



Εικονα 13. Τρόπος λειτουργίας του K-means

Πλεονεκτήματα του K-Mean:

- Εάν οι μεταβλητές είναι μεγάλες, τότε ο K-Means τις περισσότερες φορές

υπολογιστικά γρηγορότερα από τις ιεραρχικές μεθόδους ομαδοποίησης.

- Η μέθοδος K-Means παράγει στενότερες συστάδες από την ιεραρχική

μέθοδο συσταδοποίησης.

Μειονεκτήματα του αλγορίθμου κατάτμησης K-Means:

- Είναι δύσκολο να προβλεφθεί η τιμή K.
- Μεγαλύτερη δυσκολία στη σύγκριση της ποιότητας των συστάδων.
- Ο αλγόριθμος K-Means δεν λειτουργεί καλά με παγκόσμια

συστάδες.[6f]

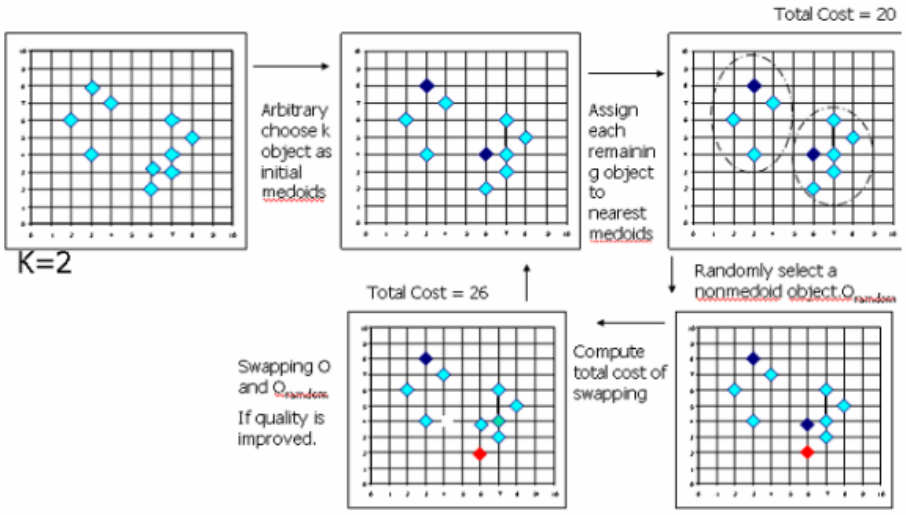
K-MEDOID

Η μέθοδος k-means βασίζεται στις τεχνικές κεντροειδούς για να αναπαριστά τη συστάδα και είναι ευαίσθητη στις ακραίες τιμές. Αυτό σημαίνει ότι, ένα αντικείμενο δεδομένων με εξαιρετικά μεγάλη τιμή μπορεί να διαταράξει την κατανομή των δεδομένων. Για να ξεπεράσουμε το πρόβλημα χρησιμοποιήσαμε τη μέθοδο k-

medoids η οποία βασίζεται σε τεχνικές αντιπροσωπευτικών αντικειμένων. Το μεσοειδές (medoid) είναι το κεντροειδές για την αναπαράσταση της συστάδας. Το μεσοειδές (medoid) είναι το το πιο κεντρικά τοποθετημένο αντικείμενο δεδομένων σε μια συστάδα. Εδώ, k αντικείμενα δεδομένων επιλέγονται τυχαία ως medoids για να αντιπροσωπεύουν k συστάδες και τα υπόλοιπα αντικείμενα δεδομένων τοποθετούνται σε σε μια συστάδα που έχει το πλησιέστερο (ή το πιο παρόμοιο) medoid με τα εν λόγω δεδομένα. αντικείμενο. Μετά την επεξεργασία όλων των αντικειμένων δεδομένων, το νέο medoid είναι που μπορεί να αντιπροσωπεύσει τη συστάδα με καλύτερο τρόπο και το όλη η διαδικασία επαναλαμβάνεται. Και πάλι όλα τα αντικείμενα δεδομένων συνδέονται με τις συστάδες με βάση τα νέα medoids. Σε κάθε επανάληψη, τα medoids αλλάζουν τη θέση τους βήμα προς βήμα. Αυτή η διαδικασία είναι συνεχίζεται έως ότου κανένα medoid δεν μετακινηθεί. Ως αποτέλεσμα, k συστάδες που αντιπροσωπεύουν ένα σύνολο n αντικειμένων δεδομένων[12].

Τα βήματα του αλγορίθμου είναι:

- Ο αλγόριθμος ξεκινά με την αυθαίρετη επιλογή των K αντικειμένων ως σημεία medoid από n σημεία δεδομένων (n>K).
- Μετά την επιλογή των K σημείων medoid, συσχετίζει κάθε αντικείμενο δεδομένων στο δεδομένο σύνολο δεδομένων με το πιο παρόμοιο medoid.
- Επιλέξτε τυχαία το μη μεδοειδές αντικείμενο O.
- Υπολογισμός του συνολικού κόστους S της ανταλλαγής αρχικού medoid με το αντικείμενο O.
- Εάν S>0, ανταλλάξτε το αρχικό medoid με το νέο.
- Επανάληψη μέχρι να μην υπάρξει καμία αλλαγή.[13]



Εικόνα 14. λειτουργεία αλγορίθμου k medoid

Πλεονεκτήματα του K-Medoid:

- Είναι απλό στην κατανόηση και εύκολο στην εφαρμογή.
- Ο αλγόριθμος K-Medoid είναι γρήγορος και συγκλίνει σε σταθερό αριθμό βημάτων.

Μειονεκτήματα του K-Medoid:

- Ο K-Mediods είναι πιο δαπανηρός από τη μέθοδο K-Means επειδή της χρονικής πολυπλοκότητάς της.
- Δεν κλιμακώνεται καλά για μεγάλα σύνολα δεδομένων.
- Τα αποτελέσματα και ο συνολικός χρόνος εκτέλεσης εξαρτώνται από τις αρχικές κατατμήσεις

Fuzzy C-means

Ο Bezdek εισήγαγε τη μέθοδο ομαδοποίησης Fuzzy C-Means σε 1981, που επεκτείνεται από τη μέθοδο συστάδων Hard C-Mean. Η FCM είναι μια αλγόριθμος ομαδοποίησης χωρίς επίβλεψη που εφαρμόζεται σε ευρύ φάσμα προβλημάτων που συνδέονται με την ανάλυση χαρακτηριστικών, την ομαδοποίηση και την σχεδιασμό ταξινομητών. Η FCM εφαρμόζεται ευρέως στη γεωργία μηχανική, την αστρονομία, τη χημεία, τη γεωλογία, την ανάλυση εικόνων, την ιατρική διάγνωση, την ανάλυση σχήματος και την αναγνώριση στόχων. Με την ανάπτυξη της ασαφούς θεωρίας, η FCM αλγόριθμος ομαδοποίησης ο οποίος στην πραγματικότητα βασίζεται στην ασαφή Ruspini θεωρία συσταδοποίησης προτάθηκε τη δεκαετία του 1980. Αυτός ο αλγόριθμος είναι χρησιμοποιείται για ανάλυση με βάση την απόσταση μεταξύ διαφόρων δεδομένων εισόδου σημείων. Οι συστάδες σχηματίζονται σύμφωνα με την απόσταση μεταξύ των σημείων δεδομένων και τα κέντρα των συστάδων σχηματίζονται για κάθε συστάδα. Στην πραγματικότητα, η FCM είναι μια τεχνική ομαδοποίησης δεδομένων στην οποία ένα σύνολο δεδομένων ομαδοποιείται σε n συστάδες με κάθε σημείο δεδομένων στην σύνολο δεδομένων σχετίζεται με κάθε συστάδα και θα έχει υψηλό βαθμό σύνδεσης σε αυτή τη συστάδα και σε ένα άλλο σημείο δεδομένων που βρίσκεται μακριά από το κέντρο μιας συστάδας το οποίο θα έχει χαμηλό βαθμό συμμετοχής σε αυτή τη συστάδα [14].

Το Fuzzy C-means είναι μια επικαλυπτόμενη τεχνική ομαδοποίησης κατά τμήματα. Είναι παρόμοια με την k-means κατά τον τρόπο που ο αρχικός αριθμός των συστάδων καθορίζεται στην αρχή. Ωστόσο, θεωρείται μη αποκλειστική επειδή τα στοιχεία μπορούν να ανήκουν σε περισσότερες από μία συστάδες, σε αντίθεση με τον k-means όπου κάθε στοιχείο εμφανίζεται μόνο μία φορά. Στο FCM, σε κάθε στοιχείο αποδίδεται μια τιμή βάρους που καθορίζει το βαθμό συμμετοχής στη συστάδα και

κυμαίνεται μεταξύ μηδέν και ένα, έτσι ώστε το άθροισμα των βαθμών συμμετοχής ενός στοιχείου να ισούται με ένα. Η εργασία στο πρότεινε ένα μοντέλο για την προσέγγιση ταξινόμησης των λιανοπωλητών με βάση τη λήψη αποφάσεων με πολλαπλά κριτήρια και την ασαφή ομαδοποίηση. Αποτελούνταν από δύο φάσεις: τον προσδιορισμό της κατάστασης των λιανοπωλητών σε συγκεκριμένα εμπορικά κέντρα με τη χρήση MCDM και την ομαδοποίηση των καταστημάτων μέσω των τιμών των προϊόντων με τη χρήση FCM. Τα ασαφή C-μέσα είναι καλύτερα από τα k-μέσα στην ανίχνευση και διόρθωση θορύβων και λανθασμένων ταξινομήσεων, αλλά χρειάζονται περισσότερο χρόνο υπολογισμού όταν πρόκειται για μεγάλα σύνολα δεδομένων. Αυτή η τεχνική χρησιμοποιείται συχνά σε συστήματα συστάσεων ηλεκτρονικού εμπορίου [10].

Τα βήματα του αλγορίθμου είναι

Έστω X = {x1, x2, x3 ..., x_n} το σύνολο των σημείων δεδομένων και V = {v1, v2, v3 ..., vc} το σύνολο των κέντρων.

- Επιλέξτε τυχαία τα κέντρα των συστάδων 'c'.
- Υπολογίστε την ασαφή συμμετοχή «μ_{ij}» χρησιμοποιώντας:
  $$\mu_{ij} = 1 \,/\, \sum_{k=1}^{c} \left(d_{ij}/d_{ik}\right)^{(2/m-1)}$$
- Υπολογίστε τα ασαφή κέντρα 'v_j' χρησιμοποιώντας:
  $$v_j = \left(\sum_{i=1}^{n} \left(\mu_{ij}\right)^m x_i\right) / \left(\sum_{i=1}^{n} \left(\mu_{ij}\right)^m\right), \forall_j = 1,2, \ldots . c$$
- Επαναλάβετε τα βήματα 2) και 3) έως ότου επιτευχθεί η ελάχιστη τιμή «J» ή ||U(k+1) - U(k)|| < β.
  όπου,
  'k' είναι το βήμα επανάληψης ,
  'β' είναι το κριτήριο τερματισμού μεταξύ [0, 1],
  'U = (μ_{ij})n*c' είναι ο ασαφής πίνακας συμμετοχής,
  'J' είναι η αντικειμενική συνάρτηση.

Πλεονεκτήματα

- Δίνει το καλύτερο αποτέλεσμα για επικαλυπτόμενα σύνολα δεδομένων και συγκριτικά καλύτερο από τον αλγόριθμο k-means.
- Σε αντίθεση με τον k-means όπου το σημείο δεδομένων πρέπει να ανήκει αποκλειστικά σε ένα κέντρο συστάδας εδώ το σημείο δεδομένων λαμβάνει συμμετοχή σε κάθε κέντρο συστάδας, με αποτέλεσμα το σημείο δεδομένων να μπορεί να ανήκει σε περισσότερα από ένα κέντρα συστάδας.

Μειονεκτήματα

- Προδιαγραφή του αριθμού των συστάδων με βάση το Apriori.
- Με χαμηλότερη τιμή του β έχουμε το καλύτερο αποτέλεσμα αλλά εις βάρος περισσότερων επαναλήψεων.
- Τα μέτρα ευκλείδειας απόστασης μπορούν να σταθμίσουν άνισα τους υποκείμενους παράγοντες.

Κατάτμηση γύρω από το Medoid PAM (Partitioning Around Medoid):

Ο αλγόριθμος ομαδοποίησης κατά τμήματα που σχετίζεται με τον αλγόριθμο k-means και τον αλγόριθμο μετατόπισης medoid αναφέρεται ως Partitioning Around Medoid (PAM). Ο αλγόριθμος PAM ομαδοποιεί τα αντικείμενα σε μια δεδομένη m μεταβλητή με κλίμακα διαστήματος και μπορεί επίσης να εφαρμοστεί όταν ο πίνακας δεδομένων εισόδου είναι ένας πίνακας ανομοιογένειας. Τόσο ο αλγόριθμος PAM όσο και ο αλγόριθμος k-mean προσπαθούν να ελαχιστοποιήσουν την απόσταση μεταξύ των σημείων που επισημαίνονται σε μια συγκεκριμένη συστάδα και ενός σημείου που επιλέγεται ως το κέντρο της συγκεκριμένης συστάδας. Όμως ο PAM είναι ισχυρότερος από τον k-means επειδή ελαχιστοποιείται το άθροισμα των ανομοιοτήτων σε αντίθεση με το άθροισμα των τετραγωνικών ευκλείδειων αποστάσεων στην περίπτωση του k-mean. Είναι ευάλωτος στο ζήτημα της αρχικής εισόδου, και επίσης η αδυναμία υπολογισμού μεγάλων συνόλων δεδομένων, ιδιαίτερα συνδεδεμένων συστάδων και συνόλων δεδομένων υψηλής διάστασης την καθιστά λιγότερο απαιτητή για την ομαδοποίηση μιας ομάδας δεδομένων, όπως τα δεδομένα γονιδιακής έκφρασης. Οι τύποι δεδομένων που χρησιμοποιούν αυτές τις μεθόδους ομαδοποίησης είναι τα κατηγορικά δεδομένα, τα διακριτά δεδομένα, τα δεδομένα κειμένου, τα δεδομένα πολυμέσων, τα αβέβαια δεδομένα.[9]

Ο αλγόριθμος έχει δύο φάσεις:

- Στην πρώτη φάση, BUILD, επιλέγεται μια συλλογή k αντικειμένων για ένα αρχικό σύνολο S.
- Στη δεύτερη φάση, SWAP, προσπαθεί κανείς να βελτιώσει την ποιότητα της ομαδοποίησης ανταλλάσσοντας επιλεγμένα αντικείμενα με μη επιλεγμένα αντικείμενα.

Η φάση BUILD περιλαμβάνει τα ακόλουθα βήματα:

- Αρχικοποίηση του S προσθέτοντας σε αυτό ένα αντικείμενο για το οποίο το άθροισμα των αποστάσεων από όλα τα άλλα αντικείμενα είναι ελάχιστο.
- Θεωρήστε ένα αντικείμενο $i \in U$ ως υποψήφιο για συμπερίληψη στο σύνολο των επιλεγμένων αντικειμένων.
- Για ένα αντικείμενο $j \in U - \{i\}$ υπολογίστε το D_j, την ανομοιότητα μεταξύ του j και του πλησιέστερου αντικειμένου στο S.

- Εάν D_j > d(i, j) το αντικείμενο j θα συμβάλει στην απόφαση επιλογής του αντικειμένου i (επειδή μπορεί να ωφεληθεί η ποιότητα της ομαδοποίησης)- έστω C_{ji} = max{D_j - d(j, i), 0}.
- Υπολογίστε το συνολικό κέρδος που προκύπτει από την προσθήκη του i στο S ως g_i = P j∈U Cji.
- Επιλέξτε εκείνο το αντικείμενο i που μεγιστοποιεί το g_i- έστω S := S ∪ {i} και U = U - {i}.

Τα βήματα αυτά εκτελούνται μέχρι να επιλεγούν k αντικείμενα.

Η δεύτερη φάση, SWAP, επιχειρεί να βελτιώσει το σύνολο των επιλεγμένων αντικειμένων και, επομένως, να βελτιώσει την ποιότητα της ομαδοποίησης. Αυτό γίνεται εξετάζοντας όλα τα ζεύγη (i, h) ∈ S ×U και συνίσταται στον υπολογισμό της επίδρασης T_{ih} στο άθροισμα των ανομοιοτήτων μεταξύ των αντικειμένων και του πλησιέστερου επιλεγμένου αντικειμένου που προκαλείται από την ανταλλαγή των i και h, δηλαδή από τη μεταφορά του i από το S στο U και από τη μεταφορά του h από το U στο S.

Ο υπολογισμός του T_{ih} περιλαμβάνει τον υπολογισμό της συνεισφοράς K_{jih} κάθε αντικειμένου j ∈ U - {h} στην ανταλλαγή των i και h. Σημειώστε ότι έχουμε είτε d(j, i) > D_j είτε d(j, i) = D_j .

Η φάση SWAP περιλαμβάνει τα ακόλουθα βήματα:

- Το K_{jih} υπολογίζεται λαμβάνοντας υπόψη τις ακόλουθες περιπτώσεις
  α) εάν d(j, i) > D_j , τότε εμφανίζονται δύο υποπεριπτώσεις:
  i. εάν d(j, h) ≥ D_j , τότε K_{jih} = 0
  ii. εάν d(j, h) < D_j , τότε K_{jih} = d(j, h) - D_j .

  Και στις δύο υποπεριπτώσεις, K_{jih} = min{d(j, h) - D_j , 0}.

  β) αν d(j, i) = D_j , έχουμε δύο υποπεριπτώσεις:
  i. αν d(j, h) < E_j , όπου E_j είναι η ανομοιότητα μεταξύ του j και του δεύτερου πλησιέστερου επιλεγμένου αντικειμένου, τότε K_{jih} = d(j, h) - D_j ; σημειώστε ότι το K_{jih} μπορεί να είναι είτε θετικό είτε αρνητικό.
  ii. αν d(j, h) ≥ E_j , τότε K_{jih} = E_j - D_j ; σε αυτή την περίπτωση K_{jih} > 0.

Σε κάθε μία από τις παραπάνω υποπεριπτώσεις έχουμε K_{jih} = min{d(j, h), E_j} - D_j .

- Υπολογίστε το συνολικό αποτέλεσμα της ανταλλαγής ως εξής
  $$T_{ih} = \sum\{K_{jih} \mid j \in U\}$$

- Επιλέξτε ένα ζεύγος (i, h) ∈ S × U που ελαχιστοποιεί το T_{ih}.
- Εάν T_{ih} < 0, η ανταλλαγή πραγματοποιείται, τα Dp και Ep ενημερώνονται για κάθε αντικείμενο p και επιστρέφουμε στο βήμα 1. Εάν min T_{ih} > 0, η τιμή του

αντικειμενικού δεν μπορεί να μειωθεί και ο αλγόριθμος σταματά. Φυσικά, αυτό συμβαίνει όταν όλες οι τιμές του T_{ih} είναι θετικές και αυτή ακριβώς είναι η συνθήκη ακινητοποίησης του αλγορίθμου.

CLARA(Clustering for Large Application):

CLARA σημαίνει ομαδοποίηση μεγάλων εφαρμογών και αναπτύχθηκε από τους Kaufman και Rousseuw το 1990. Αυτός ο αλγόριθμος κατάτμησης τέθηκε σε εφαρμογή για την επίλυση του προβλήματος Partition Around Medoids (PAM). Η CLARA επεκτείνει την προσέγγιση K-Medoids για μεγάλο αριθμό αντικειμένων. Αυτή η τεχνική επιλέγει αυθαίρετα τα δεδομένα χρησιμοποιώντας το PAM.

Σύμφωνα με τους Raymond T. Ng και Jiawei Han τα ακόλουθα βήματα εκτελούνται στην περίπτωση της CLARA όπως δίνεται από τους συγγραφείς.

- Αντλήστε ένα δείγμα 40+2k αντικειμένων τυχαία από ολόκληρο το σύνολο δεδομένων και καλέστε τον αλγόριθμο PAM για να βρείτε k medoid του δείγματος.
- Για κάθε ένα από τα αντικείμενα προσδιορίστε το συγκεκριμένο K medoid που είναι παρόμοιο με το συγκεκριμένο αντικείμενο (Oj).
- Υπολογίστε τη μέση ανομοιότητα της ομαδοποίησης που προκύπτει με αυτόν τον τρόπο. Εάν η τιμή που λαμβάνεται με αυτόν τον τρόπο είναι μικρότερη από το παρόν ελάχιστο μπορούμε να τη χρησιμοποιήσουμε και να διατηρήσουμε το K-Medoid που βρέθηκε στο δεύτερο βήμα ως το καλύτερο από τα medoid.
- Μπορούμε να επαναλάβουμε τα βήματα για την' επόμενη επανάληψη'.

Πλεονεκτήματα του CLARA:

- Ο αλγόριθμος CLARA αντιμετωπίζει μεγαλύτερα σύνολα δεδομένων από τον PAM (Partition Around Mediods).

Μειονεκτήματα του CLARA:

- Η αποτελεσματική απόδοση του CLARA εξαρτάται από το μέγεθος του συνόλου δεδομένων.
- Ένα μεροληπτικό δείγμα δεδομένων μπορεί να οδηγήσει σε παραπλανητική και κακή ομαδοποίηση ολόκληρων συνόλων δεδομένων.

CLARANS

Ο αλγόριθμος K-Medoid δεν λειτουργεί αποτελεσματικά για μεγάλα σύνολα δεδομένων. Ως εκ τούτου, ο CLARA έχει βελτιωθεί και τροποποιηθεί έτσι ώστε να χρησιμοποιεί μεγάλες βάσεις δεδομένων. Ο CLARANS αναπτύχθηκε από τους Ng και Han το 1994. Για να ξεπεραστούν οι περιορισμοί του αλγορίθμου K-Medoid εισήχθη ο clarans. Ο Clarans (Clustering large Application Based on Randomized Search) είναι μέθοδος κατάτμησης που χρησιμοποιείται για μεγάλες βάσεις δεδομένων. Είναι πιο αποδοτική και κλιμακούμενη από τους PAM και CLARA. Όπως και στην περίπτωση του CLARA οι συγγραφείς έχουν συστήσει τα ακόλουθα βήματα:

- Παράμετροι εισόδου numlocal και maxneighbour.
- Επιλέξτε τυχαία το αντικείμενο K από το αντικείμενο D της βάσης δεδομένων.
- Χαρακτηρίστε αυτά τα K αντικείμενα ως επιλεγμένα Si και όλα τα άλλα ως μη επιλεγμένα S.
- Υπολογίστε το κόστος T για το επιλεγμένο Si.
- Εάν το T είναι αρνητικό, ενημερώστε το σύνολο medoid. Διαφορετικά το επιλεγμένο medoid επιλέγεται ως τοπικό βέλτιστο.
- Ξεκινήστε εκ νέου την επιλογή ενός άλλου συνόλου medoid και βρείτε ένα άλλο τοπικό βέλτιστο.
- Το CLARANS σταματά μέχρι να επιστρέψει το καλύτερο.

Σύμφωνα με τους συγγραφείς το CLARANS χρησιμοποιεί δύο παραμέτρους - numlocal και maxneighbour. Numlocal σημαίνει τον αριθμό των τοπικών ελαχίστων που λαμβάνονται και maxneighbhour σημαίνει τον μέγιστο αριθμό των εξεταζόμενων γειτόνων. Όσο μεγαλύτερη είναι η τιμή του τελευταίου, τόσο πιο κοντά θα είναι το CLARANS στο PAM και τόσο μεγαλύτερη θα είναι η διάρκεια κάθε αναζήτησης τοπικών ελαχίστων. Αυτό αποτελεί πλεονέκτημα επειδή η ποιότητα των τοπικών ελαχίστων είναι υψηλότερη και πρέπει να βρεθεί μικρότερος αριθμός τοπικών ελαχίστων.

Πλεονεκτήματα του CLARANS:

- Είναι εύκολος ο χειρισμός των ακραίων τιμών.
- Το αποτέλεσμα του CLARANS είναι πιο αποτελεσματικό σε σύγκριση με το PAM και το CLARA.

Μειονεκτήματα του CLARANS:

- Δεν εγγυάται την αναζήτηση σε μια εντοπισμένη περιοχή.
- Χρησιμοποιεί τυχαία δείγματα για τους γείτονες.
- Δεν είναι πολύ αποτελεσματική για μεγάλα σύνολα δεδομένων.[6f]

2.2 Αλγόριθμοι ιεραρχικής ομαδοποίησης (Hierarchical Clustering).

Οι αλγόριθμοι ομαδοποίησης με κατάτμησης βρίσκουν όλες τις ομάδες ταυτόχρονα ως ένα τμήμα των δεδομένων και δεν επιβάλλουν μια ιεραρχική δομή. Οι αλγόριθμοι ιεραρχικής ομαδοποίησης βρίσκουν φωλιασμένες συστάδες. Τύποι αλγορίθμων ιεραρχικής συσταδοποίησης είναι οι εξής:

1) Συσσωρευτική λειτουργία (Agglomerative mode) είναι μια μέθοδος ομαδοποίησης από κάτω προς τα πάνω, ξεκινάμε με ένα μόνο σημείο δεδομένων ως δική του συστάδα και συγχωνεύουμε διαδοχικά το πιο παρόμοιο ζεύγος συστάδων μέχρι να προκύψει μια τελική συστάδα που έχει όλα τα σημεία δεδομένων.

2) Διαχωριστική λειτουργία (Divisive mode) είναι μέθοδος συσταδοποίησης από πάνω προς τα κάτω, ξεκινάμε με όλα τα σημεία δεδομένων που περιέχονται ως μία συστάδα και διαιρούμε αναδρομικά κάθε συστάδα σε μικρότερες συστάδες.

Βάση της ιεραρχικής συσταδοποίησης είναι ότι η λύση βρίσκεται σε ιεραρχία ξεκινώντας από 'n' ομάδες σε 1 ομάδα ή αντίστροφα. Αρχικά κάθε σημείο είναι μια ομάδα, έχουμε έναν πίνακα απόστασης μεταξύ κάθε σημείου που στην πραγματικότητα είναι ο πίνακας απόστασης μεταξύ των ομάδων. Στη συνέχεια επιλέξαμε την απόσταση που είναι η μικρότερη και φέραμε αυτά τα δύο σημεία μαζί ή αυτές τις δύο ομάδες μαζί και σχηματίσαμε μια νέα ομάδα. Τώρα βρίσκουμε την επόμενη ομάδα και ο πίνακας αποστάσεων αλλάζει. Στην συνέχεια βρίσκουμε την απόσταση μεταξύ της ομάδας που σχηματίστηκε και όλων των άλλων σημείων. Υπάρχουν διάφοροι τρόποι υπολογισμού αυτής της απόστασης μεταξύ μιας ομάδας και ενός σημείου. Αυτή η ισοδύναμη απόσταση μπορεί να γίνει με περισσότερους από έναν τρόπους, δηλ. είτε για να ληφθεί η ελάχιστη απόσταση, η μέση απόσταση ή η μέγιστη απόσταση. Εάν επιλέξουμε την ομαδοποίηση μονής σύνδεσης, τείνουμε να επιλέξουμε την ελάχιστη απόσταση του σημείου. Η ομαδοποίηση μέσης σύνδεσης επιλέγει τη μέση απόσταση εντός της συστάδας από κάποιο άλλο σημείο εκτός της συστάδας. Η πλήρης σύνδεση επιλέγει τη μεγαλύτερη απόσταση από οποιοδήποτε μέλος μιας συστάδας σε οποιοδήποτε μέλος άλλης συστάδας.

Το σύνολο της λύσης είναι δεδομένο εάν μας δίνεται ένας τρόπος εύρεσης της απόστασης και ένας τρόπος συσχέτισης της απόστασης της ομάδας με την ατομική απόσταση. Στη βιβλιογραφία υπάρχει μεγάλος αριθμός αλγορίθμων ιεραρχικής συσταδοποίησης, αλλά διαφέρουν μόνο σε δύο σημεία, πρώτον στον τρόπο υπολογισμού του συντελεστή ομοιότητας ή του μέτρου απόστασης και δεύτερον, μπορεί να είναι απλή σύνδεση, πλήρης σύνδεση ή μέση σύνδεση. Έτσι, μπορούμε να πούμε ότι υπάρχουν εννέα διαφορετικές εκδοχές αλγορίθμων συσταδοποίησης. Οι οποίες περιλαμβάνουν τρεις τρόπους υπολογισμού του συντελεστή ομοιότητας ή της απόστασης (χρήση του συντελεστή Jacards, υπολογισμός της ομοιότητας με βάση τον

αριθμό των συστατικών που επισκέπτονται και τις δύο μηχανές, πίνακας ανομοιότητας ή απόστασης) και τρεις τρόπους ορισμού της απόστασης ομάδας έναντι της ατομικής απόστασης (ελάχιστη, μέση, μέγιστη).

Το σημαντικότερο μειονέκτημα της ιεραρχικής ομαδοποίησης είναι ότι μόλις τα δύο σημεία συνδεθούν, δεν μεταφέρονται σε άλλη ομάδα σε μια ιεραρχία ή ένα δέντρο. Υπάρχουν λίγοι αλγόριθμοι που χρησιμοποιούν ιεραρχική ομαδοποίηση με κάποιες παραλλαγές. Είναι οι εξής: BIRCH, CURE, ROCK και CHAMELEON. [15]

BIRCH

Το BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) [8] εισάγει μια νέα ιεραρχική δομή δεδομένων, το CF-tree, για τη συμπίεση των δεδομένων σε πολλές μικρές υπο-συστάδες και στη συνέχεια εκτελεί ομαδοποίηση με αυτές τις περιλήψεις αντί για τα ακατέργαστα δεδομένα. Οι υπο-συστάδες αντιπροσωπεύονται από συμπαγείς περιλήψεις, που ονομάζονται χαρακτηριστικά συστάδας (CF) και αποθηκεύονται στα φύλλα. Οι κόμβοι χωρίς φύλλα αποθηκεύουν τα αθροίσματα των CF των παιδιών τους. Ένα δέντρο CF κατασκευάζεται δυναμικά και σταδιακά, απαιτώντας μια απλή σάρωση του συνόλου δεδομένων. Ένα αντικείμενο εισάγεται στην πλησιέστερη καταχώρηση φύλλου. Δύο παράμετροι εισόδου ελέγχουν τον μέγιστο αριθμό παιδιών ανά κόμβο μη φύλλου και τη μέγιστη διάμετρο των υποσυστάδων που αποθηκεύονται στα φύλλα. Μεταβάλλοντας αυτές τις παραμέτρους, το BIRCH μπορεί να δημιουργήσει μια δομή που να χωράει στην κύρια μνήμη. Μόλις δημιουργηθεί το CF-tree, οποιοιδήποτε αλγόριθμοι κατάτμησης ή ιεραρχικοί αλγόριθμοι μπορούν να το χρησιμοποιήσουν για να εκτελέσουν ομαδοποίηση στην κύρια μνήμη. Το BIRCH είναι αρκετά γρήγορο, αλλά έχει δύο σοβαρά μειονεκτήματα: ευαισθησία στη σειρά των δεδομένων και αδυναμία αντιμετώπισης μη σφαιρικών συστάδων διαφορετικού μεγέθους, επειδή χρησιμοποιεί την έννοια της διαμέτρου για τον έλεγχο των ορίων μιας συστάδας.

Ο αλγόριθμος ομαδοποίησης BIRCH αποτελείται από δύο στάδια:

- Κατασκευή του δέντρου CF: Το BIRCH συνοψίζει μεγάλα σύνολα δεδομένων σε μικρότερες, πυκνές περιοχές που ονομάζονται καταχωρήσεις χαρακτηριστικών συσταδοποίησης (CF). Τυπικά, μια εγγραφή χαρακτηριστικών συσταδοποίησης ορίζεται ως μια διατεταγμένη τριπλέτα (N, LS, SS) όπου «N» είναι ο αριθμός των σημείων δεδομένων στη συστάδα, «LS» είναι το γραμμικό άθροισμα των σημείων δεδομένων και «SS» είναι το τετραγωνικό άθροισμα των σημείων δεδομένων στη συστάδα. Μια εγγραφή CF μπορεί να αποτελείται από άλλες εγγραφές CF. Προαιρετικά, μπορούμε να συμπυκνώσουμε αυτό το αρχικό δέντρο CF σε ένα μικρότερο CF.

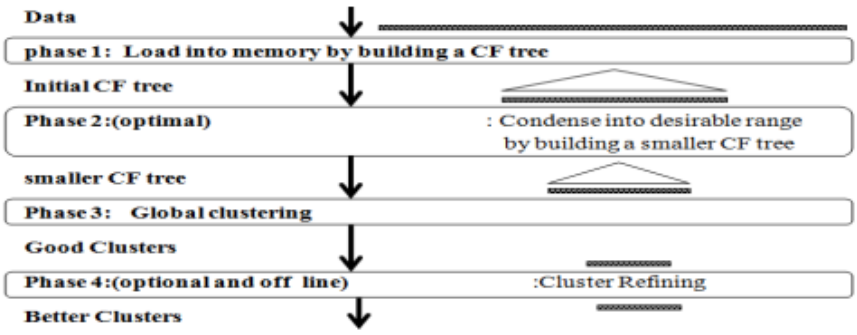
- Παγκόσμια συσταδοποίηση: Εφαρμόζει έναν υπάρχοντα αλγόριθμο συσταδοποίησης στα φύλλα του δέντρου CF. Ένα δέντρο CF είναι ένα δέντρο όπου κάθε κόμβος φύλλου περιέχει μια υπο-συστάδα. Κάθε εγγραφή σε ένα δέντρο CF περιέχει έναν δείκτη σε έναν κόμβο-παιδί και μια εγγραφή CF που αποτελείται από το άθροισμα των εγγραφών CF στους κόμβους-παιδιά. Προαιρετικά, μπορούμε να βελτιώσουμε αυτές τις συστάδες.

Πλεονεκτήματα του BIRCH:

- Ενσωματωμένη ευελιξία όσον αφορά το επίπεδο λεπτομέρειας.
- Κατάλληλο για προβλήματα που περιλαμβάνουν σημειακές συνδέσεις, π.χ. δέντρα ταξινόμησης.

Μειονεκτήματα του BIRCH:

- Αδυναμία διορθώσεων μόλις ληφθεί η απόφαση διάσπασης/συγχώνευσης.
- Έλλειψη ερμηνευσιμότητας όσον αφορά τους περιγραφείς των συστάδων.
- Ασάφεια του κριτηρίου τερματισμού.
- Απαγορευτικά δαπανηρή για σύνολα δεδομένων υψηλής διάστασης και τεράστιας κλίμακας.
- Σοβαρή υποβάθμιση της αποτελεσματικότητας σε χώρους υψηλής διάστασης λόγω του φαινομένου της διαστασιολόγησης.[11]



Εικόνα 17. Λειτουργία του BIRCH

CURE

Η ομαδοποίηση με χρήση αντιπροσώπων (CURE) είναι μια συσσωρευτική μέθοδος που εισάγει δύο καινοτομίες. Πρώτον, οι συστάδες αντιπροσωπεύονται από έναν σταθερό αριθμό καλά διασκορπισμένων σημείων αντί για ένα ενιαίο κεντροειδές. Δεύτερον, οι αντιπρόσωποι συρρικνώνονται προς τα κέντρα των συστάδων τους κατά έναν σταθερό παράγοντα. Σε κάθε επανάληψη, το ζεύγος συστάδων με τους

πλησιέστερους αντιπροσώπους συγχωνεύεται. Η χρήση πολλαπλών αντιπροσώπων επιτρέπει στο CURE να αντιμετωπίζει συστάδες αυθαίρετου σχήματος διαφορετικών μεγεθών, ενώ η συρρίκνωση αποσβένει τις επιδράσεις των ακραίων τιμών και του θορύβου. Το CURE χρησιμοποιεί έναν συνδυασμό τυχαίας δειγματοληψίας και κατάτμησης για τη βελτίωση της επεκτασιμότητας [11].

Ο CURE (Clustering Using REpresentatives) είναι ένας αλγόριθμος ομαδοποίησης δεδομένων για μεγάλες βάσεις δεδομένων που είναι πιο ανθεκτικός στις ακραίες τιμές και καταγράφει ομάδες διαφορετικών σχημάτων και μεγεθών. Αποδίδει καλά σε σύνολο δεδομένων 2 διαστάσεων. Η χρονική πολυπλοκότητά του είναι O(n2 log n).Ο BIRCH και ο CURE χειρίζονται καλά τις ακραίες τιμές. Ο BIRCH έχει καλύτερη χρονική πολυπλοκότητα αλλά υστερεί σε ποιότητα συστάδων από τον αλγόριθμο CURE [9f].

Ο αλγόριθμος CURE περιλαμβάνει τα ακόλουθα βήματα:

- Τυχαία δειγματοληψία.
    I. Προκειμένου να αντιμετωπιστούν μεγάλα σύνολα δεδομένων, χρησιμοποιείται τυχαία δειγματοληψία για τη μείωση του μεγέθους της εισόδου στον αλγόριθμο ομαδοποίησης του CURE.
    II. Το [Vit85] παρέχει αποτελεσματικούς αλγορίθμους για την τυχαία κλήρωση ενός δείγματος σε ένα πέρασμα και τη χρήση σταθερού χώρου.
    III. Αν και η τυχαία δειγματοληψία έχει συμβιβασμό μεταξύ ακρίβειας και αποτελεσματικότητας, τα πειράματα δείχνουν ότι για τα περισσότερα σύνολα δεδομένων, με μέτριου μεγέθους τυχαία δείγματα, μπορούν να προκύψουν πολύ καλές συστάδες [16].
- Κατάτμηση για επιτάχυνση.
  Η βασική ιδέα είναι να κατατμηθεί ο δειγματικός χώρος σε p κατατμήσεις. Κάθε διαμέρισμα περιέχει n/p στοιχεία. Το πρώτο πέρασμα ομαδοποιεί μερικώς κάθε διαμέριση μέχρι ο τελικός αριθμός των ομαδοποιήσεων να μειωθεί σε n/pq για κάποια σταθερά q ≥ 1. Ένα δεύτερο πέρασμα ομαδοποίησης σε n/q ομαδοποιεί μερικώς τα χωρίσματα. Για το δεύτερο πέρασμα αποθηκεύονται μόνο τα αντιπροσωπευτικά σημεία, δεδομένου ότι η διαδικασία συγχώνευσης απαιτεί μόνο αντιπροσωπευτικά σημεία προηγούμενων συστάδων πριν από τον υπολογισμό των αντιπροσωπευτικών σημείων για τη συγχωνευμένη συστάδα. Η κατάτμηση της εισόδου μειώνει τους χρόνους εκτέλεσης.
- Επισήμανση δεδομένων στο δίσκο.
  Δεδομένου ότι υπάρχουν μόνο αντιπροσωπευτικά σημεία για k συστάδες, τα υπόλοιπα σημεία δεδομένων αντιστοιχίζονται επίσης στις συστάδες. Για το σκοπό αυτό επιλέγεται ένα κλάσμα τυχαία επιλεγμένων αντιπροσωπευτικών

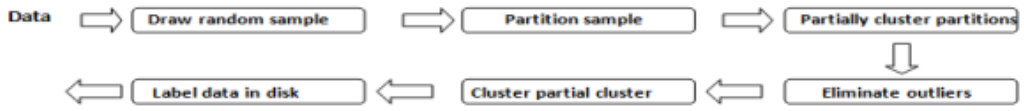
σημείων για κάθε μία από τις k συστάδες και το σημείο δεδομένων ανατίθεται στη συστάδα που περιέχει το αντιπροσωπευτικό σημείο που βρίσκεται πλησιέστερα σε αυτό.[17]

Πλεονεκτήματα του CURE

- Οι συστάδες τυχαίου σχήματος μπορούν εύκολα να αναγνωριστούν από τον αλγόριθμο ομαδοποίησης.
- Ο CURE είναι πιο ανθεκτικό στην παρουσία ακραίων τιμών.
- Για μικρότερο μέγεθος εισόδου (π.χ. 3) η πολυπλοκότητά του είναι $O(n^2)$.
- Είναι κατάλληλος αλγόριθμος για μεγάλα σύνολα δεδομένων.
- Ο CURE εκτελεί όλη τη διαδικασία χωρίς να θυσιάζει την ποιότητα της ομαδοποίησης.

Μειονεκτήματα του αλγορίθμου CURE

- Οι πληροφορίες σχετικά με τη συνολική διασυνδεσιμότητα των αντικειμένων σε δύο συστάδες αγνοούνται από το CURE.[18]



Εικόνα 18. Λειτουργία του CURE

ROCK

Ο αλγόριθμος ROCK ένας ισχυρός αλγόριθμος ιεραρχικής ομαδοποίησης είναι ένας συσσωρευτικός αλγόριθμος ιεραρχικής ομαδοποίησης που βασίζεται στην έννοια των συνδέσμων. Είναι κατάλληλος για το χειρισμό μεγάλων συνόλων δεδομένων. Ο ROCK συνδυάζει, από μια εννοιολογική άποψη, τις μεθόδους του πλησιέστερου γείτονα, της μετεγκατάστασης και της ιεραρχικής συσσωμάτωσης. Σε αυτόν τον αλγόριθμο, η ομοιότητα των συστάδων βασίζεται στον αριθμό των σημείων από διαφορετικές συστάδες που έχουν κοινούς γείτονες. Η πολυπλοκότητα χώρου του αλγορίθμου εξαρτάται από το αρχικό μέγεθος των τοπικών σωρών. Επομένως, η πολυπλοκότητα χώρου του ROCK αλγορίθμου συσταδοποίησης είναι $O(\min \{n^2, nm_m m_a\})$, όπου n είναι ο αριθμός των σημείων εισόδου, m_a και m_m είναι οι μέσος και μέγιστος αριθμός γειτόνων για ένα σημείο, αντίστοιχα. Έχει και χρόνο χειρότερης περίπτωσης πολυπλοκότητας $O(n^2 + nm_m m_a + n^2 \log n)$. Ένας εύρωστος αλγόριθμος ιεραρχικής ομαδοποίησης ROCK ήταν που χρησιμοποιεί συνδέσμους και όχι αποστάσεις για τη συγχώνευση συστάδων. Προτάθηκε μια γρήγορη έκδοση του αλγορίθμου ROCK για την ομαδοποίηση κατηγορικών δεδομένων, η οποία

ονομάζεται QROCK. Έχει πολυπλοκότητα O(n2). Οι αναλύσεις επιδόσεων καταδεικνύουν επίσης ότι ο QROCK είναι γρηγορότερος από τον ROCK.[19]

Ο αλγόριθμος ROCK είναι ο καταλληλότερος αλγόριθμος για την ομαδοποίηση κατηγορικών δεδομένων επειδή μπορεί να χρησιμοποιήσει Jaccard ή Cosine για να διαπιστώσει την ομοιότητα μεταξύ των δύο σημείων δεδομένων και επιπλέον χρησιμοποιεί την ιδέα των συνδέσμων για τον προσδιορισμό των γειτόνων. Είναι δύσκολη η διαχείριση και ο χειρισμός των μεγάλων κομματιών δεδομένων- επομένως η ομαδοποίηση μπορεί να βοηθήσει στην ομαδοποίησή τους. Αυτό που έχουμε παρατηρήσει σε γενικές γραμμές είναι ότι το έργο της εύρεσης ή αναζήτηση κάποιου εγγράφου από τον μεγάλο όγκο δεδομένων είναι δυσχερής. Επίσης, ο χρόνος απόκρισης της αναζήτησης του εγγράφου είναι πολύ υψηλός λόγω της μεγάλης κλίμακας αναζήτησης μεταξύ των δεδομένων. Έτσι, η προσέγγισή μας είναι η ομαδοποίηση των δεδομένων σε προκειμένου να χωρίσουμε τα δεδομένα σε ορισμένες ομάδες με παρόμοια χαρακτηριστικά και ως εκ τούτου να μειώσουμε το χρόνο απόκρισης του ερωτήματος κατά αναζητώντας τις συστάδες που προκύπτουν αντί για ολόκληρη τη βάση δεδομένων ή την αποθήκη δεδομένων. Το έργο αυτό λειτουργεί σε δύο στάδια:

- Ομαδοποίηση των δεδομένων με τον αλγόριθμο ROCK και αποθήκευση των ομάδων.
- Μείωση του χρόνου απόκρισης ή αναζήτησης ερωτημάτων παρέχοντας τα αποτελέσματα από τις συστάδες που λαμβάνονται αντί για τις συστάδες, αντί της βάσης δεδομένων. [20]

Εικόνα 19. Λειτουργία του ROCK

CHAMELEON

Το CHAMELEON βελτιώνει την ποιότητα της ομαδοποίησης χρησιμοποιώντας πιο περίπλοκα κριτήρια συγχώνευσης σε σύγκριση με τον CURE. Αρχικά, δημιουργείται ένας γράφος που περιέχει συνδέσεις μεταξύ κάθε σημείου και των k-κοντινότερων γειτόνων του. Στη συνέχεια, ένας αλγόριθμος κατάτμησης γράφου διαιρεί αναδρομικά το γράφο σε πολλούς μικρούς μη συνδεδεμένους υπο-γράφους. Κατά τη δεύτερη φάση, κάθε υπο-γράφος αντιμετωπίζεται ως αρχική υπο-ομάδα και ένας συσσωρευτικός ιεραρχικός αλγόριθμος συνδυάζει επανειλημμένα τις δύο πιο παρόμοιες ομάδες. Δύο συστάδες είναι επιλέξιμες για συγχώνευση μόνο εάν η συστάδα που προκύπτει έχει παρόμοια διασυνδεσιμότητα και εγγύτητα με τις δύο

μεμονωμένες συστάδες πριν από τη συγχώνευση. Λόγω του δυναμικού μοντέλου συγχώνευσης, το CHAMELEON είναι πιο αποτελεσματικό από το CURE στην ανακάλυψη συστάδων αυθαίρετου σχήματος με διαφορετική πυκνότητα. Ωστόσο, η βελτιωμένη αποτελεσματικότητα αποβαίνει εις βάρος του υπολογιστικού κόστους που είναι τετραγωνικό σε σχέση με το μέγεθος της βάσης δεδομένων [11].

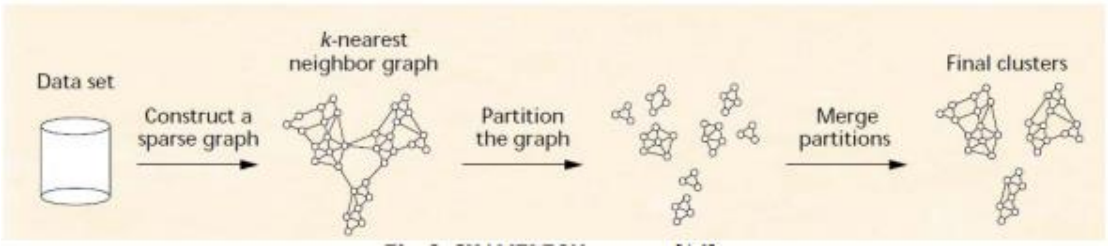
Ο CHAMELEON βασίζεται σε δύο φάσεις: στην αρχή χωρίζει τα σημεία δεδομένων σε υπο-συστάδες, χρησιμοποιώντας ένα γράφημα κατάτμησης και στη συνέχεια επανειλημμένα συγχωνεύει υπο-συστάδες από το προηγούμενο στάδιο για να λάβει τις τελικές συστάδες. Ο αλγόριθμος έχει αποδειχθεί ότι βρίσκει συστάδες με ποικίλα σχήματα, πυκνότητες και μεγέθη σε δισδιάστατο χώρο. Ο Chameleon είναι ένας αποτελεσματικός αλγόριθμος που χρησιμοποιεί ένα δυναμικό μοντέλο για να αποκτήσει συστάδες αυθαίρετων σχημάτων και αυθαίρετων πυκνοτήτων. Ο αλγόριθμος είναι κατάλληλος για εφαρμογές όπου ο όγκος των διαθέσιμων δεδομένων είναι μεγάλος. Για μεγάλο n, η χρονική πολυπλοκότητα χειρότερης περίπτωσης του αλγορίθμου είναι $O(n(\log_2 n + m))$, όπου m είναι ο αριθμός των συστάδων που σχηματίζονται μετά την ολοκλήρωση της πρώτης φάσης του αλγορίθμου.

Πλεονεκτήματα του αλγορίθμου CHAMELEON

- Αυτός ο αλγόριθμος είναι πολύ αποτελεσματικός στο χειρισμό μεγάλου όγκου δεδομένων.
- Είναι ένας αποτελεσματικός αλγόριθμος που παρέχει συστάδες αυθαίρετου σχήματος και αυθαίρετου μεγέθους.

Μειονεκτήματα του αλγορίθμου CHAMELEON

- Το CHAMELEON είναι γνωστό για χώρους χαμηλών διαστάσεων και δεν εφαρμόστηκε σε χώρους υψηλών διαστάσεων. Η χρονική πολυπλοκότητα του αλγορίθμου CHAMELEON σε υψηλές διαστάσεις είναι $O(n_2)$.



Εικόνα 20. Λειτουργία του CHAMELEON

DIANA

Ο αλγόριθμος DIANA Ο DIANA είναι μια τεχνική ιεραρχικής ομαδοποίησης που κατασκευάζει την ιεραρχία με αντίστροφη σειρά. Προσεγγίζει τον αντίστροφο αλγόριθμο της συσσωρευτικής ιεραρχικής ομαδοποίησης. Υπάρχει μια μεγάλη συστάδα που αποτελείται από όλα τα n αντικείμενα. Σε κάθε επόμενο βήμα, η μεγαλύτερη διαθέσιμη συστάδα χωρίζεται σε δύο συστάδες, έως ότου τελικά όλες οι συστάδες, αποτελούνται από μεμονωμένα αντικείμενα. Έτσι, η ιεραρχία οικοδομείται σε n - 1 βήματα. Στο πρώτο βήμα μιας συσσωρευτικής μεθόδου, εξετάζονται όλες οι πιθανές συνενώσεις δύο αντικειμένων που οδηγούν σε n (n - 1)/2 συνδυασμούς. Στη διαιρετική μέθοδο που βασίζεται στην ίδια αρχή, υπάρχουν 2n-1 - 1 δυνατότητες διαχωρισμού των δεδομένων σε δύο συστάδες. Ο αριθμός αυτός είναι σημαντικά μεγαλύτερος από αυτόν στην περίπτωση της συσσωρευτικής μεθόδου. Για την αποφυγή τόσο μεγάλων υπολογισμών ακολουθήθηκαν τα ακόλουθα βήματα:

- Η ομαδοποίηση DIANA ακολουθείται από τη συσσωρευτική ιεραρχική ομαδοποίηση μέχρι τη συστάδα που περιέχει όλα τα αντικείμενα. Στη συνέχεια, η συσταδοποίηση διαιρετικής ανάλυσης (DIANA) ακολουθεί την από πάνω προς τα κάτω προσέγγιση υποθέτοντας ότι είναι μία μόνο συστάδα που έχει επίπεδο L (0) = n και αριθμό ακολουθίας m = 0.
- Βρίσκεται το πιο ανόμοιο ζεύγος συστάδων στην τρέχουσα συστάδα, δηλαδή (r), (s) στο οποίο d [(r), (s)] = min d [(i), (j)], όπου min είναι τα πλήρη ζεύγη συστάδων στην τρέχουσα συστάδα.
- Ο αριθμός ακολουθίας αυξάνεται με τον τρόπο m = m + 1. Η συστάδα διασπάται σε συστάδες (r) και (s) για να σχηματιστεί η επόμενη συστάδα ώστε να γίνει το επίπεδο της ομαδοποίησης: L (m1) = d [(r)] και L (m2) = d [(s)].
- Ο πίνακας αποστάσεων (D) ενημερώνεται προσθέτοντας τις γραμμές και τις στήλες που αντιστοιχούν στις συστάδες (r) και (s). Η ομοιότητα μεταξύ της νέας συστάδας, που συμβολίζεται με (r, s) και της παλιάς συστάδας (k) ορίζεται με αυτόν τον τρόπο:
  D[(k),(r,s)] = min d[(k),(r)], d[(k),(s)]
- Εάν όλα τα αντικείμενα είναι ξεχωριστές συστάδες, τότε σταματήστε- διαφορετικά προχωρήστε στο βήμα-2.[21]

2.3 Αλγόριθμοι ομαδοποίησης με βάση την πυκνότητα (Density-Based Clustering).

Οι μέθοδοι ομαδοποίησης με βάση την πυκνότητα ομαδοποιούν γειτονικά αντικείμενα σε ομάδες με βάση τις τοπικές συνθήκες πυκνότητας και όχι την εγγύτητα μεταξύ των αντικειμένων. Αυτές οι μέθοδοι θεωρούν τις συστάδες ως πυκνές

περιοχές που χωρίζονται από θορυβώδεις περιοχές χαμηλής πυκνότητας. Οι μέθοδοι με βάση την πυκνότητα έχουν ανοχή στο θόρυβο και μπορούν να ανακαλύψουν μη κυρτές συστάδες [11]. Μια τιμή πυκνότητας σχετίζεται με κάθε αντικείμενο, η οποία αξιολογείται ως ο αριθμός των γειτονικών του αντικειμένων εντός μιας δεδομένης ακτίνας. Η ποιότητα αυτών των τεχνικών δεν επηρεάζεται από τις ακραίες τιμές και το σχήμα της συστάδας. Παρόμοια με την προσέγγιση με βάση το πλέγμα, η συσταδοποίηση με βάση την πυκνότητα είναι επίσης αποτελεσματική στο χειρισμό δεδομένων υψηλής διάστασης [9].

DBSCAN

Ο αλγόριθμος DBSCAN εισήγαγε ανεξάρτητα την ομαδοποίηση με βάση την πυκνότητα στην κοινότητα της επιστήμης των υπολογιστών, προτείνοντας επίσης τη χρήση δομών χωρικών δεικτών για την επίτευξη ενός κλιμακούμενου αλγορίθμου ομαδοποίησης. Υποθέτοντας ένα κατώφλι απόστασης r και ένα κατώφλι πυκνότητας k, ο DBSCAN, όπως και η μέθοδος Wishart, εκτιμά την πυκνότητα για κάθε σημείο xi ως τον αριθμό ki των σημείων που βρίσκονται εντός μιας ακτίνας r γύρω από το x δηλαδή αναζητά κεντρικά αντικείμενα των οποίων η γειτονιά (ακτίνα) περιέχει τουλάχιστον Minpts σημεία [9]. Ως σημεία πυρήνα χαρακτηρίζονται τα σημεία δεδομένων για τα οποία ki > k. Τα σημεία θεωρούνται άμεσα συνδεδεμένα εάν η μεταξύ τους απόσταση δεν είναι μεγαλύτερη από r. Οι συστάδες με βάση την πυκνότητα χαρακτηρίζονται ως μέγιστα συνδεδεμένες συνιστώσες του συνόλου των σημείων που βρίσκονται σε απόσταση r από κάποιο αντικείμενο πυρήνα (δηλ, μια συστάδα μπορεί να περιέχει σημεία xi με ki < k, τα οποία ονομάζονται συνοριακά αντικείμενα, εάν βρίσκονται σε απόσταση r από ένα κεντρικό αντικείμενο της συστάδας). Τα αντικείμενα που δεν ανήκουν σε μια συστάδα θεωρούνται θόρυβος. Ο αλγόριθμος DBSCAN κατασκευάζει συστάδες επαναληπτικά, ξεκινώντας μια νέα συστάδα C με ένα μη εκχωρημένο κεντρικό αντικείμενο x και αναθέτοντας όλα τα σημεία στην C που συνδέονται άμεσα ή μεταβατικά με το x. Για τον προσδιορισμό των άμεσα και μεταβατικά συνδεδεμένων σημείων για ένα δεδομένο σημείο, χρησιμοποιείται μια δομή χωρικού ευρετηρίου για την εκτέλεση ερωτημάτων εύρους με ακτίνα r για κάθε αντικείμενο που προστίθεται πρόσφατα σε μια τρέχουσα συστάδα, με αποτέλεσμα μια ευφυής πολυπλοκότητα χρόνου εκτέλεσης για μέτριας διάστασης δεδομένα O(N log N), όπου N είναι ο συνολικός αριθμός σημείων στο σύνολο δεδομένων, και ένας χειρότερος χρόνος εκτέλεσης $O(N^2)$, π.χ., για δεδομένα υψηλής διάστασης, όταν η απόδοση των δομών χωρικών ευρετηρίων επιδεινώνεται.[22]

Πλεονεκτήματα του DBSCAN

- Ο DBSCAN δεν απαιτεί να καθοριστεί εκ των προτέρων ο αριθμός των συστάδων στα δεδομένα, σε αντίθεση με τον k-means.
- Μπορεί να βρει συστάδες αυθαίρετου σχήματος. Μπορεί ακόμη και να βρει μια συστάδα που περιβάλλεται πλήρως από μια διαφορετική συστάδα (αλλά δεν συνδέεται με αυτήν). Λόγω της παραμέτρου MinPts, μειώνεται το λεγόμενο φαινόμενο της μονής σύνδεσης (διαφορετικές συστάδες που συνδέονται με μια λεπτή γραμμή σημείων).
- Ο DBSCAN έχει την έννοια του θορύβου και είναι ανθεκτική στις ακραίες τιμές.
- Ο DBSCAN απαιτεί μόνο δύο παραμέτρους και είναι ως επί το πλείστον αναίσθητη στη διάταξη των σημείων στη βάση δεδομένων. (Ωστόσο, τα σημεία που βρίσκονται στην άκρη δύο διαφορετικών συστάδων ενδέχεται να ανταλλάξουν την ιδιότητα μέλους συστάδας εάν αλλάξει η διάταξη των σημείων, και η ανάθεση συστάδας είναι μοναδική μόνο μέχρι ισομορφισμού).
- Είναι σχεδιασμένο για χρήση με βάσεις δεδομένων που μπορούν να επιταχύνουν τα ερωτήματα περιοχής, π.χ. χρησιμοποιώντας ένα δέντρο R*.
- Οι παράμετροι minPts και ε μπορούν να οριστούν από έναν εμπειρογνώμονα του τομέα, εάν τα δεδομένα είναι καλά κατανοητά.

Μειονεκτήματα του DBSCAN

- Ο DBSCAN δεν είναι εντελώς ντετερμινιστική: τα σημεία των ορίων που είναι προσβάσιμα από περισσότερες από μία συστάδες μπορεί να ανήκουν σε οποιαδήποτε από τις δύο συστάδες, ανάλογα με τη σειρά επεξεργασίας των δεδομένων.
- Η ποιότητα του DBSCAN εξαρτάται από το μέτρο απόστασης που χρησιμοποιείται στη συνάρτηση regionQuery(P,ε). Το πιο συνηθισμένο μέτρο απόστασης που χρησιμοποιείται είναι η ευκλείδεια απόσταση. Ειδικά για δεδομένα υψηλής διάστασης, αυτή η μετρική μπορεί να καταστεί σχεδόν άχρηστη λόγω της λεγόμενης "κατάρας της διάστασης", καθιστώντας δύσκολη την εύρεση μιας κατάλληλης τιμής για το ε. Αυτό το φαινόμενο, ωστόσο, υπάρχει και σε οποιονδήποτε άλλο αλγόριθμο που βασίζεται στην ευκλείδεια απόσταση.
- Ο DBSCAN δεν μπορεί να ομαδοποιήσει καλά σύνολα δεδομένων με μεγάλες διαφορές στις πυκνότητες, καθώς ο συνδυασμός minPts-ε δεν μπορεί τότε να επιλεγεί κατάλληλα για όλες τις συστάδες.
- Εάν τα δεδομένα και η κλίμακα δεν είναι καλά κατανοητά, η επιλογή ενός σημαντικού ορίου απόστασης ε μπορεί να είναι δύσκολη.

OPTICS

Το OPTICS (Ordering Points To Identify the Clustering Structure) είναι μια επέκταση του DBSCAN, αλλά σε αυτό η απαίτηση των παραμέτρων εισόδου δεν είναι τόσο

αυστηρή. Δημιουργεί μια διάταξη μιας βάσης δεδομένων, αποθηκεύοντας επιπλέον την απόσταση πυρήνα και μια κατάλληλη απόσταση προσπελασιμότητας για κάθε αντικείμενο. Δημιουργείται μια δομή ομαδοποίησης η οποία ορίζει ένα ευρύ φάσμα πιθανών τιμών και ομαδοποιεί αυτόματα και διαδραστικά τα δεδομένα. Το OPTICS υπολογίζει μια επαυξημένη διάταξη συστάδων η οποία έχει τις πληροφορίες για μια ζωηρή ποικιλία παραμέτρων, όπως στην ομαδοποίηση με βάση την πυκνότητα.

Ο OPTICS είναι ένας αλγόριθμος ομαδοποίησης με βάση την πυκνότητα και μπορεί να ανιχνεύσει σημαντικές ομάδες σε δεδομένα διαφορετικής πυκνότητας, παράγοντας μια γραμμική σειρά σημείων, έτσι ώστε τα σημεία που είναι χωρικά πλησιέστερα να γίνονται γείτονες κατά σειρά. Ο OPTICS ξεκινά με την προσθήκη ενός αυθαίρετου σημείου μιας συστάδας στον κατάλογο τάξης και στη συνέχεια επεκτείνει επαναληπτικά τη συστάδα προσθέτοντας ένα σημείο εντός ε - γειτονίας ενός σημείου της συστάδας το οποίο είναι επίσης πλησιέστερο σε οποιοδήποτε από τα ήδη επιλεγμένα σημεία. Η διαδικασία επαναλαμβάνεται για τις υπόλοιπες συστάδες. Εν τω μεταξύ, το OPTICS υπολογίζει επίσης την απόσταση προσπελασιμότητας για κάθε σημείο. Οι συστάδες για κάθε απόσταση ομαδοποίησης ε’ (ε’ < ε) μπορούν να εξαχθούν με βάση την υπολογισμένη σειρά και τις αποστάσεις προσπελασιμότητας.[23]

Πλεονεκτήματα του OPTICS

- Η ομαδοποίηση OPTICS δεν απαιτεί έναν προκαθορισμένο αριθμό συστάδων εκ των προτέρων.
- Οι συστάδες μπορούν να έχουν οποιοδήποτε σχήμα, συμπεριλαμβανομένων των μη σφαιρικών.
- Ικανότητα εντοπισμού ακραίων τιμών (δεδομένα θορύβου)

Μειονεκτήματα του OPTICS

- Αποτυγχάνει αν δεν υπάρχουν πτώσεις πυκνότητας μεταξύ των συστάδων.
- Είναι επίσης ευαίσθητο στις παραμέτρους που καθορίζουν την πυκνότητα (ακτίνα και ελάχιστος αριθμός σημείων) και οι κατάλληλες ρυθμίσεις των παραμέτρων απαιτούν γνώση του τομέα.

DENCLUE

Η ομαδοποίηση με βάση την πυκνότητα (DENCLUE) χρησιμοποιεί μια συνάρτηση επιρροής για να περιγράψει την επίδραση ενός σημείου σχετικά με τη γειτονιά του, ενώ η συνολική πυκνότητα του χώρου δεδομένων είναι το άθροισμα των συναρτήσεων επιρροής από όλα τα δεδομένα. Οι συστάδες προσδιορίζονται χρησιμοποιώντας ελκυστές πυκνότητας, τοπικά μέγιστα της συνολικής συνάρτησης πυκνότητας. Για τον υπολογισμό του αθροίσματος των συναρτήσεων επιρροής

χρησιμοποιείται μια δομή πλέγματος. Η DENCLUE κλιμακώνεται καλά (O(N)) και μπορεί να βρει συστάδες αυθαίρετου σχήματος, είναι ανθεκτική στο θόρυβο, είναι αναίσθητη στη διάταξη των δεδομένων, αλλά υποφέρει από την ευαισθησία της σε στις παραμέτρους εισόδου. Το φαινόμενο της κατάρας της διάστασης επηρεάζει σημαντικά την αποτελεσματικότητα της Denclue.[11]

Το DENsity-based CLUstEring (DENCLUE) χρησιμοποιεί χάρτη για τον υπολογισμό της συνάρτησης πυκνότητας και οι ακραίες τιμές θεωρούνται κύβοι με χαμηλή πυκνότητα και εξαλείφονται από τη διαδικασία ομαδοποίησης. Στη συνέχεια χρησιμοποιεί την τοπική συνάρτηση πυκνότητας για τον προσδιορισμό της συνδεσιμότητας των σημείων δεδομένων. Η απεικόνιση των δεδομένων με χάρτη ενισχύει τη συμπαγή μορφή των συστάδων και είναι επίσης υπολογιστικά αποδοτική για το χειρισμό μεγάλων συνόλων δεδομένων.

Ένας αλγόριθμος ομαδοποίησης Denclue ορίζεται από τα τοπικά μέγιστα της εκτιμώμενης συνάρτησης πυκνότητας. Η διαδικασία της αναρρίχησης (hill-climbing) ξεκινά για κάθε περίπτωση του σημείου δεδομένων, και αυτό αναθέτει την περίπτωση σε ένα τοπικό μέγιστο. Η αναρρίχηση καθοδηγείται από την κλίση της ∇Č(u) για έναν πυρήνα Gauss, η οποία έχει τη μορφή

$$\nabla \hat{C}(u) = \frac{1}{t^{d+2}} \sum_{J=1}^{N} k\left(\frac{u - u_j}{l}\right)(u_j - u)$$

Η διαδικασία για την αναρρίχηση ξεκινά από ένα σημείο δεδομένων και επαναλαμβάνεται μέχρι η πυκνότητα να παραμείνει αμετάβλητη ή να μην αυξηθεί περαιτέρω. Ο επικαιροποιημένος τύπος της επανάληψης που πρέπει να ακολουθηθεί δίνεται στην εξίσωση (ii) παρακάτω [9]:

$$u^{l+1} = u^{(1)} + \delta \frac{\nabla \hat{C}(u^1)}{\left\|\nabla \hat{C}(u^1)\right\|_2}$$

Πλεονεκτήματα του DENCLUE

- Ανακάλυψη συστάδων αυθαίρετου σχήματος με διαφορετικό μέγεθος.
- Αντοχή σε θόρυβο και ακραίες τιμές.

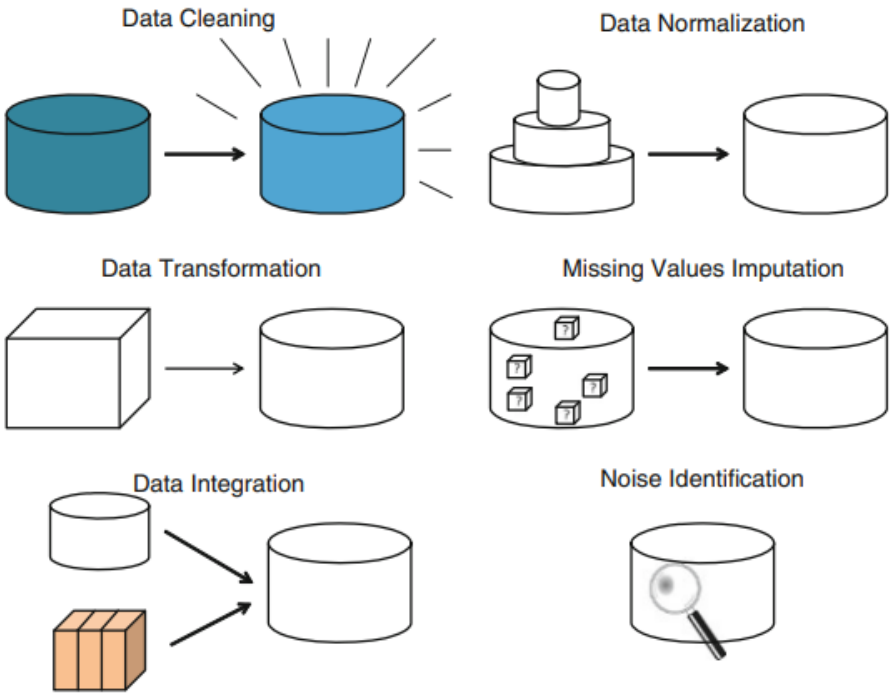
Μειονεκτήματα του DENCLUE

- Υψηλή ευαισθησία στη ρύθμιση των παραμέτρων εισόδου.
- Φτωχοί περιγραφείς συστάδων.
- Ακατάλληλο για σύνολα δεδομένων υψηλών διαστάσεων λόγω του φαινομένου της κατάρας της διαστατικότητας.

Κεφάλαιο 3 : Προεπεξεργασία Συνόλου Δεδομένων

3.1 Προετοιμασία δεδομένων (Data Preparation).

Η προετοιμασία δεδομένων είναι συνήθως ένα υποχρεωτικό βήμα. Μετατρέπει τα προηγούμενα άχρηστα δεδομένα σε νέα δεδομένα που ταιριάζουν σε μια διαδικασία DM (Data mining). Πρώτα απ' όλα, εάν τα δεδομένα δεν έχουν προετοιμαστεί, ο αλγόριθμος DM μπορεί να μην τα λάβει προκειμένου να λειτουργήσει ή σίγουρα θα αναφέρει σφάλματα κατά τη διάρκεια της εκτέλεσής του. Στην καλύτερη των περιπτώσεων, ο αλγόριθμος θα λειτουργήσει, αλλά τα προσφερόμενα αποτελέσματα δεν θα έχουν νόημα ή δεν θα θεωρηθούν ως ακριβής γνώση. [24]



Εικόνα 23. Προετοιμασία Δεδομένων

3.1.1 Καθαρισμός Δεδομένων (Data Cleaning).

Η εκκαθάριση δεδομένων, περιλαμβάνει λειτουργίες που διορθώνουν κακά δεδομένα, φιλτράρουν ορισμένα εσφαλμένα δεδομένα από το σύνολο δεδομένων και μειώνουν τις περιττές λεπτομέρειες των δεδομένων. Πρόκειται για μια γενική

έννοια που περιλαμβάνει ή επικαλύπτει άλλες γνωστές τεχνικές προετοιμασίας δεδομένων. Η αντιμετώπιση των δεδομένων που λείπουν και του θορύβου περιλαμβάνεται εδώ. Άλλες εργασίες καθαρισμού δεδομένων περιλαμβάνουν την ανίχνευση ασυμφωνιών και βρώμικων δεδομένων (τμήματα των αρχικών δεδομένων που δεν έχουν νόημα). Οι τελευταίες εργασίες σχετίζονται περισσότερο με την κατανόηση των αρχικών δεδομένων και γενικά απαιτούν ανθρώπινο έλεγχο.[24]

3.1.2 Μετασχηματισμός δεδομένων (Data Tranformation).

Σε αυτό το στάδιο προεπεξεργασίας, τα δεδομένα μετατρέπονται ή ενοποιούνται έτσι ώστε το αποτέλεσμα της διαδικασίας εξόρυξης να μπορεί να εφαρμοστεί ή να είναι πιο αποτελεσματικό. Τα επιμέρους καθήκοντα εντός του μετασχηματισμού των δεδομένων είναι η εξομάλυνση, η κατασκευή χαρακτηριστικών, η συγκέντρωση ή η σύνοψη των δεδομένων, η κανονικοποίηση, η διακριτοποίηση και η γενίκευση. Οι περισσότερες από αυτές θα διαχωριστούν ως ανεξάρτητες εργασίες, λόγω του γεγονότος ότι ο μετασχηματισμός δεδομένων, όπως η περίπτωση του καθαρισμού δεδομένων, αναφέρεται ως μια γενική οικογένεια τεχνικών προεπεξεργασίας δεδομένων. Οι εργασίες που απαιτούν ανθρώπινη επίβλεψη και εξαρτώνται περισσότερο από τα δεδομένα είναι οι κλασικές τεχνικές μετασχηματισμού δεδομένων, όπως η δημιουργία εκθέσεων, τα νέα χαρακτηριστικά που αθροίζουν τα υπάρχοντα και η γενίκευση εννοιών ειδικά σε κατηγορικά χαρακτηριστικά, όπως η αντικατάσταση πλήρων ημερομηνιών στη βάση δεδομένων μόνο με αριθμούς έτους.

3.1.3 Ολοκλήρωση δεδομένων (Data Intergation).

Περιλαμβάνει τη συγχώνευση δεδομένων από πολλαπλές αποθήκες δεδομένων. Η διαδικασία αυτή πρέπει να εκτελείται προσεκτικά, ώστε να αποφεύγονται πλεονασμοί και ασυνέπειες στο σύνολο δεδομένων που προκύπτει. Τυπικές λειτουργίες που πραγματοποιούνται στο πλαίσιο της ενοποίησης δεδομένων είναι ο προσδιορισμός και η ενοποίηση μεταβλητών και τομέων, η ανάλυση της συσχέτισης χαρακτηριστικών, η αντιγραφή πλειάδων και η ανίχνευση συγκρούσεων στις τιμές δεδομένων διαφορετικών πηγών.

3.1.4 Κανονικοποίηση δεδομένων (Data Normalization).

Η χρησιμοποιούμενη μονάδα μέτρησης μπορεί να επηρεάσει την ανάλυση δεδομένων. Όλα τα χαρακτηριστικά θα πρέπει να εκφράζονται στις ίδιες μονάδες μέτρησης και να χρησιμοποιούν κοινή κλίμακα ή εύρος. Η κανονικοποίηση των δεδομένων επιχειρεί να δώσει σε όλα τα χαρακτηριστικά την ίδια βαρύτητα και είναι ιδιαίτερα χρήσιμη στη μέθοδο στατιστικής μάθησης.[24]

3.1.5 Υπολογισμός ελλειπόντων δεδομένων (Missing Data Imputation).

Πρόκειται για μια μορφή καθαρισμού δεδομένων, όπου ο σκοπός είναι να συμπληρωθούν οι μεταβλητές που περιέχουν MVs (Missing Values) με κάποια διαισθητικά δεδομένα. Στις περισσότερες περιπτώσεις, η προσθήκη μιας λογικής εκτίμησης μιας κατάλληλης τιμής δεδομένων είναι καλύτερη από το να την αφήσουμε κενή.

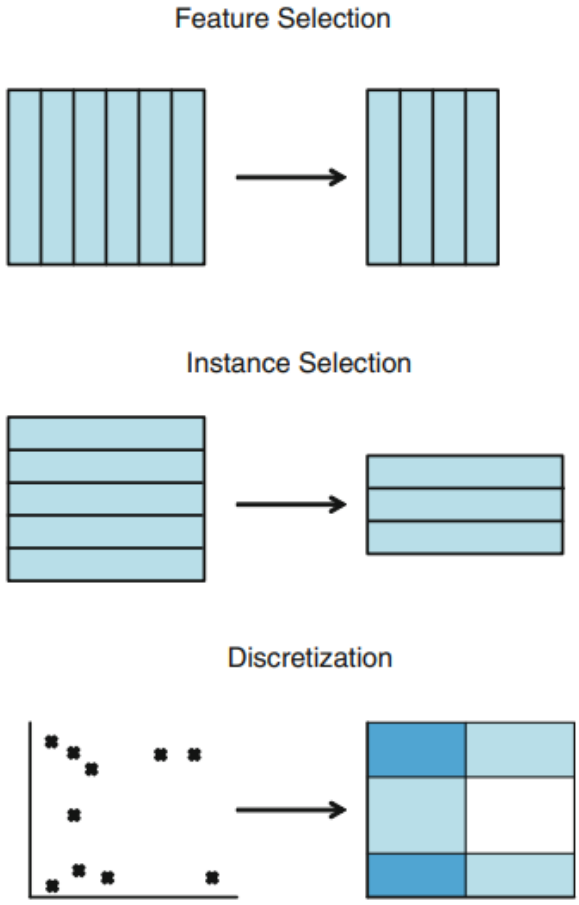
3.1.6 Προσδιορισμός θορύβου (Noise Identification).

Περιλαμβάνεται ως ένα βήμα του καθαρισμού δεδομένων και είναι επίσης γνωστή ως εξομάλυνση στο μετασχηματισμό δεδομένων, με κύριο στόχο την ανίχνευση τυχαίων σφαλμάτων ή αποκλίσεων σε μια μετρούμενη μεταβλητή. Σημειώστε ότι αναφερόμαστε στην ανίχνευση του θορύβου αντί της αφαίρεσης του θορύβου, η οποία σχετίζεται περισσότερο με την εργασία IS στο πλαίσιο της μείωσης δεδομένων. Μόλις ανιχνευθεί ένα θορυβώδες παράδειγμα, μπορούμε να εφαρμόσουμε μια διαδικασία με βάση τη διόρθωση, η οποία θα μπορούσε να περιλαμβάνει κάποιο είδος υποκείμενης λειτουργίας.

3.2 Μείωση δεδομένων (Data Reduction).

Η μείωση δεδομένων περιλαμβάνει το σύνολο των τεχνικών που, με τον ένα ή τον άλλο τρόπο, επιτυγχάνουν μια μειωμένη αναπαράσταση των αρχικών δεδομένων. Για εμάς, η διάκριση των τεχνικών προετοιμασίας δεδομένων είναι εκείνες που απαιτούνται για την κατάλληλη προσαρμογή των δεδομένων ως είσοδο μιας εργασίας DM. Όπως αναφέραμε και προηγουμένως, αυτό σημαίνει ότι αν η προετοιμασία των δεδομένων δεν γίνει σωστά, οι αλγόριθμοι DM δεν θα εκτελούνται ή σίγουρα θα αναφέρουν λανθασμένα αποτελέσματα μετά την εκτέλεσή τους. Στην περίπτωση της μείωσης δεδομένων, τα δεδομένα που παράγονται συνήθως διατηρούν την ουσιαστική δομή και ακεραιότητα των αρχικών δεδομένων, αλλά ο όγκος των δεδομένων μειώνεται. Μπορούν, λοιπόν, τα αρχικά δεδομένα να

χρησιμοποιηθούν, χωρίς την εφαρμογή διαδικασίας μείωσης δεδομένων, ως είσοδος μιας διαδικασίας DM; Η απάντηση είναι ναι, αλλά πρέπει να ληφθούν υπόψη άλλα σημαντικά ζητήματα, τα οποία είναι εξίσου κρίσιμα με τα ζητήματα που αντιμετωπίζονται από την προετοιμασία δεδομένων. Ως εκ τούτου, με μια ματιά, μπορεί να θεωρηθεί ως ένα προαιρετικό βήμα. Ωστόσο, αυτή η διαβεβαίωση μπορεί να είναι αντικρουόμενη. Παρόλο που διατηρείται η ακεραιότητα των δεδομένων, είναι γνωστό ότι κάθε αλγόριθμος έχει ορισμένη χρονική πολυπλοκότητα που εξαρτάται από διάφορες παραμέτρους. Στο DM, μία από αυτές τις παραμέτρους είναι κατά κάποιο τρόπο ευθέως ανάλογη με το μέγεθος της βάσης δεδομένων εισόδου. Εάν το μέγεθος υπερβαίνει το όριο, το οποίο εξαρτάται σε μεγάλο βαθμό από τον τύπο των αλγορίθμων DM, η εκτέλεση του αλγορίθμου μπορεί να είναι απαγορευτική και τότε η εργασία μείωσης των δεδομένων είναι εξίσου κρίσιμη με την προετοιμασία των δεδομένων. Όσον αφορά άλλους παράγοντες, όπως η μείωση της πολυπλοκότητας και η βελτίωση της ποιότητας των μοντέλων που προκύπτουν, ο ρόλος της μείωσης των δεδομένων είναι και πάλι καθοριστικός. [24]



Εικόνα 24. Απεικόνιση της Μείωσης δεδομένων

3.2.1 Επιλογή χαρακτηριστικών (Feature Selection).

Επιτυγχάνει τη μείωση του συνόλου δεδομένων με την αφαίρεση άσχετων ή περιττών χαρακτηριστικών (ή διαστάσεων). Ο στόχος της FS είναι να βρεθεί ένα ελάχιστο σύνολο χαρακτηριστικών, έτσι ώστε η κατανομή πιθανότητας των χαρακτηριστικών εξόδου των δεδομένων (ή κλάσεων) που προκύπτει να είναι όσο το δυνατόν πιο κοντά στην αρχική κατανομή που προκύπτει με τη χρήση όλων των χαρακτηριστικών. Διευκολύνει την κατανόηση του προτύπου που εξάγεται και αυξάνει την ταχύτητα του σταδίου μάθησης.[24]

3.2.2 Επιλογή παραδείγματος (Instance Selection).

Συνίσταται στην επιλογή ενός υποσυνόλου από το σύνολο των διαθέσιμων δεδομένων για την επίτευξη του αρχικού σκοπού της εφαρμογής DM, σαν να είχαν χρησιμοποιηθεί όλα τα δεδομένα. Αποτελεί την οικογένεια προσανατολισμένων μεθόδων που εκτελούν με κάπως ευφυή τρόπο την επιλογή του καλύτερου δυνατού υποσυνόλου παραδειγμάτων από τα αρχικά δεδομένα με τη χρήση ορισμένων κανόνων ή/και ευρετικών μεθόδων. Η τυχαία επιλογή παραδειγμάτων είναι συνήθως γνωστή ως δειγματοληψία και υπάρχει σε πολύ μεγάλο αριθμό μοντέλων DM για τη διεξαγωγή εσωτερικής επικύρωσης και για την αποφυγή υπερβολικής προσαρμογής.[24]

3.2.3 Διακριτοποίηση (Discretization).

Η διαδικασία αυτή μετατρέπει τα ποσοτικά δεδομένα σε ποιοτικά δεδομένα, δηλαδή τα αριθμητικά χαρακτηριστικά σε διακριτά ή ονομαστικά χαρακτηριστικά με πεπερασμένο αριθμό διαστημάτων, επιτυγχάνοντας μια μη επικαλυπτόμενη διαμέριση ενός συνεχούς τομέα. Στη συνέχεια, δημιουργείται μια συσχέτιση μεταξύ κάθε διαστήματος με μια αριθμητική διακριτή τιμή. Μόλις πραγματοποιηθεί η διακριτοποίηση, τα δεδομένα μπορούν να αντιμετωπιστούν ως ονομαστικά δεδομένα κατά τη διάρκεια οποιασδήποτε διαδικασίας DM. Αξίζει να σημειωθεί ότι η διακριτοποίηση είναι στην πραγματικότητα είναι μια υβριδική τεχνική προεπεξεργασίας δεδομένων που περιλαμβάνει τόσο εργασίες προετοιμασίας δεδομένων όσο και εργασίες μείωσης δεδομένων. Ορισμένες πηγές περιλαμβάνουν τη διακριτοποίηση στην κατηγορία μετασχηματισμού δεδομένων και άλλες πηγές

θεωρούν ότι πρόκειται για διαδικασία μείωσης δεδομένων. Στην πράξη, η διακριτοποίηση μπορεί να θεωρηθεί ως μέθοδος μείωσης δεδομένων, δεδομένου ότι αντιστοιχίζει δεδομένα από ένα τεράστιο φάσμα αριθμητικών τιμών σε ένα σημαντικά μειωμένο υποσύνολο διακριτών τιμών. Η απόφασή μας είναι να τη συμπεριλάβουμε κυρίως στη μείωση δεδομένων αν και συμφωνούμε και με την άλλη τάση. Το κίνητρο πίσω από αυτό είναι ότι τα πρόσφατα συστήματα διακριτοποίησης προσπαθούν να μειώσουν τον αριθμό των διακριτών διαστημάτων όσο το δυνατόν περισσότερο, διατηρώντας παράλληλα την απόδοση της περαιτέρω διαδικασίας DM. Με άλλα λόγια, είναι συχνά πολύ εύκολο να εκτελεστεί βασική διακριτοποίηση με οποιονδήποτε τύπο δεδομένων, δεδομένου ότι τα δεδομένα είναι κατάλληλα για έναν συγκεκριμένο αλγόριθμο με έναν απλό χάρτη μεταξύ συνεχών και κατηγορικών τιμών. Ωστόσο, η πραγματική δυσκολία είναι να επιτευχθεί καλή μείωση χωρίς να διακυβευτεί η ποιότητα των δεδομένων, και μεγάλο μέρος της προσπάθειας που καταβάλλουν οι ερευνητές ακολουθεί αυτή την τάση.[24]

3.2.4 Εξόρυξη χαρακτηριστικών/παραγωγή περιστατικών (Feature Extraction/Instance Generation).

Επεκτείνει τόσο το χαρακτηριστικό όσο και το IS επιτρέποντας την τροποποίηση των εσωτερικών τιμών που αντιπροσωπεύουν κάθε παράδειγμα ή χαρακτηριστικό. Στην εξαγωγή χαρακτηριστικών, εκτός από την πράξη αφαίρεσης χαρακτηριστικών, υποσύνολα χαρακτηριστικών μπορούν να συγχωνευθούν ή να συμβάλουν στη δημιουργία τεχνητών υποκατάστατων χαρακτηριστικών. Όσον αφορά τη δημιουργία παραδειγμάτων, η διαδικασία είναι παρόμοια με την έννοια των παραδειγμάτων. Επιτρέπει τη δημιουργία ή την προσαρμογή τεχνητών υποκατάστατων παραδειγμάτων που θα μπορούσαν να αντιπροσωπεύουν καλύτερα τα όρια απόφασης στην επιβλεπόμενη μάθηση.[24]

Κεφάλαιο 4 : Αποτελέσματα εφαρμογής αλγόριθμου ομαδοποίησης με διαφορετικές μετρικές.

Στόχος της εργασίας είναι να χρησιμοποιηθεί ένα σύνολο δεδομένων, το οποίο περιέχει χρήστες, ταινίες καθώς και την αξιολόγηση που έχουν κάνει οι χρήστες αυτοί και εφαρμόζοντας έναν αλγόριθμο ομαδοποίησης μαζί με διάφορες μετρικές μετά τον αλγόριθμο, ψάχνοντας να βρούμε την συσχέτιση μεταξύ τους. Με σκοπό σε θεωρητικό επίπεδο να προτείνουμε αντικείμενα στους χρήστες με παρόμοιο γούστο.

Η εργασία μας ξεκινά φορτώνοντας το σύνολο δεδομένων μας “Dataset.npy”, αρχείο σε NumPy μορφή, το οποίο περιέχει ένα μονοδιάστατο πίνακα (1D array) και φορτώνεται με την np.load .

```
file_path = 'Dataset.npy'
loaded_array = np.load(file_path)
```

Στην συνέχεια κάθε στοιχείο της λίστας διαχωρίζεται στα σημεία όπου υπάρχει κόμμα, δημιουργώντας ένα νέο array το split_elements.

```
split_elements = np.array([element.split(',') for element in loaded_array], dtype=np.string_)
```

Για καλύτερη και ευκολότερη επεξεργασία στα δεδομένα μας δημιουργούμε ένα pandas DataFrame από το Numpy αρχείο με τις εξής στήλες :

- users
- items
- rating
- date

Και η στήλη που περιέχει τις βαθμολογίες των χρηστών (rating) την μετατρέπουμε από string σε αριθμητικό τύπο χρησιμοποιώντας την συνάρτηση “pd.to_numeric”.

```
df = pd.DataFrame(split_elements, columns=['users', 'items', 'rating', 'date'])

# Convert the 'rating' column to numeric
df['rating'] = pd.to_numeric(df['rating'])
```

Ομαδοποιούμε τους χρήστες και μετράμε πόσα αντικείμενα έχουν βαθμολογηθεί από τον κάθε χρήστη. Εφαρμόζουμε ένα φιλτράρισμα στο σύνολο δεδομένων μας και κρατάμε τους χρήστες οι οποίοι έχουν αξιολογήσει περισσότερα από εκατό και λιγότερα από πεντακόσια αντικείμενα αποθηκεύοντας τα στο filtered_users. Και συνεχίζουμε συγχωνεύοντας του φιλτραρισμένους χρήστες στο αρχικό DataFrame ώστε να έχουμε όλα μας τα δεδομένα. Ακόμα γίνεται ένας υπολογισμός των μοναδικών χρηστών και αντικειμένων χρησιμοποιώντας την μέθοδο nunique().

```python
# Group by 'users' and count the number of items rated by each user
user_counts = df.groupby('users').size().reset_index(name='num_items_rated')

# Filter users who have rated more than 100 items and less than 500 items
filtered_users = user_counts[(user_counts['num_items_rated'] > 100) & (user_counts['num_items_rated'] < 500)]

# Merge the filtered users with the original DataFrame to get the data for those users
Newfiltered_df = pd.merge(df, filtered_users[['users']], on='users', how='inner')

# Now filtered_df contains the data for users who have rated > 100 items and < 500
print(Newfiltered_df)

# Print the number of unique users and unique items
num_unique_users = Newfiltered_df['users'].nunique()
num_unique_items = Newfiltered_df['items'].nunique()

print(f'Number of unique users: {num_unique_users}')
print(f'Number of unique items: {num_unique_items}')
```

Αποτέλεσμα:

```
                users         items  rating                date
0         b'ur4111911'  b'tt0074486'      10  b'16 January 2005'
1         b'ur4111911'  b'tt0082971'      10  b'16 January 2005'
2         b'ur4111911'  b'tt0099685'      10  b'16 January 2005'
3         b'ur4111911'  b'tt0114369'      10  b'17 January 2005'
4         b'ur4111911'  b'tt0119488'      10  b'17 January 2005'
...               ...           ...     ...                 ...
508569  b'ur125165080'  b'tt3025994'       8  b'1 December 2020'
508570  b'ur125165080'  b'tt2999390'       7  b'2 December 2020'
508571  b'ur125165080'  b'tt2617456'       8  b'2 December 2020'
508572  b'ur125165080'  b'tt2194499'       9  b'3 December 2020'
508573  b'ur125165080'  b'tt0790636'       9  b'3 December 2020'

[508574 rows x 4 columns]
Number of unique users: 2621
Number of unique items: 111608
```

Εδώ γίνεται διαχείριση διπλότυπων ζευγαριών (χρήστη – αντικειμένου).

```
# Handle duplicate user-item pairs by averaging their ratings
Newfiltered_df = Newfiltered_df.groupby(['users', 'items'], as_index=False)['rating'].mean()
```

Έτσι ώστε εάν ένας χρήστης έχει βαθμολογήσει το ίδιο αντικείμενο πάνω από μια φορά ο κώδικα υπολογίζει ένα μέσο όρο τον βαθμολογιών του, με σκοπό να ελαχιστοποιηθούν οι επαναλαμβανόμενες καταχωρήσεις που μπορεί να επηρεάσουν τα αποτελέσματα των συστάσεων.

Στο παρακάτω κομμάτι κώδικα έχουμε την δημιουργία ενός δισδιάστατου πίνακα (pivot table) ή αλλιώς μιας μήτρας, η οποία έχει ως γραμμές τους χρήστες και ως στήλες τα αντικείμενα με ενδιάμεσες τιμές της βαθμολογίες που έχει βάλει κάθε χρήστης για κάθε αντικείμενο. Οι κενές τιμές (NaN) έχουν αντικατασταθεί με την τιμή 0 για να προχωρήσει πιο εύκολα το μοντέλο μας.

```
# Create a pivot table with users as rows and items as columns
user_item_matrix = Newfiltered_df.pivot(index='users', columns='items', values='rating')

# Fill NaN values with 0 (assuming unrated items should be treated as 0)
user_item_matrix = user_item_matrix.fillna(0)

# Now user_item_matrix is the desired 2D matrix
print(user_item_matrix)
```

Αποτέλεσμα:

```
items          b'tt0000001'  b'tt0000003'  b'tt0000005'  b'tt0000007'  \
users
b'ur0001220'            0.0           0.0           0.0           0.0
b'ur0002746'            0.0           0.0           0.0           0.0
b'ur0003696'            0.0           0.0           0.0           0.0
b'ur0004646'            0.0           0.0           0.0           0.0
b'ur0007613'            0.0           0.0           0.0           0.0
...                     ...           ...           ...           ...
b'ur99519886'           0.0           0.0           0.0           0.0
b'ur9972457'            0.0           0.0           0.0           0.0
b'ur99782462'           0.0           0.0           0.0           0.0
b'ur99809306'           0.0           0.0           0.0           0.0
b'ur99964320'           0.0           0.0           0.0           0.0

items          b'tt0000008'  b'tt0000010'  b'tt0000012'  b'tt0000013'  \
users
b'ur0001220'            0.0           0.0           0.0           0.0
b'ur0002746'            0.0           0.0           0.0           0.0
b'ur0003696'            0.0           0.0           0.0           0.0
b'ur0004646'            0.0           0.0           0.0           0.0
b'ur0007613'            0.0           0.0           0.0           0.0
...                     ...           ...           ...           ...
b'ur99519886'           0.0           0.0           0.0           0.0
b'ur9972457'            0.0           0.0           0.0           0.0
b'ur99782462'           0.0           0.0           0.0           0.0
b'ur99809306'           0.0           0.0           0.0           0.0
b'ur99964320'           0.0           0.0           0.0           0.0

items          b'tt0000014'  b'tt0000015'  ...  b'tt9913022'  b'tt9913036'  \
users                                      ...
b'ur0001220'            0.0           0.0  ...           0.0           0.0
b'ur0002746'            0.0           0.0  ...           0.0           0.0
b'ur0003696'            0.0           0.0  ...           0.0           0.0
b'ur0004646'            0.0           0.0  ...           0.0           0.0
b'ur0007613'            0.0           0.0  ...           0.0           0.0
...                     ...           ...  ...           ...           ...
b'ur99519886'           0.0           0.0  ...           0.0           0.0
b'ur9972457'            0.0           0.0  ...           0.0           0.0
b'ur99782462'           0.0           0.0  ...           0.0           0.0
b'ur99809306'           0.0           0.0  ...           0.0           0.0
b'ur99964320'           0.0           0.0  ...           0.0           0.0
```

items	b'tt9913038'	b'tt9913040'	b'tt9913050'	b'tt9914414'	\
users					
b'ur0001220'	0.0	0.0	0.0	0.0	
b'ur0002746'	0.0	0.0	0.0	0.0	
b'ur0003696'	0.0	0.0	0.0	0.0	
b'ur0004646'	0.0	0.0	0.0	0.0	
b'ur0007613'	0.0	0.0	0.0	0.0	
...	...	...	...	...	
b'ur99519886'	0.0	0.0	0.0	0.0	
b'ur9972457'	0.0	0.0	0.0	0.0	
b'ur99782462'	0.0	0.0	0.0	0.0	
b'ur99809306'	0.0	0.0	0.0	0.0	
b'ur99964320'	0.0	0.0	0.0	0.0	

items	b'tt9914598'	b'tt9915686'	b'tt9916190'	b'tt9916204'
users				
b'ur0001220'	0.0	0.0	0.0	0.0
b'ur0002746'	0.0	0.0	0.0	0.0
b'ur0003696'	0.0	0.0	0.0	0.0
b'ur0004646'	0.0	0.0	0.0	0.0
b'ur0007613'	0.0	0.0	0.0	0.0
...	...	...	...	...
b'ur99519886'	0.0	0.0	0.0	0.0
b'ur9972457'	0.0	0.0	0.0	0.0
b'ur99782462'	0.0	0.0	0.0	0.0
b'ur99809306'	0.0	0.0	0.0	0.0
b'ur99964320'	0.0	0.0	3.0	0.0

[2621 rows x 111608 columns]

Σε αυτό το σημείο εφαρμόζουμε δύο μεθόδους οι οποίες θα μας βοηθήσουν να βρούμε τον βέλτιστο αριθμό των ομάδων (clusters) που χρειαζόμαστε, έτσι ώστε να μπορέσουμε αργότερα να πραγματοποιήσουμε την διαδικασία της ομαδοποίησης (clustering).

Μέθοδος 1^{η} (Elbow Method):

Η μέθοδος του αγκώνα είναι μια οπτική προσέγγιση που χρησιμοποιείται για τον προσδιορισμό του ιδανικού «K» (αριθμός συστάδων) στην ομαδοποίηση K-μέσων. Λειτουργεί με τον υπολογισμό του αθροίσματος τετραγώνων εντός της συστάδας (WCSS), το οποίο είναι το σύνολο των τετραγωνικών αποστάσεων μεταξύ των σημείων δεδομένων και του κέντρου της συστάδας τους. Ωστόσο, υπάρχει ένα σημείο όπου η αύξηση του K δεν οδηγεί πλέον σε σημαντική μείωση του WCSS και ο ρυθμός μείωσης επιβραδύνεται. Το σημείο αυτό αναφέρεται συχνά ως αγκώνας.

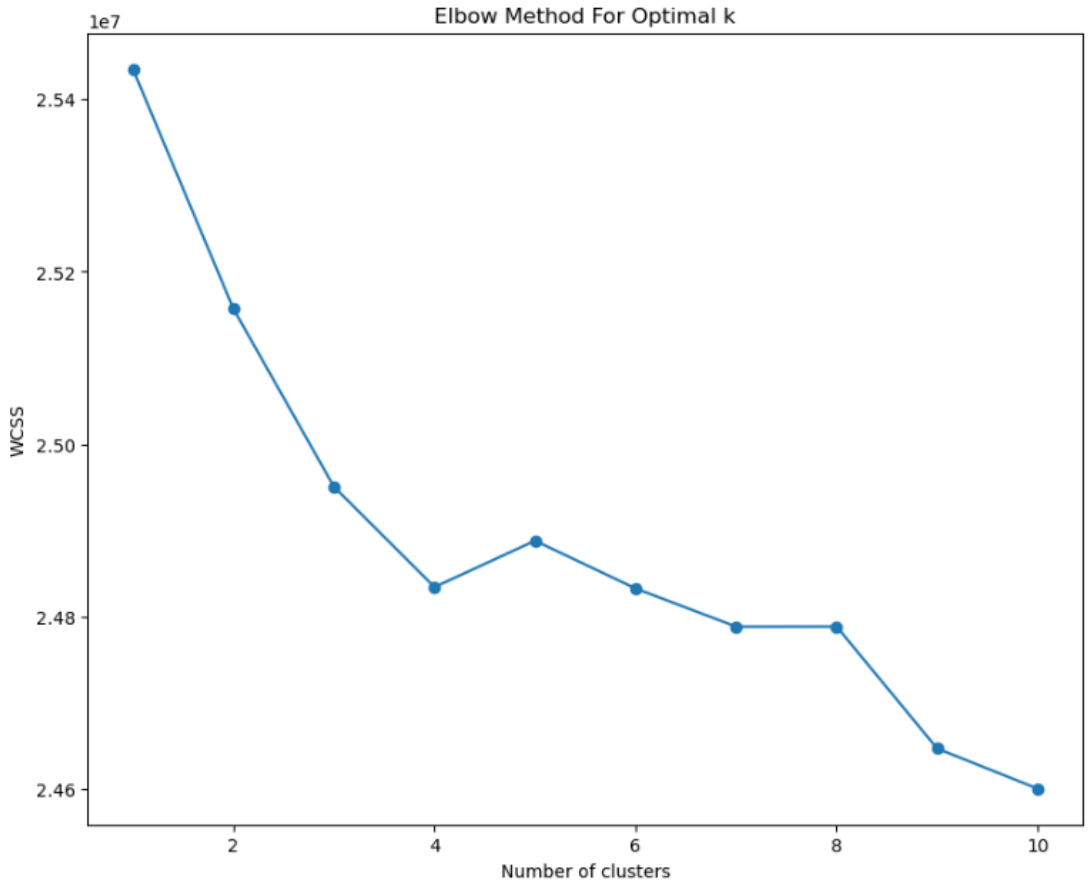
Η μέθοδος του αγκώνα είναι μια απλή αλλά αποτελεσματική τεχνική που χρησιμοποιείται για τον προσδιορισμό του βέλτιστου αριθμού συστάδων (K) σε έναν αλγόριθμο συσταδοποίησης K-Means. Εξετάζει τη σχέση μεταξύ του αριθμού των

συστάδων και του αθροίσματος τετραγώνων εντός της συστάδας (WCSS), ένα μέτρο της διακύμανσης εντός κάθε συστάδας.

Ο κώδικας που χρησιμοποιήθηκε :

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Elbow method for determining the number of clusters
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(user_item_matrix)
    wcss.append(kmeans.inertia_)

# Plot the elbow graph
plt.figure(figsize=(10, 8))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Elbow Method For Optimal k')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

Αποτέλεσμα:



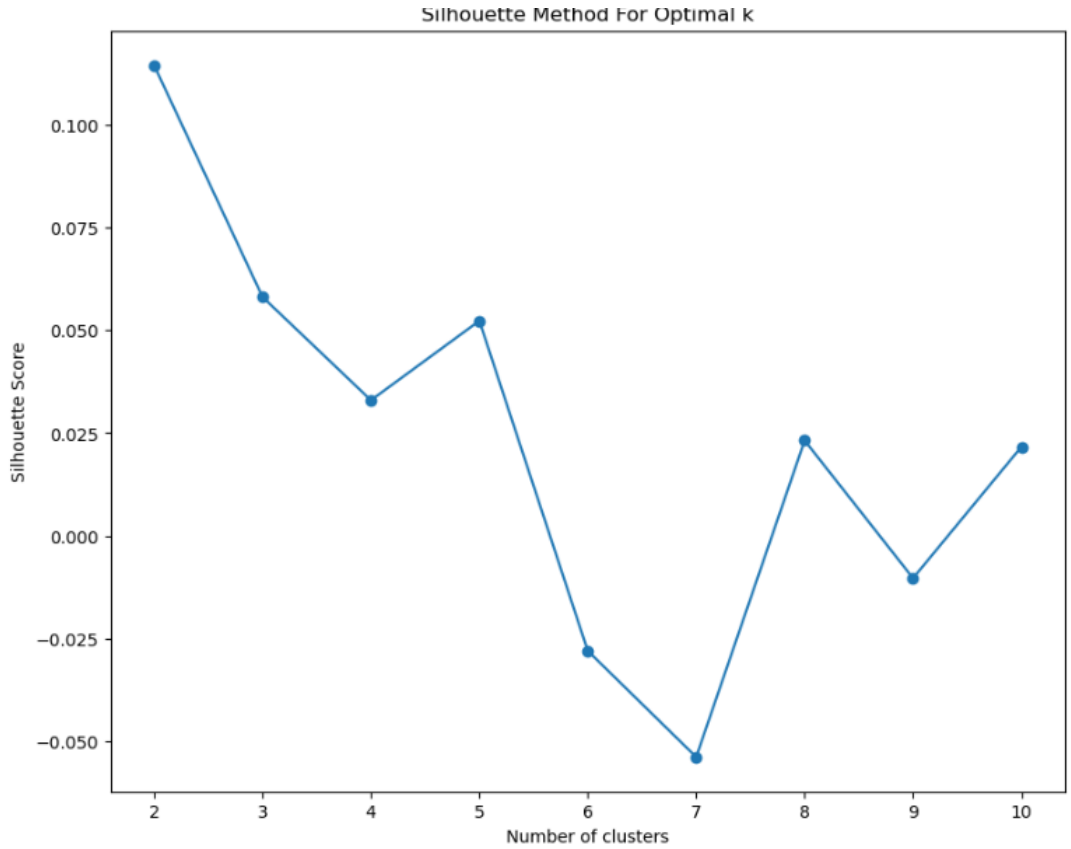
Μέθοδος 2η (Silhouette Score):

Ο συντελεστής σιλουέτας ή ν στην ομαδοποίηση k-means μετρά την ομοιότητα ενός σημείου δεδομένων εντός της συστάδας του (συνοχή) σε σύγκριση με άλλες συστάδες (διαχωρισμός).

Ο κώδικας που χρησιμοποιήθηκε:

```
# Silhouette method for determining the number of clusters
silhouette_scores = []
for i in range(2, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    cluster_labels = kmeans.fit_predict(user_item_matrix)
    silhouette_avg = silhouette_score(user_item_matrix, cluster_labels)
    silhouette_scores.append(silhouette_avg)

# Plot the silhouette scores
plt.figure(figsize=(10, 8))
plt.plot(range(2, 11), silhouette_scores, marker='o')
plt.title('Silhouette Method For Optimal k')
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette Score')
plt.show()
```

Αποτέλεσμα:



Συνοψίζοντας παρατηρούμε ότι ο βέλτιστος αριθμός των clusters για τον αλγόριθμο K-means που θα χρησιμοποιήσουμε είναι οι τέσσερις ομάδες.

Στο κομμάτι που ακολουθεί πραγματοποιούμε μια διαδικασία ομαδοποίησης με την χρήση του αλγορίθμου ομαδοποίησης K-means. Για κάθε σημείο δεδομένων, προσπαθεί να το αναθέσει στο πλησιέστερο κεντροειδές (κέντρο ομάδας). Στη συνέχεια, αναπροσαρμόζει τα κεντροειδή βάσει των σημείων που ανήκουν σε κάθε ομάδα και επαναλαμβάνει αυτή τη διαδικασία μέχρι να σταθεροποιηθούν οι ομάδες.

```
from sklearn.cluster import KMeans

# Assuming user_item_matrix is your DataFrame

# Initialize the KMeans clustering model with 4 clusters
kmeans = KMeans(n_clusters=4, random_state=42)

# Fit the model on the user-item matrix
kmeans.fit(user_item_matrix)

# Get the cluster labels for each user
cluster_labels = kmeans.labels_

# Assign these labels back to your user_item_matrix DataFrame for better insight
user_item_matrix['Cluster'] = cluster_labels

# Print the first few rows to see the clustering result
print(user_item_matrix.head())

# Print the cluster centers
print("Cluster Centers:")
print(kmeans.cluster_centers_)
```

Εισάγουμε τον αλγόριθμο από την βιβλιοθήκη sklearn.cluster, του ζητάμε να χωρίσει τα δεδομένα μας σε τέσσερις ομάδες και με “kmeans.fit(user_item_matrix)” εφαρμόζουμε το μοντέλο του K-means πάνω στα δεδομένα του user_item_matrix που έχουμε αναφέρει παραπάνω. Οι ετικέτες των clusters που επιστράφηκαν από το μοντέλο προστίθενται ως νέα στήλη (Cluster) στο DataFrame user_item_matrix. Τελειώνοντας ζητάμε να μας εκτυπώσει τα κεντροειδής που είναι το μέσο σημείο των δεδομένων που ανήκουν σε μία συγκεκριμένη ομάδα (cluster). Το κεντροειδές είναι ένα διάνυσμα που αντιπροσωπεύει τα χαρακτηριστικά που ορίζουν την "κεντρική" θέση της ομάδας στον χώρο των δεδομένων.

Αποτέλεσμα:

```
Cluster Centers:
[[ 4.33680869e-19 -8.67361738e-19  0.00000000e+00 ...  1.08420217e-19
   0.00000000e+00  1.08420217e-19]
 [ 0.00000000e+00  3.62318841e-02  2.53623188e-02 ...  0.00000000e+00
  -4.33680869e-19  0.00000000e+00]
 [ 4.19947507e-03 -1.56125113e-17  4.19947507e-03 ...  1.04986877e-03
   3.67454068e-03  1.04986877e-03]
 [ 4.33680869e-19  0.00000000e+00  0.00000000e+00 ...  1.08420217e-19
   0.00000000e+00  1.08420217e-19]]
```

items	b'tt0000001'	b'tt0000003'	b'tt0000005'	b'tt0000007'	\
users					
b'ur0001220'	0.0	0.0	0.0	0.0	
b'ur0002746'	0.0	0.0	0.0	0.0	
b'ur0003696'	0.0	0.0	0.0	0.0	
b'ur0004646'	0.0	0.0	0.0	0.0	
b'ur0007613'	0.0	0.0	0.0	0.0	

items	b'tt0000008'	b'tt0000010'	b'tt0000012'	b'tt0000013'	\
users					
b'ur0001220'	0.0	0.0	0.0	0.0	
b'ur0002746'	0.0	0.0	0.0	0.0	
b'ur0003696'	0.0	0.0	0.0	0.0	
b'ur0004646'	0.0	0.0	0.0	0.0	
b'ur0007613'	0.0	0.0	0.0	0.0	

items	b'tt0000014'	b'tt0000015'	...	b'tt9913036'	b'tt9913038'	\
users			...			
b'ur0001220'	0.0	0.0	...	0.0	0.0	
b'ur0002746'	0.0	0.0	...	0.0	0.0	
b'ur0003696'	0.0	0.0	...	0.0	0.0	
b'ur0004646'	0.0	0.0	...	0.0	0.0	
b'ur0007613'	0.0	0.0	...	0.0	0.0	

items	b'tt9913040'	b'tt9913050'	b'tt9914414'	b'tt9914598'	\
users					
b'ur0001220'	0.0	0.0	0.0	0.0	
b'ur0002746'	0.0	0.0	0.0	0.0	
b'ur0003696'	0.0	0.0	0.0	0.0	
b'ur0004646'	0.0	0.0	0.0	0.0	
b'ur0007613'	0.0	0.0	0.0	0.0	

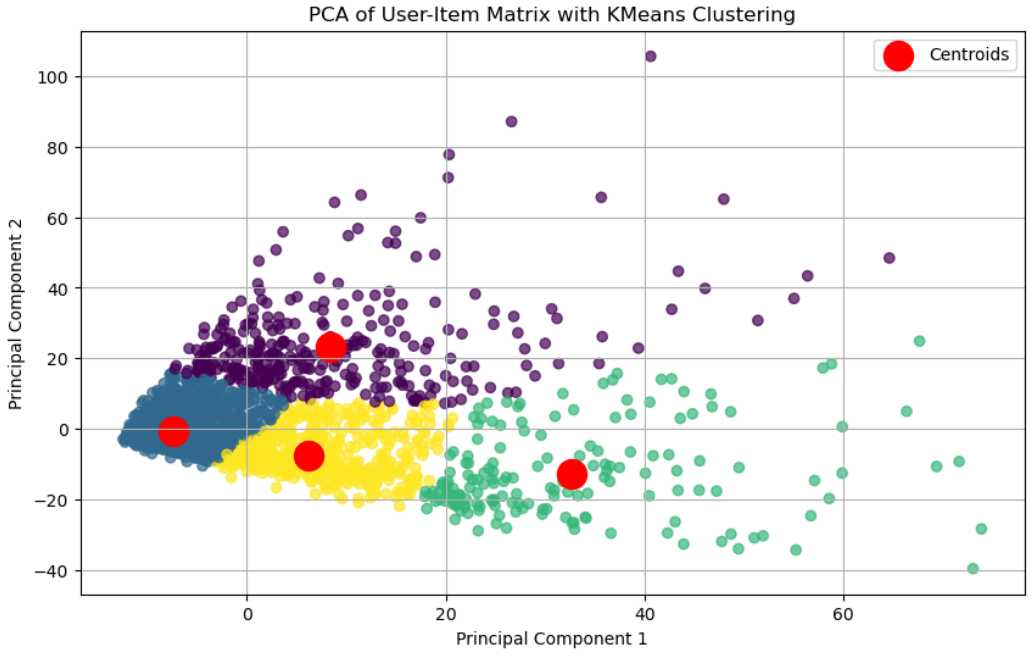
items	b'tt9915686'	b'tt9916190'	b'tt9916204'	Cluster
users				
b'ur0001220'	0.0	0.0	0.0	2
b'ur0002746'	0.0	0.0	0.0	0
b'ur0003696'	0.0	0.0	0.0	2
b'ur0004646'	0.0	0.0	0.0	2
b'ur0007613'	0.0	0.0	0.0	2

Στην συνέχεια θέλουμε να ελαχιστοποιήσουμε και να οπτικοποιήσουμε τα δεδομένα μας εφαρμόζοντας μια PCA (Principal Component Analysis).

```python
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

# Assuming user_item_matrix is your DataFrame

# Initialize the PCA model to reduce dimensions to 2 for visualization
pca = PCA(n_components=2)

# Fit and transform the user-item matrix using PCA
user_item_matrix_pca = pca.fit_transform(user_item_matrix)

# Initialize the KMeans clustering model with 4 clusters
kmeans = KMeans(n_clusters=4, random_state=42)

# Fit the model on the reduced PCA data
kmeans.fit(user_item_matrix_pca)

# Get the cluster labels
labels = kmeans.labels_

# Print the cluster centers (in the reduced PCA space)
print("Cluster Centers (PCA-reduced):")
print(kmeans.cluster_centers_)

# Plot the data points with their corresponding clusters
plt.figure(figsize=(10, 6))
plt.scatter(user_item_matrix_pca[:, 0], user_item_matrix_pca[:, 1], c=labels, cmap='viridis', marker='o', alph
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red', label='Centroids')
plt.title('PCA of User-Item Matrix with KMeans Clustering')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend()
plt.grid(True)
plt.show()
```

Αποτέλεσμα:

```
Cluster Centers (PCA-reduced):
[[  8.43859288  23.25092008]
 [ -7.45929304  -0.64078331]
 [ 32.64855054 -12.66800073]
 [  6.24829955  -7.37526225]]
```



Μετά την ομαδοποίηση μπορούμε να παρατηρήσουμε πως χωρίζονται οι χρήστες και πόσοι εμπεριέχονται στον κάθε έναν από τους clusters.

```
# Count the number of users in each cluster
cluster_counts = user_item_matrix['Cluster'].value_counts()

# Print the counts
print(cluster_counts)
```

Αποτέλεσμα:

```
Cluster
2    1905
0     288
1     276
3     152
Name: count, dtype: int64
```

Στην συνέχεια θα εφαρμόσουμε τρεις μετρικές για να παρατηρήσουμε την συσχέτιση μεταξύ των χρηστών. Οι μετρικές αυτές είναι η Ευκλείδεια Απόσταση τους, το Cosine Similarity και το Pearson Correlation.

Pearson Correlation Coefficient

Ο συντελεστής συσχέτισης Pearson, επίσης γνωστός ως συντελεστής συσχέτισης Pearson ή απλά συντελεστής συσχέτισης, είναι ένα στατιστικό μέτρο που ποσοτικοποιεί τη γραμμική σχέση μεταξύ δύο μεταβλητών. Μετρά πόσο στενά τα σημεία δεδομένων των μεταβλητών ευθυγραμμίζονται σε μια ευθεία γραμμή, υποδεικνύοντας την ισχύ και την κατεύθυνση της σχέσης.

$$r(A,B) = \frac{\sum\big((A-\mu_A)*(B-\mu_B)\big)}{\sqrt{\sum(A-\mu_A)^2}*\sqrt{\sum(B-\mu_B)^2}}$$

Ο συντελεστής συσχέτισης Pearson συμβολίζεται με το σύμβολο "r" και λαμβάνει τιμές μεταξύ -1 και 1. Η τιμή του συντελεστή υποδηλώνει τα εξής:

- r = 1: Τέλεια θετική συσχέτιση. Οι μεταβλητές έχουν ισχυρή θετική γραμμική σχέση, που σημαίνει ότι καθώς αυξάνεται η μία μεταβλητή, αυξάνεται αναλογικά και η άλλη μεταβλητή.
- r = -1: Τέλεια αρνητική συσχέτιση. Οι μεταβλητές έχουν ισχυρή αρνητική γραμμική σχέση, που σημαίνει ότι καθώς αυξάνεται η μία μεταβλητή, η άλλη μεταβλητή μειώνεται αναλογικά.
- r = 0: Δεν υπάρχει γραμμική συσχέτιση. Δεν υπάρχει γραμμική σχέση μεταξύ των μεταβλητών. Είναι ανεξάρτητες μεταξύ τους.

Cluster 0:

Στην γραμμή cluster_0_users = user_item_matrix[user_item_matrix['Cluster'] == 0]: Φιλτράρει τις γραμμές που αντιστοιχούν σε χρήστες στη συστάδα 0.

Στην συνέχεια cluster_0_users = cluster_0_users.drop(columns=['Cluster']) Αποβάλλει τη στήλη "Cluster" για να διασφαλίσει ότι δεν επηρεάζει τον υπολογισμό της συσχέτισης.

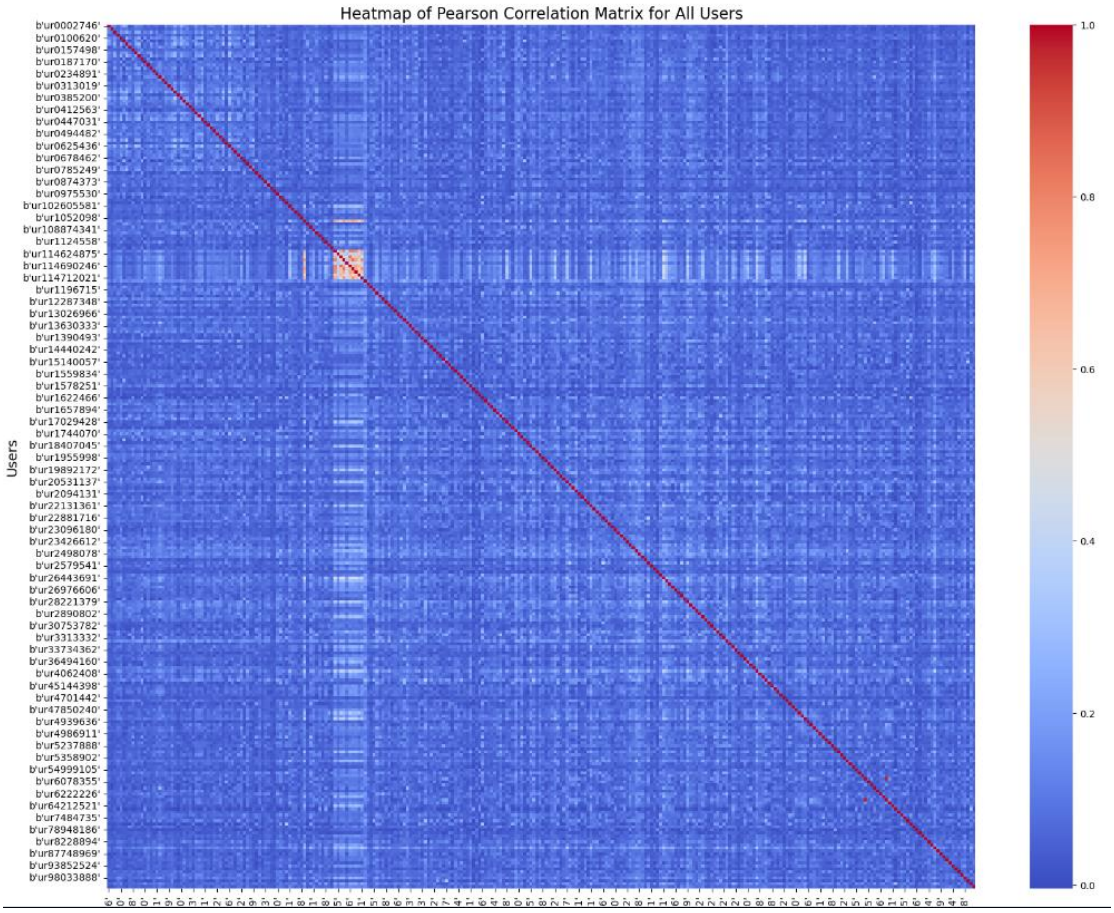
Εδώ correlation_matrix = cluster_0_users.T.corr(). Υπολογίζει τον πίνακα συσχέτισης Pearson για τα φιλτραρισμένα δεδομένα.

```
# Filter the users in cluster 0
cluster_0_users = user_item_matrix[user_item_matrix['Cluster'] == 0]

# Drop the 'Cluster' column as it is not part of the original features
cluster_0_users = cluster_0_users.drop(columns=['Cluster'])

# Calculate the Pearson correlation matrix
correlation_matrix = cluster_0_users.T.corr()

# Print the correlation matrix
print("Pearson Correlation Matrix for Cluster 0:")
print(correlation_matrix)
```

Αποτέλεσμα:

```
Pearson Correlation Matrix for Cluster 0:
users          b'ur0002746'  b'ur0018365'  b'ur0079652'  b'ur0090767'  \
users
b'ur0002746'       1.000000      0.122814      0.140959      0.055226
b'ur0018365'       0.122814      1.000000      0.088941      0.057359
b'ur0079652'       0.140959      0.088941      1.000000      0.070747
b'ur0090767'       0.055226      0.057359      0.070747      1.000000
b'ur0100620'       0.146171      0.106337      0.183535      0.084159
...                     ...           ...           ...           ...
b'ur96416540'      0.047278      0.055071      0.052675      0.068818
b'ur98033888'      0.077375      0.070267      0.066974      0.156095
b'ur98240498'      0.011419      0.017584      0.016792      0.089136
b'ur9972457'       0.057806      0.055975      0.120930      0.076130
b'ur99782462'      0.028836      0.013037      0.037200      0.070987

users          b'ur0100620'  b'ur0110721'  b'ur0111563'  b'ur0140921'  \
users
b'ur0002746'       0.146171      0.241977      0.189991      0.021434
b'ur0018365'       0.106337      0.087674      0.108246      0.052395
b'ur0079652'       0.183535      0.153923      0.249344      0.147000
b'ur0090767'       0.084159      0.043843      0.039660      0.069488
b'ur0100620'       1.000000      0.012360      0.268920      0.132520
...                     ...           ...           ...           ...
b'ur96416540'      0.089493      0.069491      0.060746      0.088930
b'ur98033888'      0.134820      0.074646      0.045752      0.074991
b'ur98240498'      0.059335      0.017021     -0.001692      0.076015
b'ur9972457'       0.141435      0.031950      0.053500      0.057882
b'ur99782462'      0.073987      0.043902      0.013356      0.071519

users          b'ur0157498'  b'ur0166317'  ...  b'ur9150302'  b'ur9187930'  \
users                                      ...
b'ur0002746'       0.064695      0.144806  ...      0.027594      0.043046
b'ur0018365'       0.062581      0.096048  ...      0.052239      0.044473
b'ur0079652'       0.053202      0.228519  ...      0.041817      0.040626
b'ur0090767'       0.045319      0.058416  ...      0.081006      0.049489
b'ur0100620'       0.055009      0.162631  ...      0.046776      0.075912
...                     ...           ...  ...           ...           ...
b'ur96416540'      0.054405      0.043105  ...      0.110599      0.093525
b'ur98033888'      0.069136      0.083583  ...      0.127912      0.075486
b'ur98240498'      0.047432      0.009967  ...      0.072518      0.044688
b'ur9972457'       0.081167      0.059839  ...      0.077811      0.033284
b'ur99782462'      0.048944      0.045951  ...      0.104827      0.120621
```

```
users          b'ur93852524'  b'ur9480570'  b'ur9521536'  b'ur96416540'  \
users
b'ur0002746'       0.042265      0.048599      0.069111       0.047278
b'ur0018365'       0.047773      0.041840      0.040736       0.055071
b'ur0079652'       0.047318      0.021870      0.086573       0.052675
b'ur0090767'       0.115619      0.029615      0.060647       0.068818
b'ur0100620'       0.068864      0.061543      0.044234       0.089493
...                     ...           ...           ...            ...
b'ur96416540'      0.132268      0.094838      0.039854       1.000000
b'ur98033888'      0.189856      0.070971      0.096441       0.124077
b'ur98240498'      0.058879      0.042561      0.020237       0.097455
b'ur9972457'       0.071043      0.039027      0.150820       0.041178
b'ur99782462'      0.083661      0.061246      0.026914       0.097278

users          b'ur98033888'  b'ur98240498'  b'ur9972457'  b'ur99782462'
users
b'ur0002746'       0.077375       0.011419      0.057806       0.028836
b'ur0018365'       0.070267       0.017584      0.055975       0.013037
b'ur0079652'       0.066974       0.016792      0.120930       0.037200
b'ur0090767'       0.156095       0.089136      0.076130       0.070987
b'ur0100620'       0.134820       0.059335      0.141435       0.073987
...                     ...            ...           ...            ...
b'ur96416540'      0.124077       0.097455      0.041178       0.097278
b'ur98033888'      1.000000       0.039648      0.137711       0.113865
b'ur98240498'      0.039648       1.000000      0.022407       0.054428
b'ur9972457'       0.137711       0.022407      1.000000       0.059932
b'ur99782462'      0.113865       0.054428      0.059932       1.000000

[288 rows x 288 columns]
```



Η ίδια διαδικασία συνεχίζεται και για τους υπόλοιπους clusters.

Euclidean Distance

Στο πλαίσιο ενός πίνακα χρήστη-αντικειμένου, η ευκλείδεια απόσταση μετρά την ευθεία απόσταση μεταξύ δύο χρηστών (ή αντικειμένων) σε έναν πολυδιάστατο χώρο, όπου κάθε διάσταση αντιπροσωπεύει ένα διαφορετικό αντικείμενο (ή χρήστη). Για αυτό το σύστημα, η απόσταση υπολογίζεται με βάση τις διαφορές στις αξιολογήσεις ή τις τιμές μεταξύ δύο χρηστών για τα ίδια αντικείμενα.

Cluster 1

Υπολογίζεται η ευκλείδεια απόσταση μεταξύ κάθε ζεύγους χρηστών στη συστάδα 1. Η βιβλιοθήκη scipy.spatial.distance παρέχει μια συνάρτηση pdist που υπολογίζει τις αποστάσεις ανά ζεύγη.

Στην γραμμή euclidean_distances = pdist(cluster_1_users, metric='euclidean') Υπολογίζει τις κατά ζεύγη ευκλείδειες αποστάσεις μεταξύ των γραμμών (χρηστών) του cluster_1_users.

Στην συνέχεια μετατρέπουμε τον συμπυκνωμένο πίνακα αποστάσεων από το pdist σε τετραγωνική μορφή, όπου η απόσταση μεταξύ κάθε ζεύγους χρηστών παρουσιάζεται σε μορφή πίνακα distance_matrix = squareform(euclidean_distances).

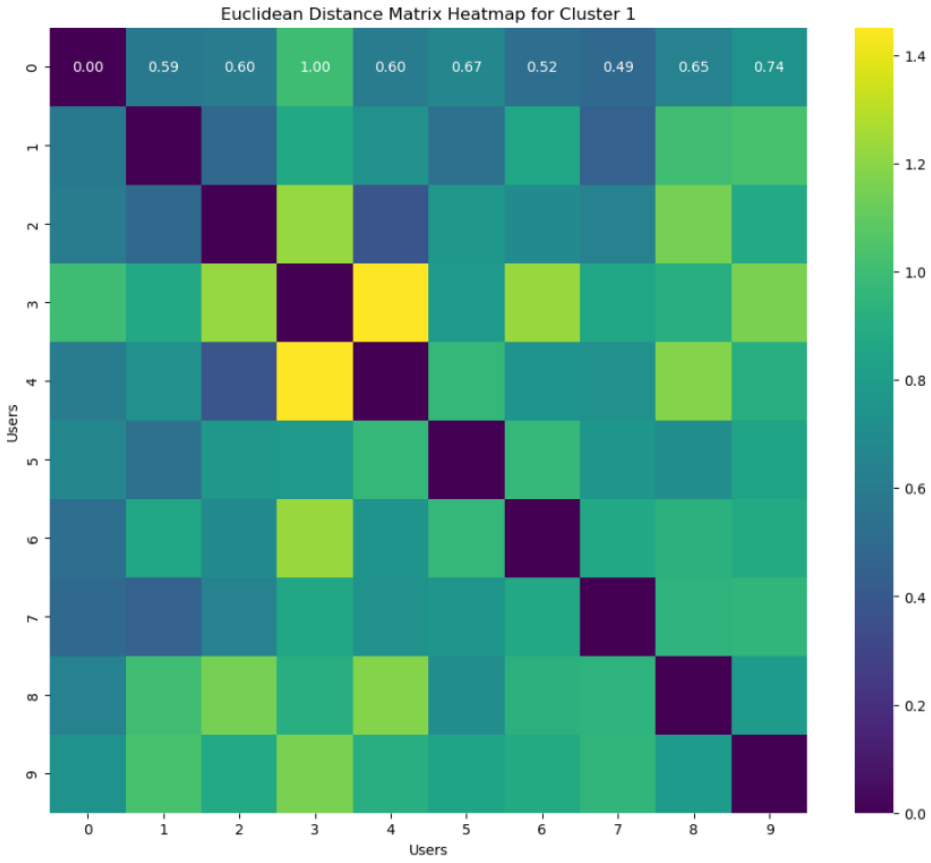
Το distance_df που προκύπτει είναι ένα DataFrame που περιέχει τις κατά ζεύγη ευκλείδειες αποστάσεις, με τις γραμμές και τις στήλες να επισημαίνονται με δείκτες χρήστη distance_df = pd.DataFrame(distance_matrix, index=cluster_1_users.index, columns=cluster_1_users.index).

```
from scipy.spatial.distance import pdist, squareform

# Filter out users that belong to Cluster 1
cluster_1_users = user_item_matrix[user_item_matrix['Cluster'] == 1]

# Drop the 'Cluster' column as it is not needed for distance calculations
cluster_1_users = cluster_1_users.drop('Cluster', axis=1)

# Calculate the pairwise Euclidean distances
euclidean_distances = pdist(cluster_1_users, metric='euclidean')

# Convert the distances to a square matrix form (distance matrix)
distance_matrix = squareform(euclidean_distances)

# Create a DataFrame for better readability
distance_df = pd.DataFrame(distance_matrix, index=cluster_1_users.index, columns=cluster_1_users.index)

# Print the Euclidean distance matrix
print("Euclidean Distance Matrix for Cluster 1:")
print(distance_df)
```

Αποτέλεσμα:

```
Euclidean Distance Matrix for Cluster 1:
users          b'ur0277234'  b'ur0283074'  b'ur0503545'  b'ur0609951'  \
users
b'ur0277234'       0.000000    124.667558    125.502988    143.380612
b'ur0283074'     124.667558      0.000000    120.137421    137.767921
b'ur0503545'     125.502988    120.137421      0.000000    143.181703
b'ur0609951'     143.380612    137.767921    143.181703      0.000000
b'ur0879559'     164.742223    156.869372    163.006135    178.991620
...                     ...           ...           ...           ...
b'ur95614173'    105.773343     99.055540    106.296754    128.023435
b'ur96545050'    126.174482    111.919614    125.495020    137.934767
b'ur96803587'    124.996000    118.042365    121.995902    142.597335
b'ur97849810'    117.639279    103.744879    118.970585    131.799090
b'ur98002597'    145.880088    139.681781    144.499135    161.823978

users          b'ur0879559'  b'ur100147162'  b'ur100248460'  b'ur101128147'  \
users
b'ur0277234'     164.742223      148.899295      114.271606      123.369364
b'ur0283074'     156.869372      139.316187      107.359210      112.374374
b'ur0503545'     163.006135      148.276768      115.295273      123.227432
b'ur0609951'     178.991620      165.339046      136.359818      140.776418
b'ur0879559'       0.000000      175.274071      148.243044      156.108936
...                     ...             ...             ...             ...
b'ur95614173'    145.354738      128.042962       84.628600       93.765665
b'ur96545050'    160.605106      146.051361      107.266024      101.715289
b'ur96803587'    156.511980      136.978100      110.054532      120.166551
b'ur97849810'    156.662057      137.905765      101.936255       94.514549
b'ur98002597'    177.129896      164.067059      129.988461      140.616500

users          b'ur102184407'  b'ur102222939'  ...  b'ur93138442'  \
users                                          ...
b'ur0277234'       135.572859      122.951210  ...     136.378151
b'ur0283074'       132.619757      115.572488  ...     123.971771
b'ur0503545'       137.473634      120.374416  ...     136.132289
b'ur0609951'       157.860698      142.460521  ...     148.811962
b'ur0879559'       164.869645      157.933530  ...     162.692962
...                       ...             ...  ...            ...
b'ur95614173'      115.160757       99.844880  ...     109.421205
b'ur96545050'      135.653972      120.320406  ...     119.377552
b'ur96803587'      133.708638      115.329961  ...     125.733846
b'ur97849810'      129.556937      113.639782  ...     113.639782
b'ur98002597'      154.780490      139.907112  ...     154.427977
```

```
users          b'ur93969415'  b'ur94671559'  b'ur94859844'  b'ur95182476'  \
users
b'ur0277234'      125.598567     126.964562     119.641130     127.263506
b'ur0283074'      118.511603     119.272797     111.534748     123.020324
b'ur0503545'      120.962804     127.526468     117.093979     122.527548
b'ur0609951'      141.127602     145.938343     138.823629     145.835524
b'ur0879559'      158.256122     158.933949     155.370525     161.387732
...                      ...            ...            ...            ...
b'ur95614173'      99.684502     101.044545      96.218501     103.990384
b'ur96545050'     123.065023     116.558998     115.896506     123.983870
b'ur96803587'     114.258479     123.069086     113.030969     120.116610
b'ur97849810'     114.245350     112.120471     110.738431     120.511410
b'ur98002597'     140.925512     144.779142     137.102152     144.322555

users          b'ur95614173'  b'ur96545050'  b'ur96803587'  b'ur97849810'  \
users
b'ur0277234'      105.773343     126.174482     124.996000     117.639279
b'ur0283074'       99.055540     111.919614     118.042365     103.744879
b'ur0503545'      106.296754     125.495020     121.995902     118.970585
b'ur0609951'      128.023435     137.934767     142.597335     131.799090
b'ur0879559'      145.354738     160.605106     156.511980     156.662057
...                      ...            ...            ...            ...
b'ur95614173'       0.000000      93.263069     100.129916      85.621259
b'ur96545050'      93.263069       0.000000     123.588025      90.426766
b'ur96803587'     100.129916     123.588025       0.000000     117.970335
b'ur97849810'      85.621259      90.426766     117.970335       0.000000
b'ur98002597'     125.199840     142.691976     138.942434     139.039563

users          b'ur98002597'
users
b'ur0277234'      145.880088
b'ur0283074'      139.681781
b'ur0503545'      144.499135
b'ur0609951'      161.823978
b'ur0879559'      177.129896
...                      ...
b'ur95614173'     125.199840
b'ur96545050'     142.691976
b'ur96803587'     138.942434
b'ur97849810'     139.039563
b'ur98002597'       0.000000

[276 rows x 276 columns]
```



Cosine Similarity:

Η ομοιότητα συνημίτονου είναι ένα μέτρο που χρησιμοποιείται για τον προσδιορισμό της ομοιότητας μεταξύ δύο μη μηδενικών διανυσμάτων σε έναν διανυσματικό χώρο. Υπολογίζει το συνημίτονο της γωνίας μεταξύ των διανυσμάτων, αντιπροσωπεύοντας τον προσανατολισμό και την ομοιότητά τους.

$$CosineSimilarity(A, B) = \frac{A \cdot B}{\|A\| \, \|B\|}$$

- A - B δηλώνει το τετραγωνικό γινόμενο των διανυσμάτων A και B, το οποίο είναι το άθροισμα του στοιχειομετρικού πολλαπλασιασμού των αντίστοιχων συνιστωσών τους.
- ||A|| αντιπροσωπεύει την ευκλείδεια νόρμα ή το μέγεθος του διανύσματος A, που υπολογίζεται ως η τετραγωνική ρίζα του αθροίσματος των τετραγώνων των συνιστωσών του.
- ||B|| αντιπροσωπεύει την ευκλείδεια νόρμα ή το μέγεθος του διανύσματος B.

Η προκύπτουσα τιμή κυμαίνεται από -1 έως 1, όπου το 1 υποδηλώνει ότι τα διανύσματα έχουν την ίδια κατεύθυνση (δηλ. είναι εντελώς παρόμοια), το -1 υποδηλώνει ότι έχουν αντίθετες κατευθύνσεις (δηλ. είναι εντελώς ανόμοια) και το 0 υποδηλώνει ότι είναι ορθογώνια ή ανεξάρτητα (δηλ. δεν υπάρχει ομοιότητα). Είναι ιδιαίτερα χρήσιμο σε σενάρια όπου το μέγεθος των διανυσμάτων δεν είναι σημαντικό και η εστίαση είναι στην κατεύθυνση ή τον σχετικό προσανατολισμό των διανυσμάτων.

Cluster 2

Στο cosine_sim = cosine_similarity(cluster_2_users) υπολογίζεται η ομοιότητα του συνημίτονου μεταξύ των γραμμών (χρηστών) του cluster_2_users.

Και στο cosine_sim_df = pd.DataFrame(cosine_sim, index=cluster_2_users.index, columns=cluster_2_users.index) το cosine_sim_df που προκύπτει είναι ένα πλαίσιο δεδομένων που περιέχει τις βαθμολογίες ομοιότητας συνημίτονου, με γραμμές και στήλες επισημασμένες με δείκτες χρηστών.

```
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity

# Filter out users that belong to Cluster 1
cluster_2_users = user_item_matrix[user_item_matrix['Cluster'] == 2]

# Drop the 'Cluster' column as it is not needed for similarity calculations
cluster_2_users = cluster_2_users.drop('Cluster', axis=1)

# Calculate the cosine similarity
cosine_sim = cosine_similarity(cluster_2_users)

# Create a DataFrame for better readability
cosine_sim_df = pd.DataFrame(cosine_sim, index=cluster_2_users.index, columns=cluster_2_users.index)

# Print the cosine similarity matrix
print("Cosine Similarity Matrix for Cluster 2:")
print(cosine_sim_df)
```

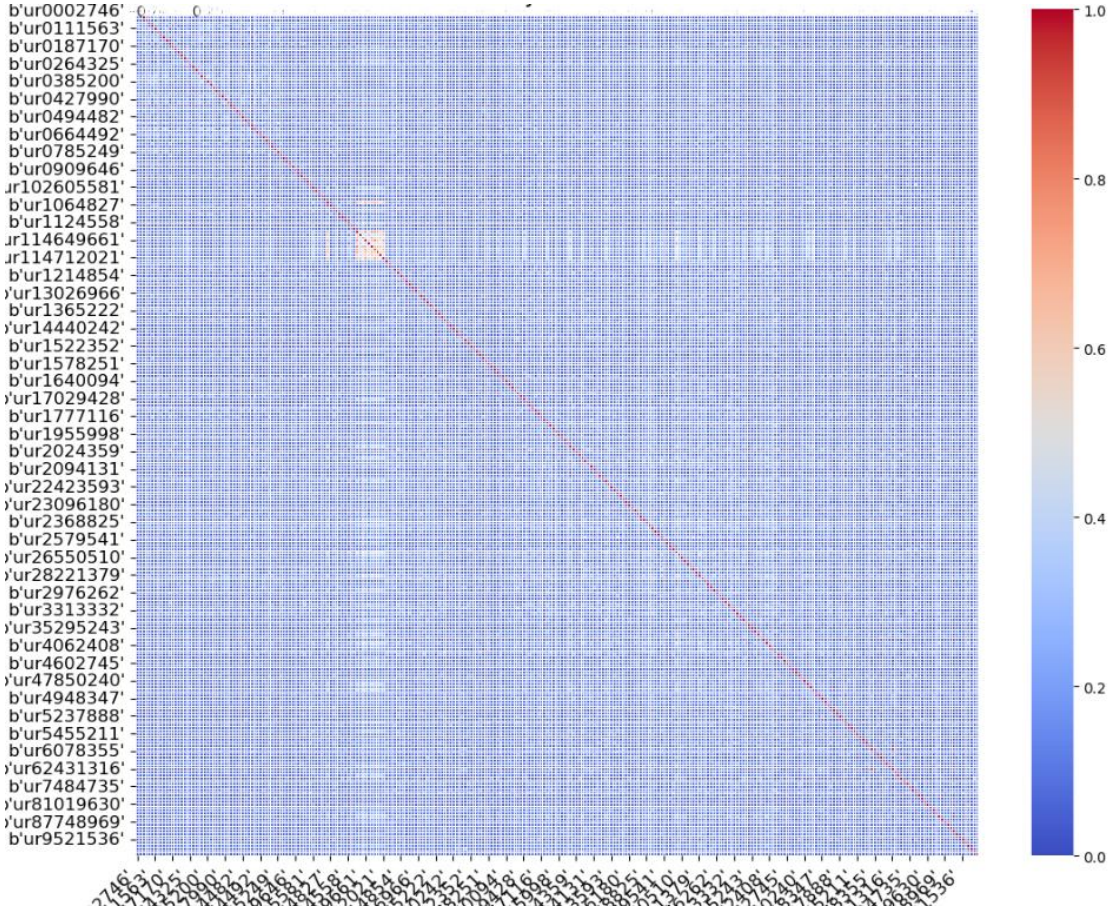
Αποτέλεσμα:

```
Cosine Similarity Matrix for Cluster 2:
users          b'ur0001220'  b'ur0003696'  b'ur0004646'  b'ur0007613'  \
users
b'ur0001220'       1.000000      0.003759      0.000000      0.043697
b'ur0003696'       0.003759      1.000000      0.000000      0.000000
b'ur0004646'       0.000000      0.000000      1.000000      0.000000
b'ur0007613'       0.043697      0.000000      0.000000      1.000000
b'ur0009605'       0.003233      0.031724      0.013051      0.007925
...                     ...           ...           ...           ...
b'ur98946302'      0.010902      0.009737      0.000000      0.000000
b'ur99310909'      0.011838      0.000000      0.000000      0.033674
b'ur99519886'      0.024106      0.000000      0.012263      0.028086
b'ur99809306'      0.005582      0.000000      0.000000      0.017214
b'ur99964320'      0.010551      0.000000      0.000000      0.013580

users          b'ur0009605'  b'ur0010198'  b'ur0011596'  b'ur0019286'  \
users
b'ur0001220'       0.003233      0.031419      0.025252      0.017521
b'ur0003696'       0.031724      0.056444      0.059038      0.036659
b'ur0004646'       0.013051      0.009594      0.023108      0.000000
b'ur0007613'       0.007925      0.021139      0.006359      0.014395
b'ur0009605'       1.000000      0.055179      0.081516      0.014224
...                     ...           ...           ...           ...
b'ur98946302'      0.029483      0.016847      0.000000      0.019919
b'ur99310909'      0.000000      0.000889      0.000000      0.005223
b'ur99519886'      0.007378      0.012675      0.014849      0.030881
b'ur99809306'      0.000000      0.005198      0.000000      0.000000
b'ur99964320'      0.000000      0.000000      0.010358      0.000000

users          b'ur0020866'  b'ur0023796'  ...  b'ur9830389'  b'ur98341434'  \
users                                      ...
b'ur0001220'       0.015722      0.025051  ...      0.007553       0.006302
b'ur0003696'       0.015943      0.038046  ...      0.000000       0.003953
b'ur0004646'       0.015765      0.009148  ...      0.009421       0.000000
b'ur0007613'       0.035758      0.059039  ...      0.013460       0.007886
b'ur0009605'       0.031292      0.000000  ...      0.007165       0.031168
...                     ...           ...  ...           ...            ...
b'ur98946302'      0.018930      0.014634  ...      0.007257       0.023444
b'ur99310909'      0.013464      0.022632  ...      0.000000       0.003531
b'ur99519886'      0.012233      0.041963  ...      0.000000       0.090731
b'ur99809306'      0.006271      0.000000  ...      0.000000       0.014679
b'ur99964320'      0.000000      0.006899  ...      0.000000       0.008092
```

users	b'ur98435364'	b'ur98571307'	b'ur98727037'	b'ur98946302'	\
users					
b'ur0001220'	0.027598	0.014054	0.005496	0.010902	
b'ur0003696'	0.008803	0.015413	0.000000	0.009737	
b'ur0004646'	0.019504	0.001967	0.001652	0.000000	
b'ur0007613'	0.031306	0.020951	0.009751	0.000000	
b'ur0009605'	0.000000	0.000000	0.039589	0.029483	
...	...	...	...	...	
b'ur98946302'	0.011848	0.023434	0.008556	1.000000	
b'ur99310909'	0.052338	0.044345	0.038015	0.013530	
b'ur99519886'	0.066701	0.073703	0.005537	0.000000	
b'ur99809306'	0.008580	0.013777	0.006488	0.019852	
b'ur99964320'	0.048567	0.051932	0.015478	0.000000	

users	b'ur99310909'	b'ur99519886'	b'ur99809306'	b'ur99964320'
users				
b'ur0001220'	0.011838	0.024106	0.005582	0.010551
b'ur0003696'	0.000000	0.000000	0.000000	0.000000
b'ur0004646'	0.000000	0.012263	0.000000	0.000000
b'ur0007613'	0.033674	0.028086	0.017214	0.013580
b'ur0009605'	0.000000	0.007378	0.000000	0.000000
...	...	...	...	...
b'ur98946302'	0.013530	0.000000	0.019852	0.000000
b'ur99310909'	1.000000	0.036670	0.042405	0.034819
b'ur99519886'	0.036670	1.000000	0.007707	0.031677
b'ur99809306'	0.042405	0.007707	1.000000	0.017408
b'ur99964320'	0.034819	0.031677	0.017408	1.000000

[1905 rows x 1905 columns]

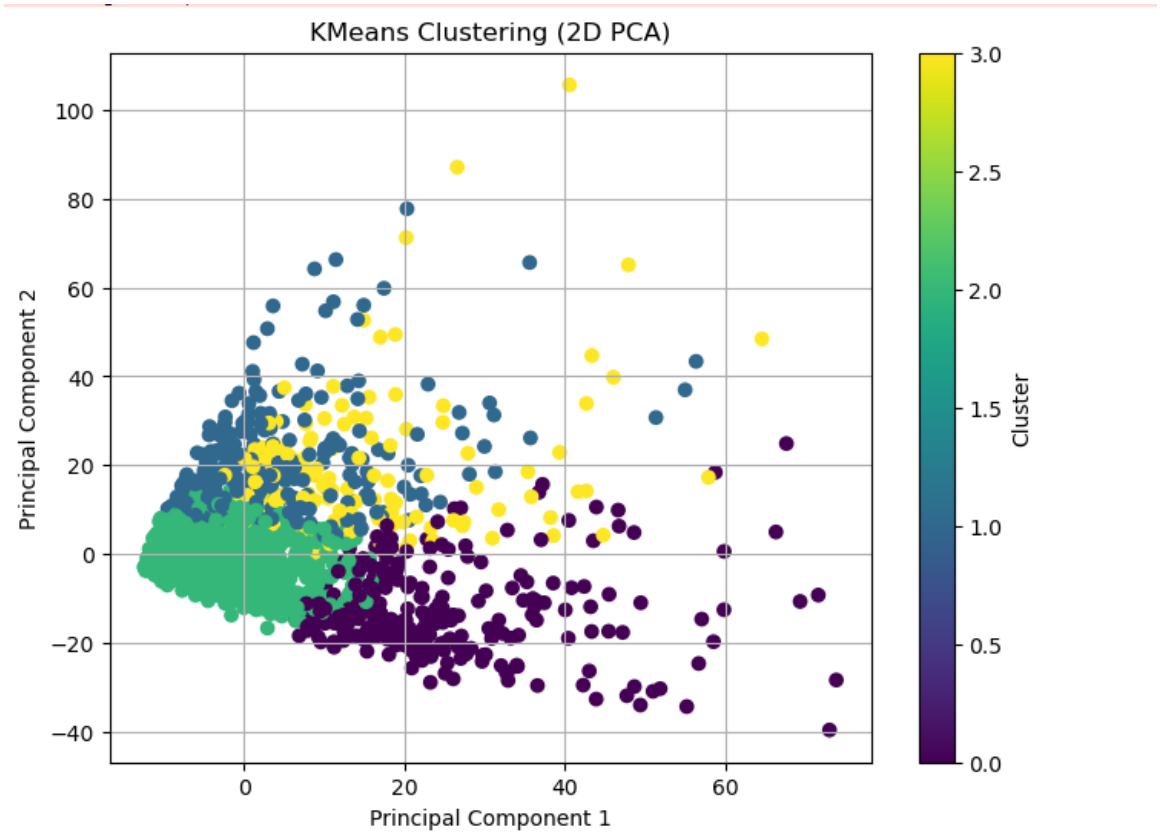

b'ur0002746'
b'ur0111563'
b'ur0187170'
b'ur0264325'
b'ur0385200'
b'ur0427990'
b'ur0494482'
b'ur0664492'
b'ur0785249'
b'ur0909646'
ur102605581'
b'ur1064827'
b'ur1124558'
ur114649661'
ur114712021'
b'ur1214854'
'ur13026966'
b'ur1365222'
'ur14440242'
b'ur1522352'
b'ur1578251'
b'ur1640094'
'ur17029428'
b'ur1777116'
b'ur1955998'
b'ur2024359'
b'ur2094131'
'ur22423593'
'ur23096180'
b'ur2368825'
b'ur2579541'
'ur26550510'
'ur28221379'
b'ur2976262'
b'ur3313332'
'ur35295243'
b'ur4062408'
b'ur4602745'
'ur47850240'
b'ur4948347'
b'ur5237888'
b'ur5455211'
b'ur6078355'
'ur62431316'
b'ur7484735'
'ur81019630'
'ur87748969'
b'ur9521536'
1.0
0.8
0.6
0.4
0.2
0.0

Κεφάλαιο 5 : Κατασκευή Γραφημάτων Συστάδων

Το παρακάτω γγράφημα είναι χρήσιμο για την οπτικοποίηση του αποτελέσματος της ομαδοποίησης, βοηθώντας στην κατανόηση της δομής και της κατανομής των δεδομένων στις διαφορετικές ομάδες (clusters).

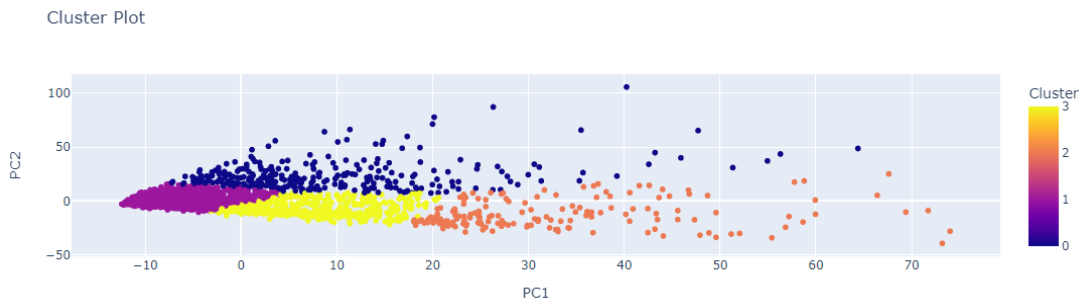
Οι δύο άξονες αντιπροσωπεύουν τις δύο κύριες συνιστώσες που προέκυψαν από την ανάλυση PCA, οι οποίες αιχμαλωτίζουν τη μέγιστη δυνατή διασπορά των δεδομένων. Με τα δεδομένα να έχουν ομαδοποιηθεί σε τέσσερα διαφορετικά clusters, τα οποία διακρίνονται με διαφορετικά χρώματα. Το χρώμα κάθε σημείου δείχνει σε ποιο cluster ανήκει. Τέλος τα σημεία που βρίσκονται πιο κοντά μεταξύ τους και έχουν το ίδιο χρώμα ανήκουν στον ίδιο cluster, υποδεικνύοντας ότι έχουν παρόμοια χαρακτηριστικά σύμφωνα με τα δεδομένα και τον αλγόριθμο K-means.

```
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

# Assuming user_item_matrix is your DataFrame

# Initialize the KMeans clustering model with 4 clusters
kmeans = KMeans(n_clusters=4, random_state=42)

# Fit the model on the user-item matrix
kmeans.fit(user_item_matrix)

# Get the cluster labels for each user
cluster_labels = kmeans.labels_

# Assign these labels back to your user_item_matrix DataFrame for better insight
user_item_matrix['Cluster'] = cluster_labels

# Reduce dimensionality of the data to 2D for visualization using PCA
pca = PCA(n_components=2)
user_item_matrix_2d = pca.fit_transform(user_item_matrix.drop('Cluster', axis=1))

# Plot the clusters
plt.figure(figsize=(8, 6))

# Scatter plot for each cluster
plt.scatter(user_item_matrix_2d[:, 0], user_item_matrix_2d[:, 1], c=cluster_labels, cmap='viridis')

# Annotate the plot
plt.title('KMeans Clustering (2D PCA)')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.colorbar(label='Cluster')
plt.grid(True)

# Show plot
plt.show()
```



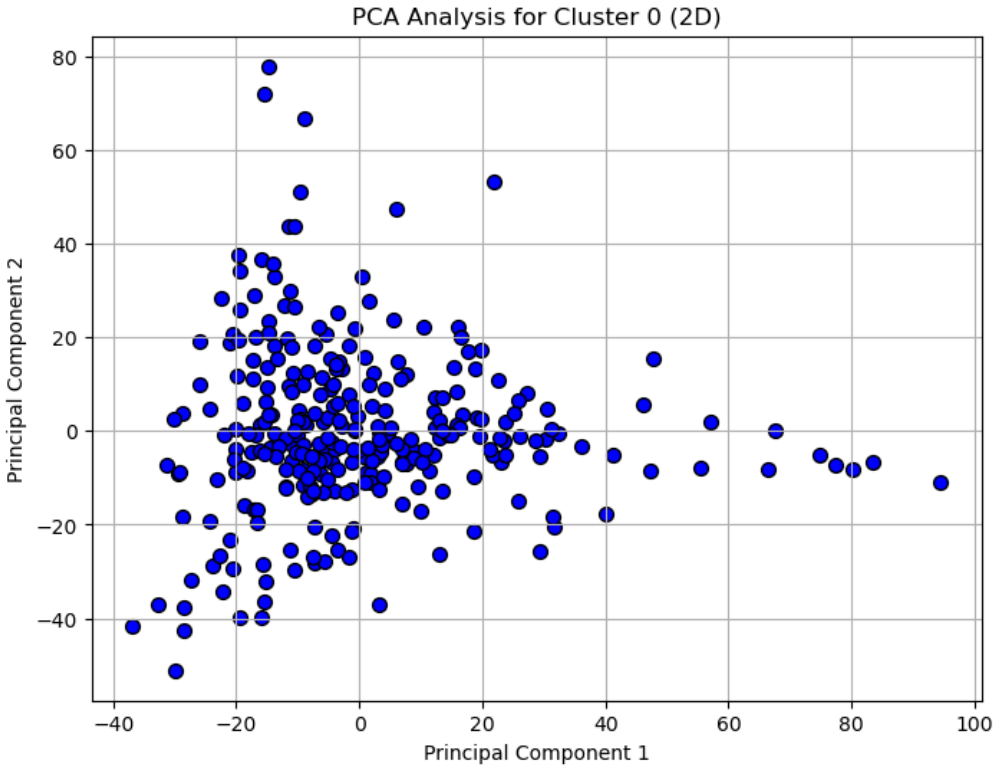
Ένα ακόμη γράφημα για τους clusters έγινε χρησιμοποιώντας την βιβλιοθήκη ploty.express και έχει πιο διαδραστική μορφή.

```
import plotly.express as px

# Δημιουργία DataFrame
df = pd.DataFrame(data_pca, columns=['PC1', 'PC2'])
df['Cluster'] = labels

# Διαδραστικό scatter plot με Plotly
fig = px.scatter(df, x='PC1', y='PC2', color='Cluster', title='Cluster Plot')
fig.show()
```

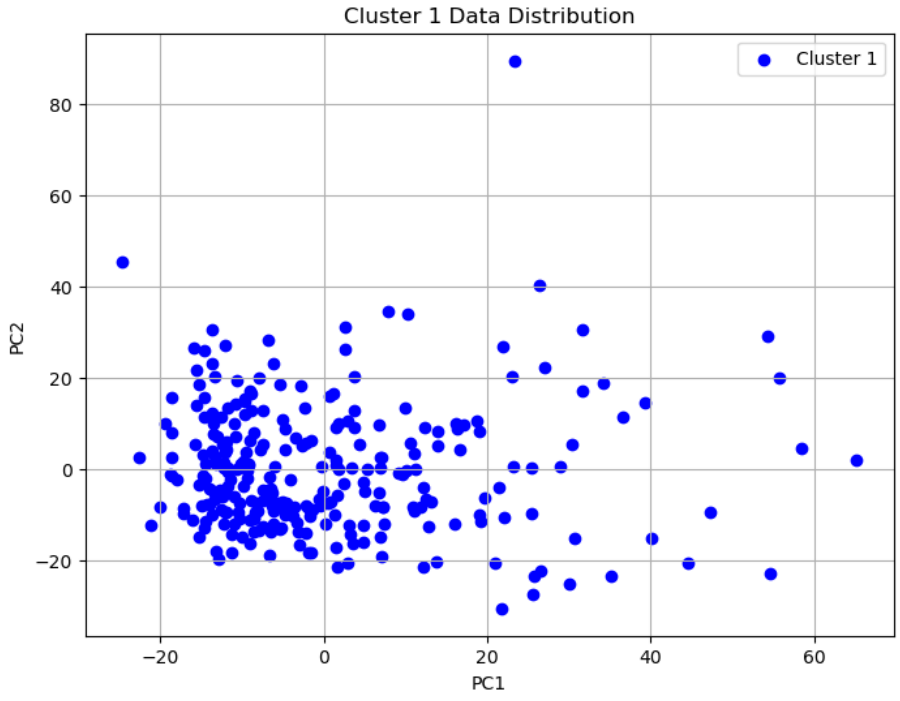


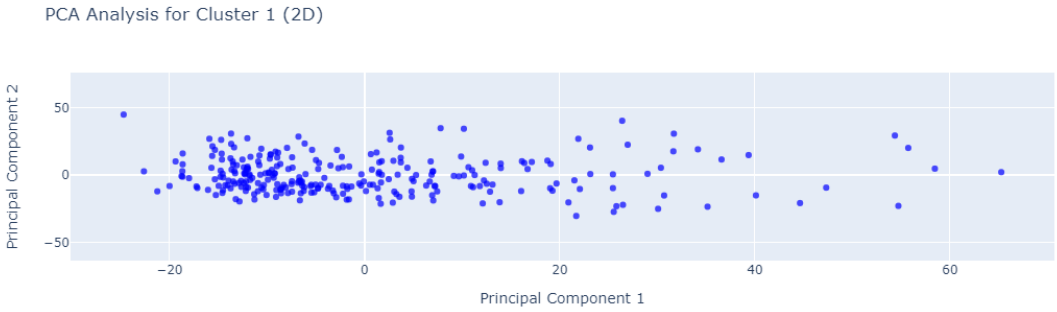
Στην συνέχεια ακολουθούν δύο σχεδιαγράμματα για το πως είναι κατανεμημένοι οι χρήστες στον cluster τους και μερικές συγκρίσεις ανάμεσα τους .

Cluster 0 :

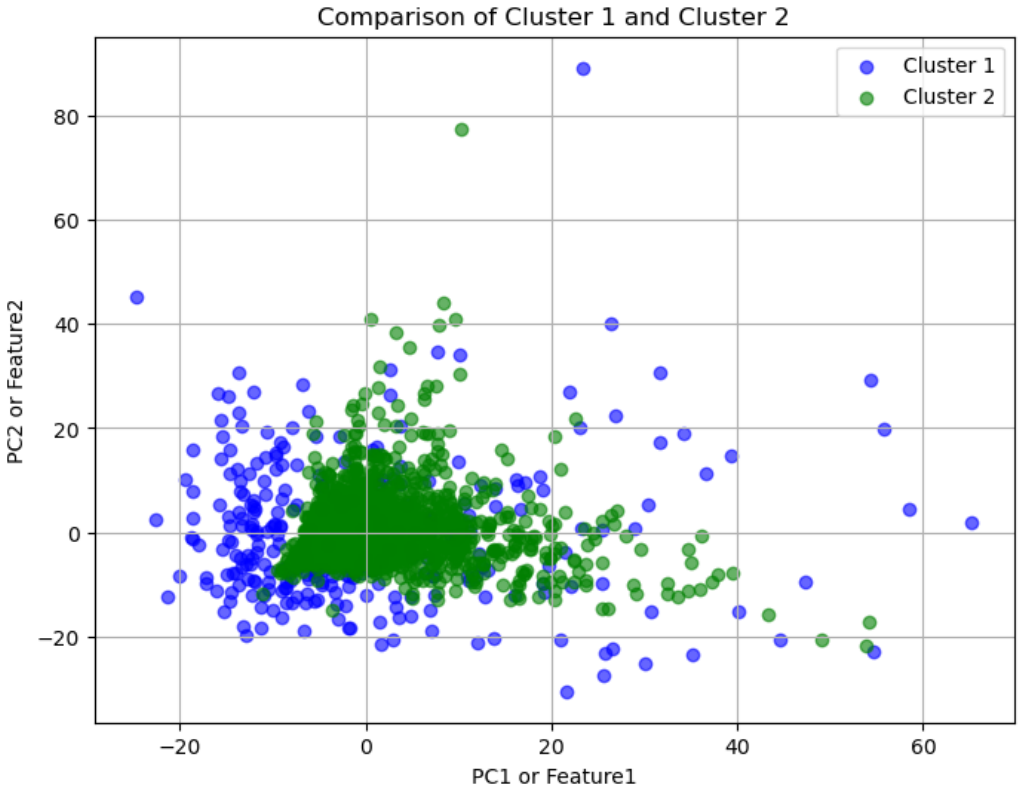


Cluster 1:

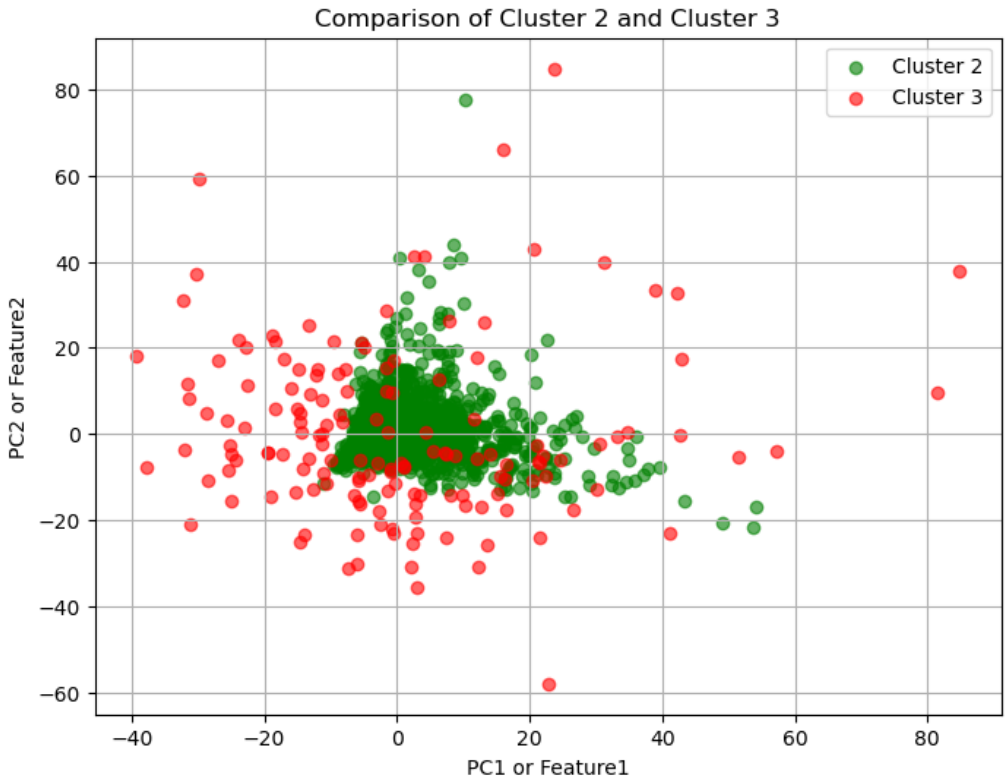




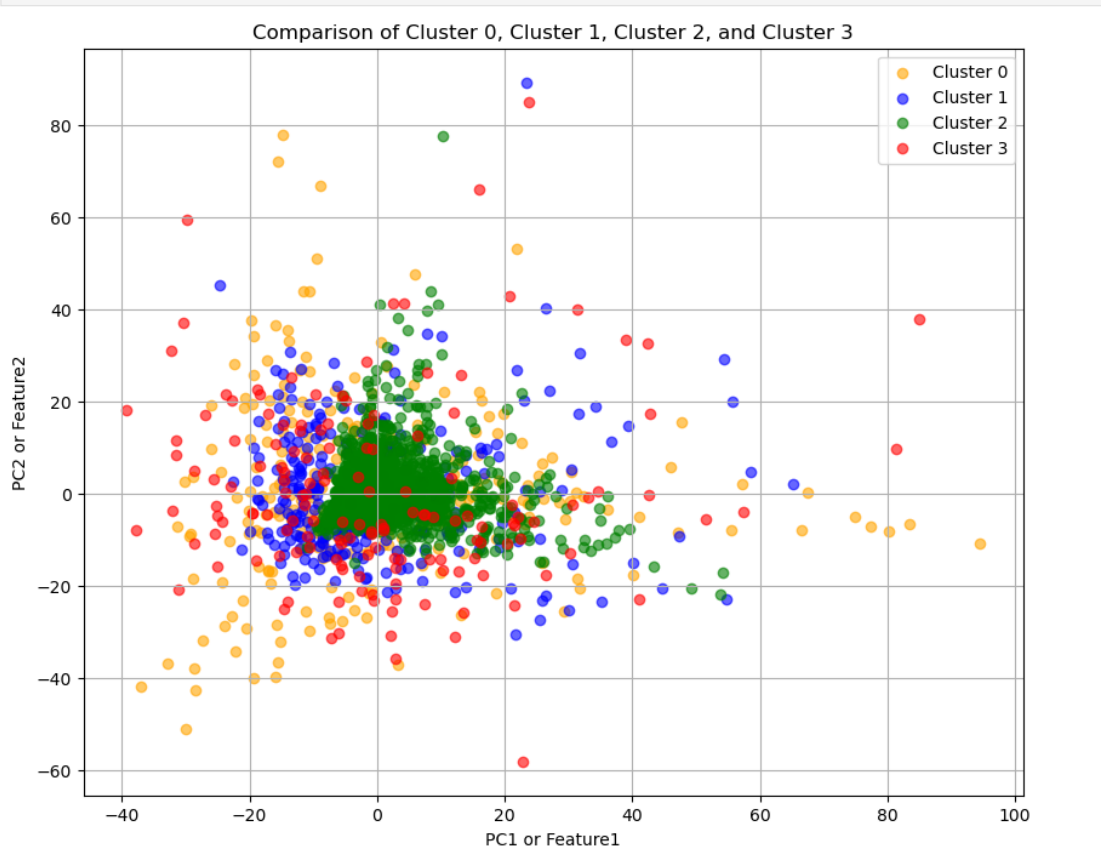
Σύγκριση cluster 1 και 2 :

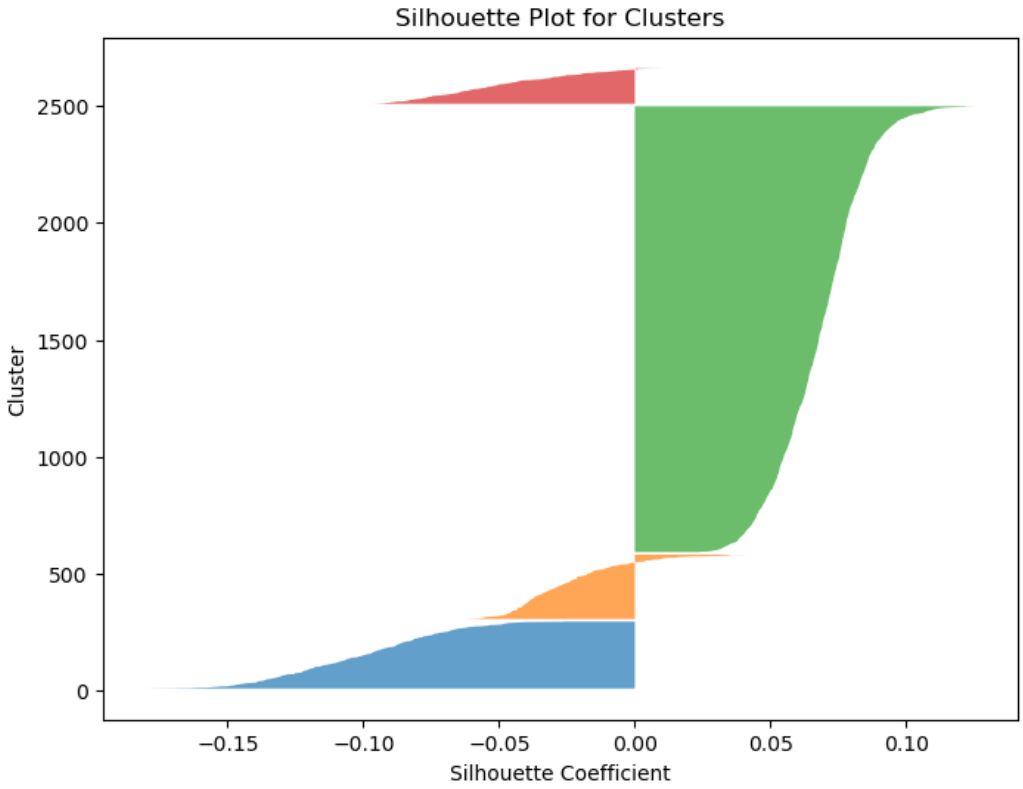


Σύγκριση cluster 2 και 3 :



Γενική σύγκριση σε όλους :





Κεφάλαιο 6 : Συμπεράσματα και μελλοντικές επεκτάσεις

Ο αλγόριθμος K-means, αν και αποτελεσματικός για την ομαδοποίηση δεδομένων, παρουσιάζει ευαισθησία στην αρχική επιλογή των κεντροειδών, γεγονός που μπορεί να οδηγήσει σε τοπικά βέλτιστες λύσεις. Οι μετρικές απόστασης ομοιότητας, όπως η Pearson Correlation, η Cosine Similarity και η Euclidean Distance, παίζουν κρίσιμο ρόλο στην ποιότητα των ομάδων που δημιουργούνται. Κάθε μετρική έχει τα δικά της πλεονεκτήματα και περιορισμούς, καθιστώντας την επιλογή τους κρίσιμη για το αποτέλεσμα. Για τη βελτίωση της απόδοσης του K-means, μπορεί να εφαρμοστεί η τεχνική K-means++ για καλύτερη αρχικοποίηση, καθώς και η κανονικοποίηση των δεδομένων για την εξισορρόπηση της επιρροής των χαρακτηριστικών. Συνολικά, μια πιο στρατηγική προσέγγιση στην επιλογή των μετρικών και των παραμέτρων του αλγορίθμου μπορεί να οδηγήσει σε πιο ακριβή και αξιόπιστα αποτελέσματα.

Βιβλιογραφία :

[1] «A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields» by Hyeyoung , Suyeon Lee, Yoonseo Park and Anna Choi, 3 January 2022 .

[2] «Recommendation systems: Principles, methods and evaluation» by F.O. Isinkaye, Y.O. Folajimi , B.A. Ojokoh. Egyptian Informatics Journal Τόμος 16, Τεύχος 3 , Νοέμβριος 2015, Σελίδες 261-273

[3] «Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities» by Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim and Rasha Kashef, 2 November 2020.

[4] Roy, D., Dutta, M. A systematic review and research perspective on recommender systems. *J Big Data* **9**, 59 (2022).

[5] «Survey on Recommendation System» International Journal of Computer Applications (0975 - 8887) Τόμος 137 - Αρ.7, Μάρτιος 2016 by Lipi Shah, Hetal Gaudani and Prem Balani.

[6] «A Review of Content-Based and Context-Based Recommendation Systems» by Suhuai Luo, Ibrahim A. Hameed, Umair Javed, Farhat Iqbal, Talha Mahboob Alam, (2020).

[7] Gigimol S and Sincy John, «A Survey on different types of recommendation systems» International Research Journal of Advanced Engineering and Science, Τόμος 1, Τεύχος 4, σελ. 111-113, 2016.

[8] «OCA: Ordered Clustering-Based Algorithm for E-Commerce Recommendation System» by Yonis Gulzar, Ali A. Alwan, Radhwan M. Abdullah, Abedallah Zaid Abualkishik and Mohamed Oumrani, 6 February 2023.

[9] «Data Clustering: Algorithms and Its Applications» by Jelili Oyelade , Itunuoluwa Isewon, Olufunke Oladipupo, Onyeka Emebo, Zacchaeus Omogbadegun, Olufemi Aromolaran, Efosa Uwoghiren, Damilare Olaniyan and Obembe Olawole, 2019.

[10] «OCA: Ordered Clustering-Based Algorithm for E-Commerce Recommendation System» by Yonis Gulzar, Ali A. Alwan, Radhwan M. Abdullah, Abedallah Zaid Abualkishik and Mohamed Oumrani, 2023.

[11] «Clustering Techniques: A Brief Survey of Different Clustering Algorithms» by Deepti Sisodia, Lokesh Singh, Lokesh Singh and Khushboo Saxena, 2012.

[12] «Review Paper: A Comparative Study on Partitioning Techniques of Clustering Algorithms» by Gopi Gandhi, Rohit Srivastava, International Journal of Computer Applications (0975 - 8887) Τόμος 87 - Αρ.9, Φεβρουάριος 2014.

[13] «An Overview of Partitioning Algorithms in Clustering Techniques» by Swarndeep Saket J, Dr. Sharnil Pandya, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Τόμος 5, Τεύχος 6, Ιούνιος 2016.

[14] «Comparative Analysis of K-Means and Fuzzy CMeans Algorithms» by Soumi Ghosh, Sanjay Kumar Dubey, ((IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.4, 2013.

[15] «Review based on data clustering algorithms» by Arpita Nagpal, Arnan Jatain, Deepti Gaur, 2013.

[16] «CURE: An Efficient Clustering Algorithm for Large Databases» by Sudipto Guha, Rajeev Rastogi, Kyuseok Shim, 2017.

[17] «CURE: An Efficient Clustering Algorithm for Large Databases» by Guha, Sudipto, Rastogi, Rajeev, Shim, Kyuseok, 1998.

[18] «A study of hierarchical clustering algorithms» by Sakshi Patel, Shivani Sihmar, Aman Jatain, 2015.

[19] «A survey of hierarchical clustering algorithms» by Marjan Kuchaki Rafsanjani, Zahra Asghari Varzaneh, Nasibeh Emami Chukanlo, The Journal of Mathematics and Computer Science Vol .5 No.3 (2012) 229-24.

[20] «Implementation Of ROCK Clustering Algorithm For The Optimization Of Query Searching Time» by Ashwina Tyagi, Sheetal Sharma , 2012.

[21] «Divisive Analysis (DIANA) of hierarchical clustering and GPS data for level of service criteria of urban streets» by Ashish Kumar Patnaik, Prasanta Kumar Bhuyan, K.V. Krishna Rao, Alexandria Engineering Journal Τόμος 55, Τεύχος 1 , Μάρτιος 2016, Σελίδες 407-418.

[22] Sander, J. (2011). Density-Based Clustering. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine.

[23] Deng, Z., Hu, Y., Zhu, M. *et al.* A scalable and fast OPTICS for clustering trajectory big data. *Cluster Comput* 18, 549–562 (2015).

[24] «Data Preprocessing in Data Mining» by Salvador García, Julián Luengo, Francisco Herrera.