



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

DEPARTMENT OF INFORMATICS

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ ΚΑΙ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ 2023-2024

Βαιλάκης Παναγιώτης	ΜΠΣΠ 2305	mpsp2305@unipi.gr
Γκαγκάκης Βασίλης		
Σταυρόπουλος Γεώργιος	ΜΠΣΠ 2347	mpsp2347@unipi.gr

Github repository: https://github.com/gakiasst/ml_2024/blob/main/Anagnorisi.ipynb

Dataset: <https://www.dropbox.com/s/6bcfscyexabgy0w/Dataset.npy?e=2&dl=0>

Στο παρόν έγγραφο παρουσιάζουμε τα αποτελέσματα των αλγορίθμων ομαδοποίησης δεδομένων και της παραγωγής συστάσεων. Ο κώδικας με σχόλια και αποτελέσματα υπάρχει στο παραπάνω github repository.

Προ-επεξεργασία Δεδομένων

1) Να βρείτε το σύνολο των μοναδικών χρηστών U και το σύνολο των μοναδικών αντικειμένων I .

Αρχικά με τα δεδομένα που έχουμε δημιουργούμε το dataframe που θα χρησιμοποιήσουμε.

	UserID	MovieID	Rating	Date
0	4592644	120884	10	2005-01-16
1	3174947	118688	3	2005-01-16
2	3780035	387887	8	2005-01-16
3	4592628	346491	1	2005-01-16
4	3174947	94721	8	2005-01-16
...
4669815	581842	107977	6	2005-01-16
4669816	3174947	103776	8	2005-01-16
4669817	4592639	107423	9	2005-01-16
4669818	4581944	102614	8	2005-01-16
4669819	1162550	325596	7	2005-01-16

4669820 rows × 4 columns

Αφού κατηγοριοποιήσουμε τα δεδομένα με βάση τους χρήστες και τις ταινίες, παρατηρούμε ότι πολλοί χρήστες έχουν βαθμολογήσει πάνω από μια φορά την ίδια ταινία. Για να καθαρίσουμε τα δεδομένα μας, επιλέγουμε να κρατήσουμε την πιο πρόσφατη βαθμολογία και αφαιρούμε τις υπόλοιπες. Φιλτράρουμε τα δεδομένα και καταλήγουμε σε 1.499.238 μοναδικούς χρήστες και 351.109 μοναδικές ταινίες.

2) Να περιορίσετε τα σύνολα των μοναδικών χρηστών U και μοναδικών αντικειμένων I στα αντίστοιχα σύνολα U και I έτσι ώστε $\forall u \in U, R_{min} \leq |qu| \leq R_{max}$ όπου R_{min} και R_{max} ο ελάχιστος απαιτούμενος και ο μέγιστος επιτρεπτός αριθμός αξιολογήσεων ανά χρήστη.

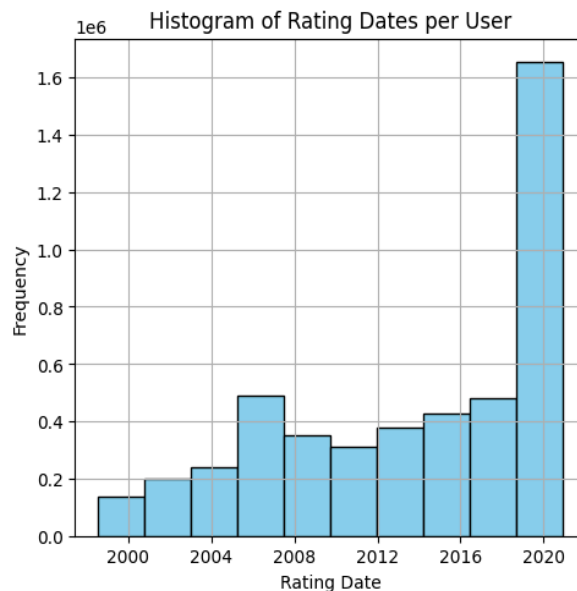
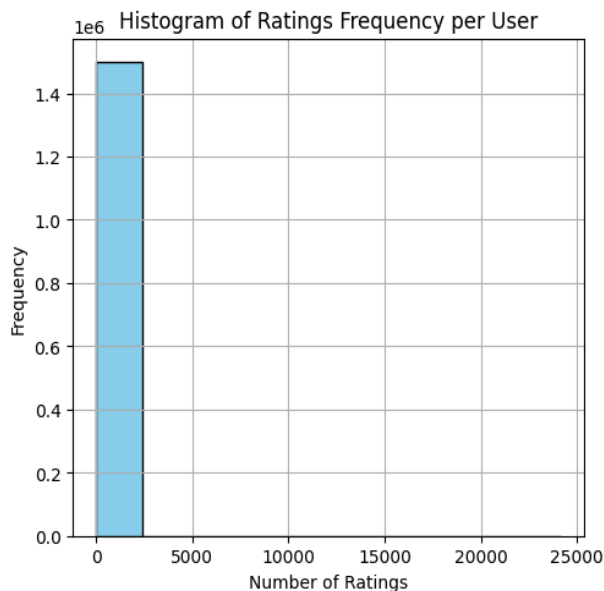
Στην συνέχεια βρίσκουμε τον αριθμό αξιολογήσεων ανά άτομο. Εφόσον υπάρχουν χρήστες με αξιολογήσεις για πολλές ταινίες, περιορίζουμε τα δεδομένα, ώστε να έχουμε τους χρήστες με αριθμό αξιολογήσεων μεταξύ 1000 και 2000. Έτσι καταλήγουμε σε 140 χρήστες με 196.619 αξιολογήσεις.

	UserID	MovieID	Rating	Date
2915231	11762	167260	8	2018-10-19
3520157	11762	2235108	7	2019-10-03
2373413	11762	116778	3	2017-04-20
3520152	11762	442268	8	2019-10-03
1175371	11762	887883	2	2010-05-10
...
3401731	104603847	268126	2	2019-07-24
3352229	104603847	480687	6	2019-06-24
3352227	104603847	910936	10	2019-06-24
3352240	104603847	1139797	7	2019-06-24
3352266	104603847	1170358	6	2019-06-24

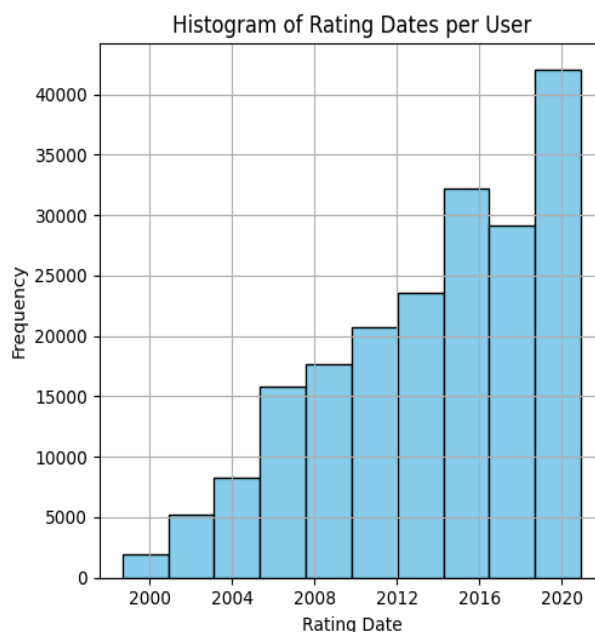
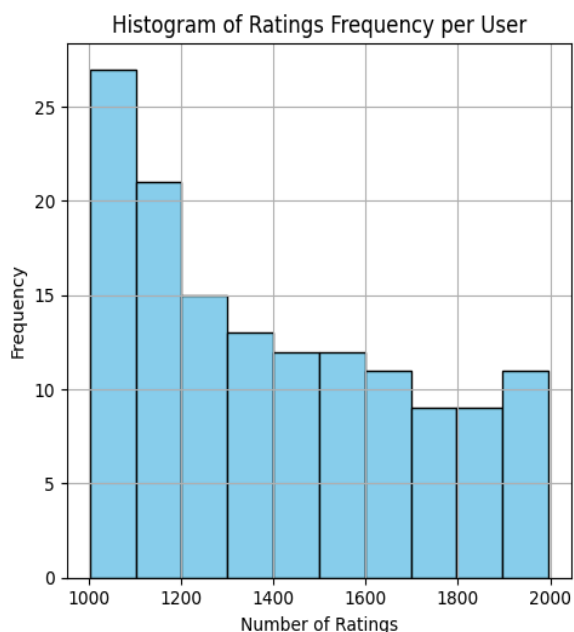
196619 rows × 4 columns

3) Να δημιουργήσετε και να αναπαραστήσετε γραφικά τα ιστογράμματα συχνότητας για το πλήθος αλλά και για το χρονικό εύρος των αξιολογήσεων του κάθε χρήστη.

Εφόσον βρίσκουμε ότι από τους αρχικούς 1.499.238 χρήστες, οι 1.069.533 έχουν μόνο μια βαθμολόγηση, και οι υπόλοιποι 429.705 πάνω από μια, δημιουργούμε τα εξής ιστογράμματα συχνότητας για το πλήθος και το χρονικό εύρος των αξιολογήσεων.



Χρησιμοποιώντας τα δεδομένα του υποσυνόλου των 140 χρηστών που βρήκαμε στο ερώτημα 2, δημιουργούμε τα αντίστοιχα ιστογράμματα συχνότητας για το πλήθος και το χρονικό εύρος των αξιολογήσεων.




4) Δημιουργήστε μια εναλλακτική αναπαράσταση του συνόλου των δεδομένων ως ένα σύνολο διανυσμάτων προτιμήσεων.

Σε αυτό το σημείο θα αναπαραστήσουμε τα δεδομένα ως ένα σύνολο διανυσμάτων προτιμήσεων, σε ένα dataframe με **columns names** τα id των ταινιών και **rownames** τα id των χρηστών.

	1	3	5	10	12	14	41	49	75	91	...	13540912	13540914	13540918	13540920	13556386	13563016	13563480
11762	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
70535	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
102677	0	0	5	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
102816	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
178741	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
...
67430579	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
79950921	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
89333699	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
94289145	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
104603847	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

140 rows × 74571 columns



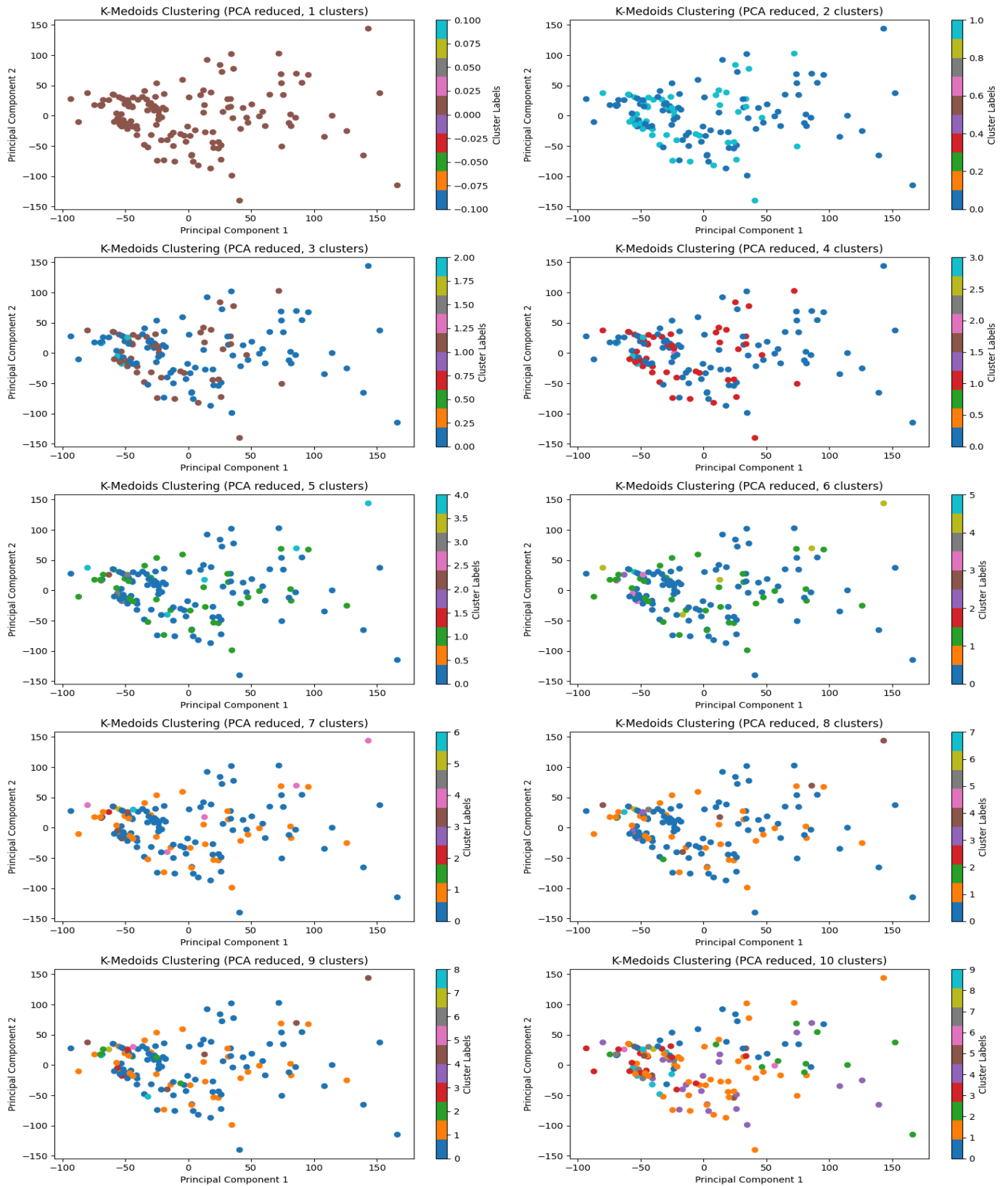
Οι τιμές του κάθε κελιού αναπαριστούν στη βαθμολογία του χρήστη με id το rowname, για την ταινία με id to column name.

Αλγόριθμοι Ομαδοποίησης Δεδομένων

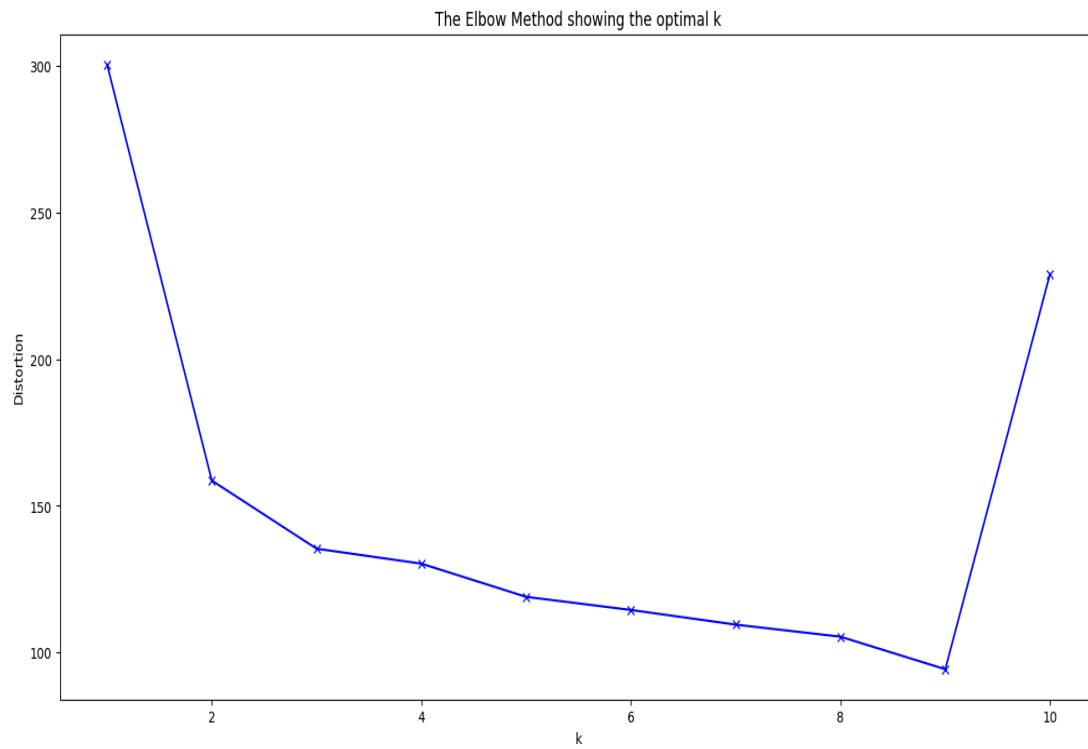
1) Να οργανώσετε το περιορισμένο σύνολο των χρηστών σε συστάδες (clusters) βασιζόμενοι στην διανυσματική αναπαράσταση των προτιμήσεών τους μέσω του συνόλου R .

Για το σκοπό αυτό, χρησιμοποιούμε τον αλγόριθμο **k-means** με τις μετρικές **Euclidean distance** και **cosine distance**. Αρχικά υπολογίζουμε τις αποστάσεις μεταξύ δύο διανυσμάτων για τις δύο μετρικές και στη συνέχεια υπολογίζουμε τις αποστάσεις μεταξύ όλων των διανυσμάτων. Παρακάτω αναπαριστώνται οι συστάδες που αναγνωρίστηκαν:

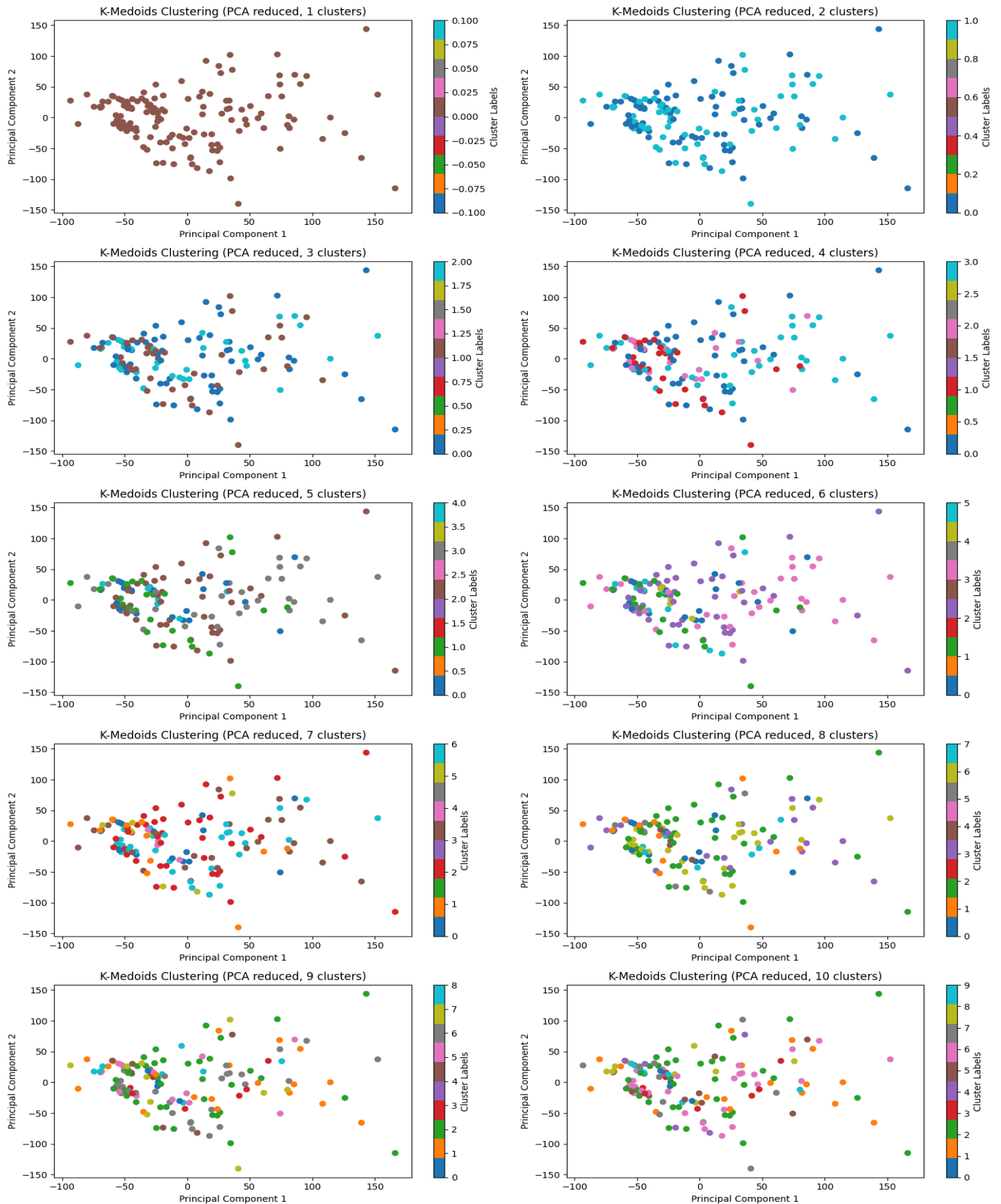
α) για την **Euclidean distance**:



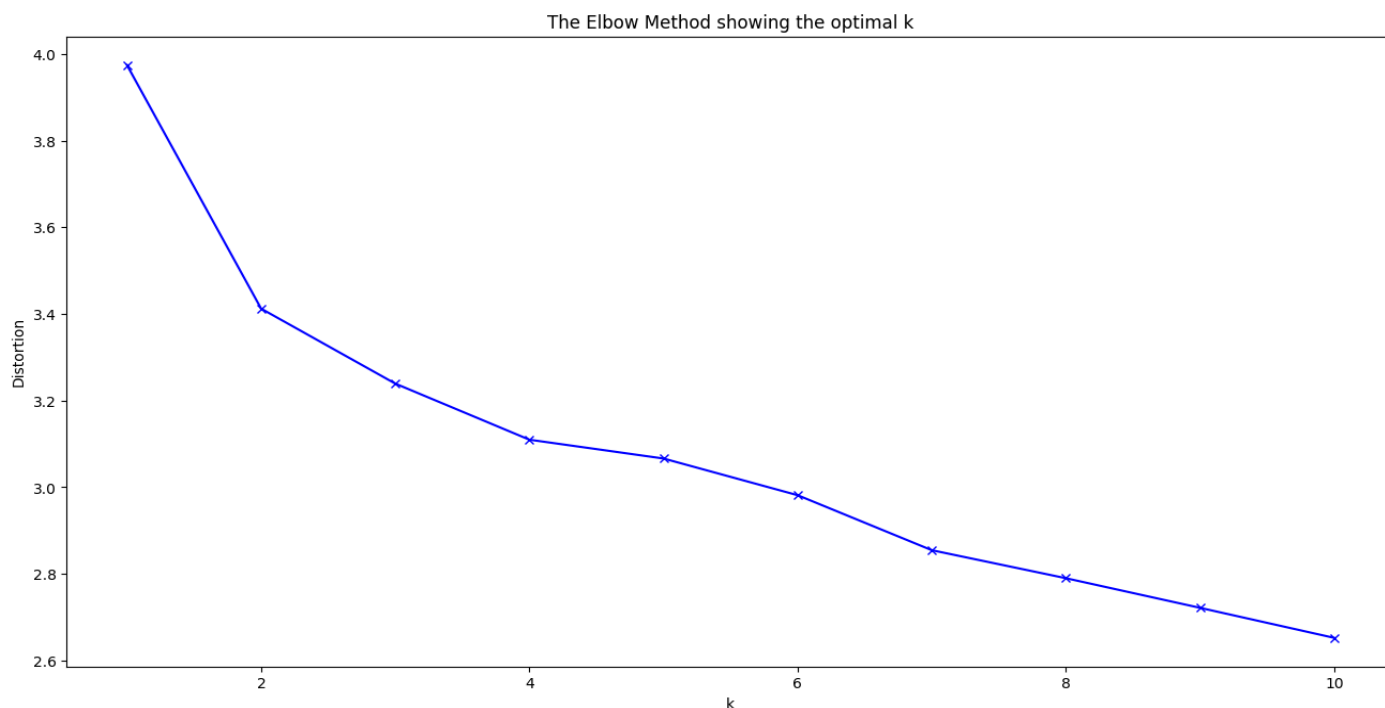
Εφαρμόζουμε τη μέθοδο **Elbow** για να βρούμε τον βέλτιστο αριθμό συστάδων για τον αλγόριθμο **K-Medoids - Euclidean distance** και εμφανίζουμε σε ένα διάγραμμα το αποτέλεσμα:



β) συστάδες που αναγνωρίστηκαν για την μετρική **cosine distance**:



Εφαρμόζουμε τη μέθοδο Elbow για να βρούμε τον βέλτιστο αριθμό συστάδων για τον αλγόριθμο K-Medoids και εμφανίζουμε σε ένα διάγραμμα το αποτέλεσμα.



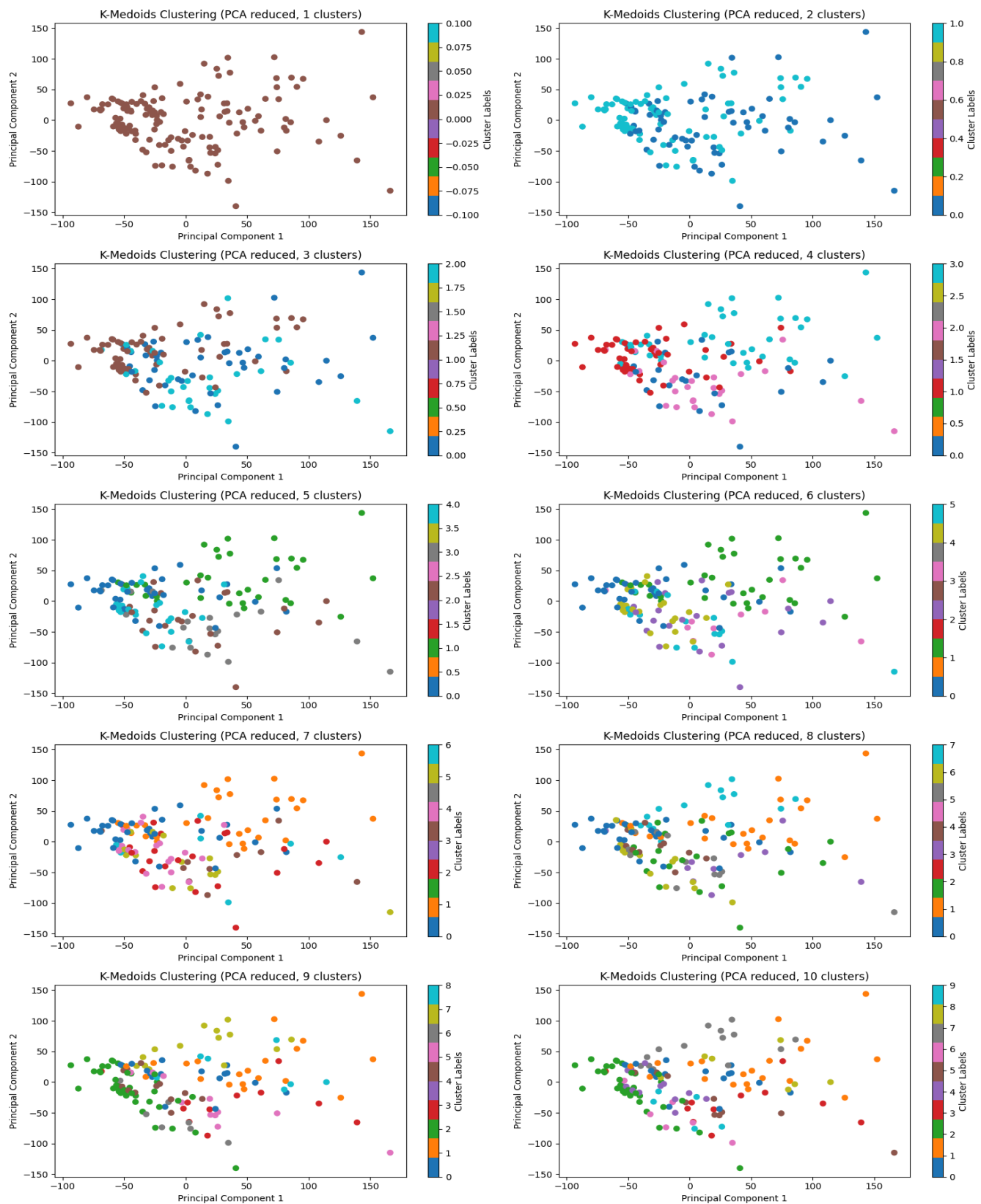
δ) Όσον αφορά την αποτελεσματικότητα των συγκεκριμένων μετρικών στην αποτίμηση της ομοιότητας μεταξύ ενός ζεύγους διανυσμάτων προτιμήσεων χρηστών, το πρόβλημα με την μέτρηση της απόστασης μεταξύ δύο σημείων με συντεταγμένες οι οποίες περιέχουν αριθμούς από το 0 έως το 10 στις οποίες το 0 δεν είναι το μικρότερο προκαλεί τον αλγόριθμο να βρίσκει σημεία τα οποία δεν θα έπρεπε να είναι κοντά μεταξύ τους, να έχουν πολύ μικρή απόσταση και να εισέρχονται στις ίδιες συστάδες. Αυτό συνέβη αρκετές φορές στον κώδικα μας διότι άμα επιλέξουμε χρήστες που έχουν αξιολογήσει πολλές ταινίες τότε ενώ περιορίζουμε τους χρήστες πολύ, δεν περιορίζουμε τα στοιχεία. Σε αυτήν την κατάσταση υπάρχει μεγάλη περίπτωση να έχουμε χρήστες που έχουν κρίνει τελείως διαφορετικές ταινίες να έχουν μηδενική απόσταση. Αυτό θα μπορούσε να λυθεί εάν προσθέσουμε μία πολύ μεγάλη τιμή, όπως το `numpy.inf` που θα μπαίνει όταν οι χρήστες δεν έχουν κανένα κοινό στοιχείο.

Αλγόριθμοι Παραγωγής Συστάσεων με Χρήση Τεχνητών Νευρωνικών Δικτύων

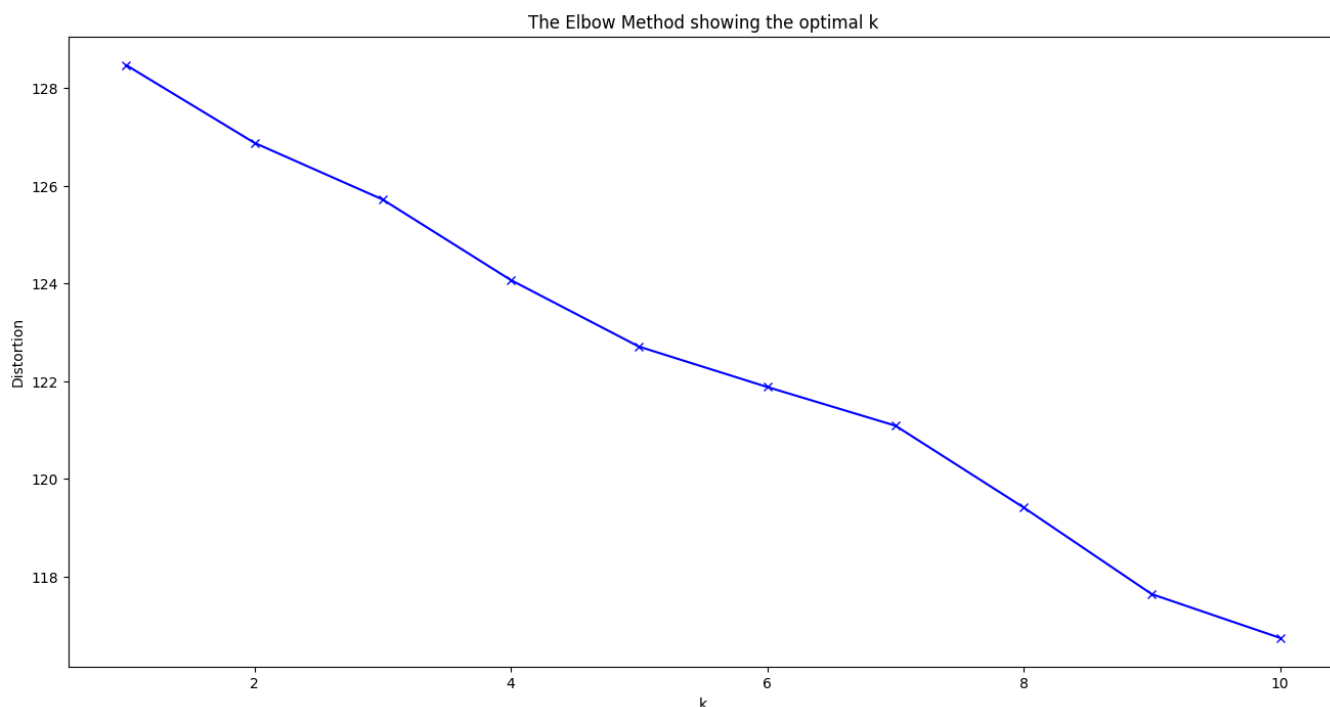
- 2) Να δημιουργήσετε μια εναλλακτική οργάνωση του περιορισμένου συνόλου των χρηστών σε συστάδες, κάνοντας χρήση τις παρακάτω μετρικής (5) $\text{dist}(u,v)=1-|\phi(u)\cap\phi(v)|/|\phi(u)\cup\phi(v)|$

Χρησιμοποιώντας την συνάρτηση `jaccard_distance(u, v)`, υπολογίζουμε την απόσταση μεταξύ δύο διανυσμάτων u και v με βάση τις αξιολογήσεις τους. Στη συνέχεια υπολογίζεται η τομή των συνόλων u_set και v_set και η ένωσή τους. Υπολογίζεται η Jaccard απόσταση ως η αντίστροφη του ποσοστού της τομής προς την ένωση των συνόλων u_set και v_set , όπως φαίνεται στη σχέση (5). Με τις αποστάσεις αυτές δημιουργούμε έναν τετραγωνικό πίνακα, τον οποίο θα χρησιμοποιήσουμε για να επαναλάβουμε την ομαδοποίηση που έγινε παραπάνω με τον αλγόριθμο `kmedoids`.

Στα παρακάτω διαγράμματα βλέπουμε τις συστάδες που αναγνωρίστηκαν με την απόσταση jaccard.



Όπως και νωρίτερα, εφαρμόζουμε τη μέθοδο Elbow για να βρούμε τον βέλτιστο αριθμό συστάδων για τον αλγόριθμο K-Medoids και εμφανίζουμε σε ένα διάγραμμα το αποτέλεσμα:



Η μετρική αυτή προσπαθεί να ομαδοποιήσει τους χρήστες ανάλογα με το ποιές ταινίες έχουν βαθμολογήσει, όσα περισσότερα στοιχεία έχει βαθμολογήσει κάποιος σε σχέση με κάποιον άλλον, τόσο μεγαλύτερη απόσταση θα έχουν. Αντίθετα με τους άλλους αλγορίθμους, δεν κοιτάμε την βαθμολογία, αλλά εαν έχουν αλληλεπιδράσει με την ταινία γενικά, αυτό σημαίνει ότι δύο άτομα που έχουν βάλει βαθμό, ο ένας 1 και ο άλλος 9, σε αυτή την περίπτωση θα μπουν στην ίδια ομάδα. Επειδή όμως υπάρχουν πολλές ταινίες υπάρχει μεγάλη περίπτωση οι χρήστες να έχουν μεγάλη απόσταση και έτσι να γίνει δύσκολο να βλέπουμε ομοιότητες.