

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
Προγραμματιστική Εργασία στο Μάθημα
«Ανακάλυψη Γνώσης από Βάσεις Δεδομένων»

Εξάμηνο: 7^ο



Στοιχεία Φοιτητή
Βαράτης Γεώργιος iis22053

~Ακαδημαϊκό Έτος 2024 - 2025~

Στο παρακάτω έγγραφο αναλύεται η διαδικασία υλοποίησης του συστήματος συστάσεων ταινιών στα πλαίσια της προγραμματιστικής εργασίας του μαθήματος.

Για την υλοποίηση του συστήματος δημιουργήθηκαν τα εξής **3 αρχεία κώδικα** σε γλώσσα **Python** με τις ανάλογες **συναρτήσεις** το κάθε ένα.

utils.py

- *print_progress*:
Παρακολουθεί και εμφανίζει την πρόοδο σε πραγματικό χρόνο (% , χρόνος που έχει παρέλθει/υπολειπόμενος χρόνος) κατά τη διάρκεια μεγάλων εργασιών.
- *train_test_split_ratings*:
Διαχωρίζει τα δεδομένα σε trainsets και testsets, ενώ διασφαλίζει την ύπαρξη χρηστών/ταινιών των συνόλων test στα δεδομένα train.
- *filter_ratings*:
Φιλτράρει το αρχικό dataset ώστε να διατηρεί ταινίες με βαθμολογίες $\geq M$ και χρήστες με βαθμολογίες $\geq M'$. (Για χρήση στο τρίτο πείραμα)
- *create_sparse_matrix*:
Μετατρέπει τα δεδομένα σε έναν αποδοτικό για τη μνήμη αραιό πίνακα στοιχείων χρήστη.
- *compute_similarity*:
Υπολογίζει τις ομοιότητες αντικειμένου-αντικειμένου με χρήση ομοιότητας cosine και του συντελεστή pearson.
- *parallel_prediction*:
Εκτελεί τις προβλέψεις παράλληλα χρησιμοποιώντας κοινή μνήμη για αποδοτικότητα. (Αναπτύχθηκε λόγω μικρής μνήμης RAM για παραλληλοποίηση διαδικασιών ώστε να μην χρειαστεί να γίνει φιλτράρισμα στα πρώτα δύο πειράματα.)
- *calculate_metrics*:
Υπολογίζει μέσο απόλυτο σφάλμα (MAE), μέση ακρίβεια (macro average precision) και μέση ανάκληση (macro average recall), χρησιμοποιώντας threshold συνάφειας ανάλογα με τον χρήστη.

Βασική ροή:

filter_ratings → create_sparse_matrix → compute_similarity → parallel_prediction → calculate_metrics

prediction_functions.py

- *predict*:
Λειτουργεί ως συνάρτηση δρομολόγησης που επιλέγει την κατάλληλη μέθοδο πρόβλεψης με βάση τον επιλεγμένο τύπο στάθμισης. Εάν δεν ζητηθεί προσαρμογή

της δημοτικότητας, χρησιμοποιείται ο βασικός σταθμισμένος μέσος όρος με βάση την ομοιότητα- εάν καθοριστεί «favor popular» ή «favor unpopular», δρομολογείται στην αντίστοιχη συνάρτηση πρόβλεψης με προσαρμογή της δημοτικότητας. Με τον τρόπο αυτό συγκεντρώνεται η λογική της απόφασης και απλοποιούνται οι συγκρίσεις των πειραμάτων.

- ***predict_weighted_average:***

Εφαρμόζει την πρότυπη πρόβλεψη σταθμισμένου μέσου όρου χρησιμοποιώντας τους N πλησιέστερους γείτονες, όπου τα weights βασίζονται αποκλειστικά στην ομοιότητα (Cosine ή Pearson). Υπολογίζει την προβλεπόμενη βαθμολογία ως έναν σταθμισμένο ως προς την ομοιότητα μέσο όρο των βαθμολογιών των γειτόνων, χρησιμεύοντας ως η βασική μέθοδος συνεργατικού φιλτραρίσματος χωρίς καμία πρόσθετη προκατάληψη δημοτικότητας.

- ***predict_weighted_popularity:***

Εφαρμόζει μια πρόβλεψη σταθμισμένου μέσου όρου προσαρμοσμένου στη δημοτικότητα που μπορεί να λειτουργήσει με δύο τρόπους - ευνοώντας δημοφιλή στοιχεία ή ευνοώντας μη δημοφιλή στοιχεία. Όταν έχει οριστεί να ευνοεί το δημοφιλές, η συνάρτηση αυξάνει το weight κάθε γείτονα κατά έναν παράγοντα που αυξάνεται με τον λογάριθμο του αριθμού των αξιολογήσεων που έχει ένα στοιχείο (δηλαδή, τα πιο δημοφιλή στοιχεία έχουν μεγαλύτερη επιρροή). Όταν έχει οριστεί να ευνοεί τα μη δημοφιλή, εφαρμόζει μια αντίστροφη προσαρμογή, έτσι ώστε τα πολύ δημοφιλή στοιχεία να λαμβάνουν χαμηλότερα βάρη, δίνοντας έτσι μεγαλύτερη επιρροή στα λιγότερο συχνά βαθμολογημένα στοιχεία.

RecommendationSystem.py

- ***run_experiment_1:***

Εκτελεί το πρώτο πείραμα εκτελώντας μια σειρά δοκιμών σε $T = 80\%$ (χωρίς φιλτράρισμα) για διαφορετικά μεγέθη γειτονιάς (τιμές N) και για τις δύο μεθόδους ομοιότητας με τις τρεις συναρτήσεις πρόβλεψης. (Συνολικά 6 συστήματα ανα τιμή N)

- ***run_experiment_2:***

Εκτελεί το δεύτερο πείραμα χρησιμοποιώντας το καλύτερο N από το πρώτο πείραμα, και στη συνέχεια συγκρίνει την απόδοση του συστήματος σε διαφορετικές αναλογίες εκπαίδευσης ($T = 50\%$, 70% και 90%) για τα ίδια 6 συστήματα.

- ***run_experiment_3:***

Δοκιμάζει διάφορους συνδυασμούς φιλτραρίσματος (διαφορετικές τιμές M και M') σε $T = 80\%$ χρησιμοποιώντας το καλύτερο N για να παρατηρήσει την επίδραση στην πυκνότητα του πίνακα βαθμολογιών και τη συνολική απόδοση.

- ***Main Execution Block (Argument Parsing):***

Διαβάζει το όρισμα της γραμμής εντολών (-1, -2 ή -3) για να καθορίσει ποιο πείραμα θα εκτελεστεί και, για τα πειράματα 2 και 3, ζητάει από το χρήστη την καλύτερη τιμή N. Επιτρέπει να εκτελείτε οποιοδήποτε πείραμα ανεξάρτητα σε ένα ενιαίο αρχείο, καθιστώντας τη ροή εργασίας ευέλικτη και αρθρωτή.

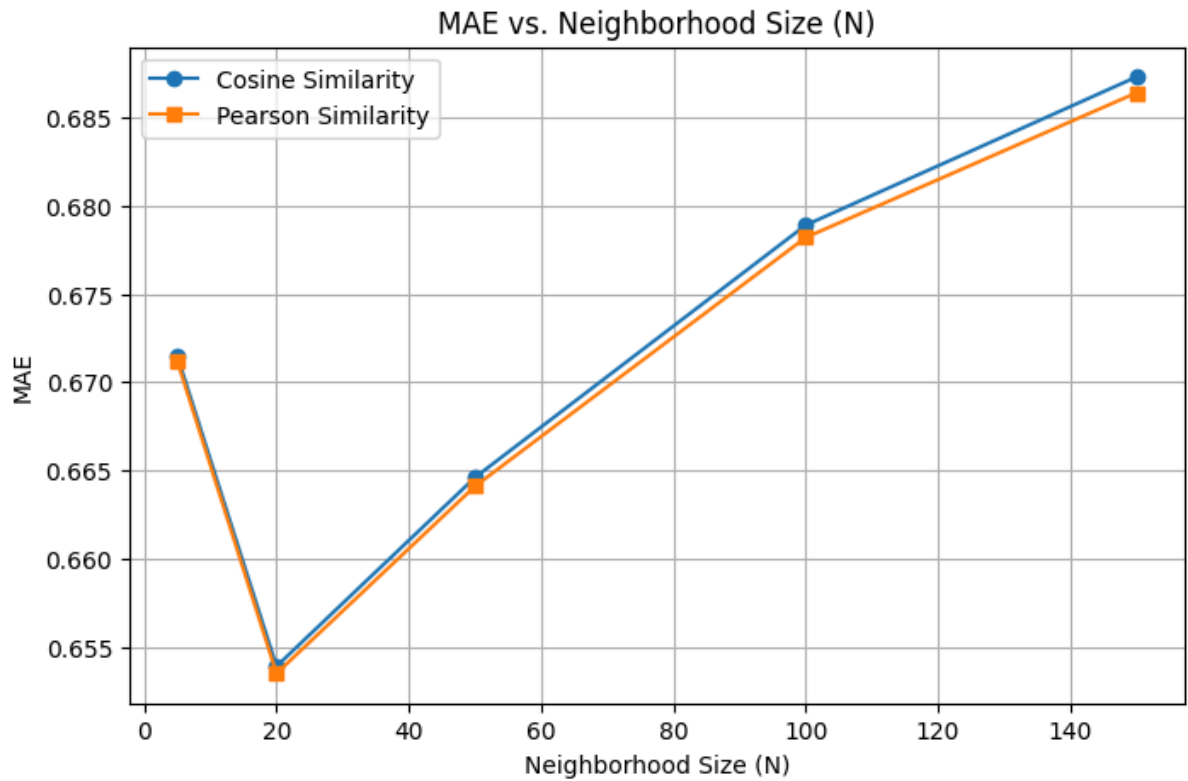
Βιβλιοθήκες που χρησιμοποιήθηκαν		
Βιβλιοθήκη	Σκοπός	Χρήση στον κώδικα
pandas	Χειρισμός δεδομένων (φόρτωση CSV, φιλτράρισμα, pivoting).	Ανάγνωση βαθμολογιών CSV, δημιουργία πινάκων χρηστών-στοιχείων, φιλτράρισμα χρηστών/ταινιών
numpy	Αριθμητικές πράξεις (μαθηματικά συστοιχιών, χειρισμός NaN).	Πράξεις πινάκων ομοιότητας, μετρικοί υπολογισμοί (MAE, precision, recall)
spicy.sparse	Αποδοτική αποθήκευση αραιών πινάκων (βελτιστοποίηση μνήμης).	Αποθήκευση αξιολογήσεων στοιχείων χρηστών σε μορφή csr_matrix
sklearn	Βοηθητικά προγράμματα μηχανικής μάθησης (μετρικές ομοιότητας, διαχωρισμός δεδομένων)	cosine_similarity, train_test_split
multiprocessing	Παράλληλη επεξεργασία για επιτάχυνση της πρόβλεψης με παραλληλοποίηση	shared_memory για κοινή χρήση του πίνακα ομοιότητας
collections.defaultdict	Αποτελεσματική παρακολούθηση των μετρήσεων ανά χρήστη (TP/FP/FN counts)	Αποθήκευση στατιστικών στοιχείων του πίνακα σύγκρισης στο calculate_metrics
time	Παρακολούθηση του χρόνου εκτέλεσης και αναφορά προόδου	print_progress για ενημερώσεις σε πραγματικό χρόνο

Πρώτο πείραμα

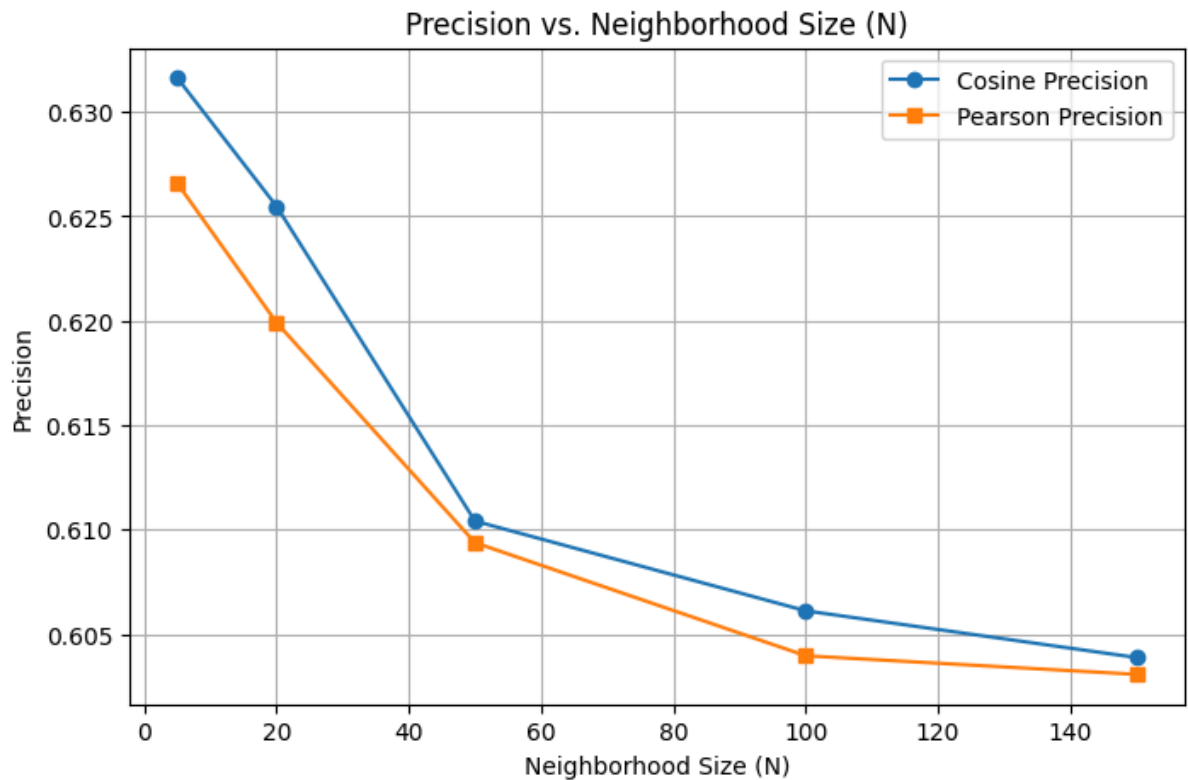
Το **πρώτο πείραμα** υλοποιείται με την χρησιμοποίηση της συνάρτησης **run_experiment_1** του αρχείου **RecommendationSystem.py** όπως αναφέρθηκε και παραπάνω. Για το τρέξιμο αυτού του πειράματος χρησιμοποιήθηκαν αρχικά οι τιμές **N[5,10,15,20,25]** μέχρι που παρατηρήθηκε ότι τα αποτελέσματα των 6 συστημάτων **δεν διέφεραν** οριακά καθόλου. Για αυτόν τον λόγο **ο πίνακας N άλλαξε** σε **N[5, 20, 50, 100, 150]** με την ελπίδα ότι η **διαφορά** των αποτελεσμάτων θα είναι **μεγαλύτερη**. Αυτό που επιτεύχθηκε είναι **λίγο πιο εμφανή αποτελέσματα** για την επιλογή καλύτερου N όπως θα φανεί και παρακάτω. Το πείραμα αυτό έτρεξε **5 φορές** με **μέσο χρόνο εκτέλεσης 13 λεπτά και 30 δευτερόλεπτα** και δημιουργήθηκαν τα εξής **διαγράμματα** από τον μέσο όρο των **αποτελεσμάτων**.

N	Similarity	Favor Popular	MAE	Precision	Recall
5	Cosine	-	0,671539	0,631629	0,678605
5	Cosine	TRUE	0,672188	0,629334	0,687296
5	Cosine	FALSE	0,674124	0,634673	0,667686
5	Pearson	-	0,671192	0,626588	0,681153
5	Pearson	TRUE	0,672051	0,628136	0,691377
5	Pearson	FALSE	0,673606	0,631240	0,671650
20	Cosine	-	0,653862	0,625486	0,720870
20	Cosine	TRUE	0,655067	0,609483	0,740982
20	Cosine	FALSE	0,654289	0,625865	0,679720
20	Pearson	-	0,653478	0,619919	0,719563
20	Pearson	TRUE	0,654967	0,604264	0,743803
20	Pearson	FALSE	0,653946	0,625529	0,684941
50	Cosine	-	0,664583	0,610416	0,734295
50	Cosine	TRUE	0,666864	0,591895	0,773966
50	Cosine	FALSE	0,664029	0,628370	0,670210
50	Pearson	-	0,664054	0,609382	0,739331
50	Pearson	TRUE	0,666720	0,591523	0,780230
50	Pearson	FALSE	0,663202	0,626934	0,683146
100	Cosine	-	0,678904	0,606119	0,743793
100	Cosine	TRUE	0,681602	0,586672	0,788226
100	Cosine	FALSE	0,678102	0,629595	0,665933
100	Pearson	-	0,678181	0,603967	0,747784
100	Pearson	TRUE	0,681352	0,586290	0,792961
100	Pearson	FALSE	0,676977	0,627039	0,681355
150	Cosine	-	0,687340	0,603883	0,745432
150	Cosine	TRUE	0,690148	0,584655	0,793040
150	Cosine	FALSE	0,686386	0,629617	0,662893
150	Pearson	-	0,686450	0,603077	0,750136
150	Pearson	TRUE	0,689725	0,585261	0,797465
150	Pearson	FALSE	0,685063	0,626010	0,678080

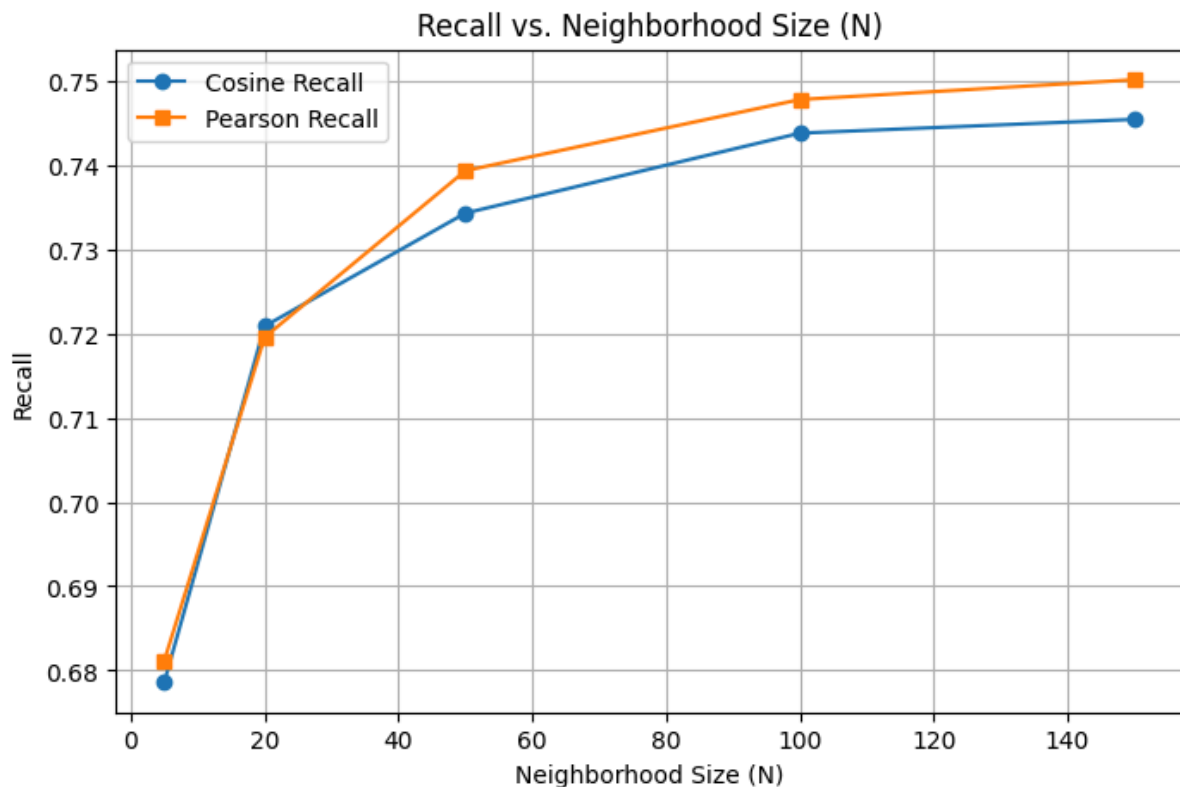
Πίνακας 1: Πίνακας μέσων όρων αποτελεσμάτων πρώτου πειράματος.



Διάγραμμα 1.1: Μέσο απόλυτο σφάλμα (MAE) σε σχέση με το μέγεθος της γειτονιάς (N) για τα μέτρα ομοιότητας Cosine και Pearson.



Διάγραμμα 1.2: Ακρίβεια (Precision) σε σχέση με το μέγεθος της γειτονιάς (N) για τα μέτρα ομοιότητας Cosine και Pearson.



Διάγραμμα 1.3: Ανάκληση (Recall) σε σχέση με το μέγεθος της γειτονιάς (N) για τα μέτρα ομοιότητας Cosine και Pearson.

Συμπεράσματα πρώτου πειράματος

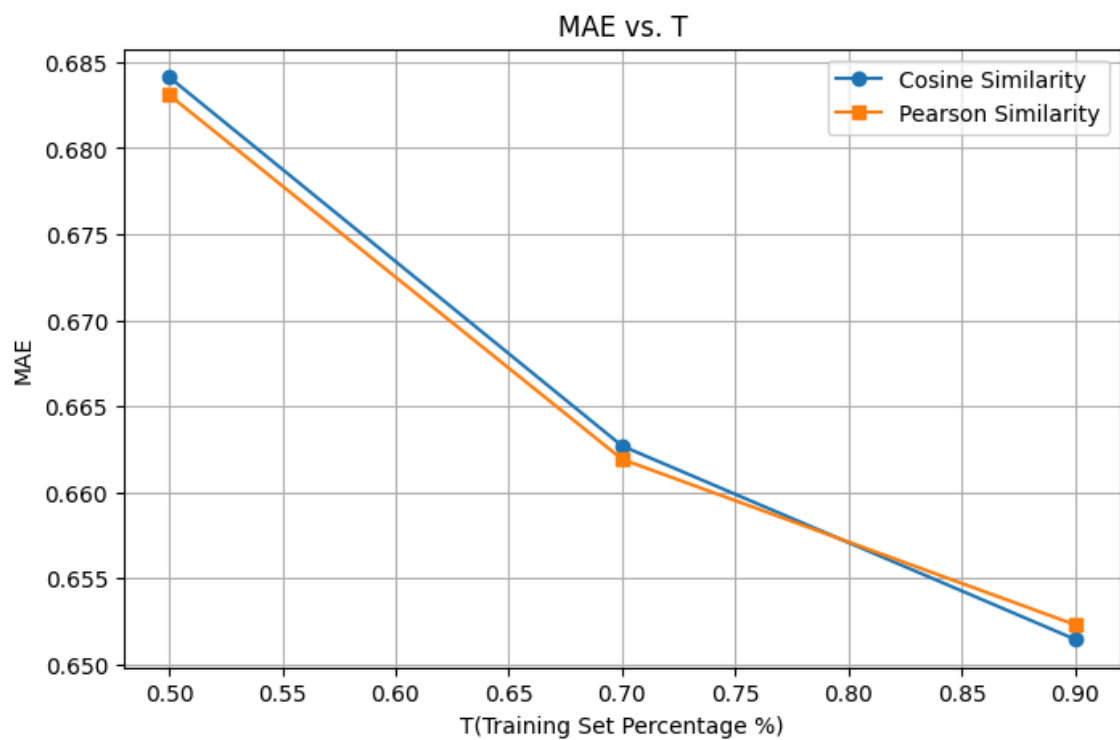
Από τα παραπάνω αποτελέσματα φαίνεται πώς το **N=20** είναι το **καλύτερο μέγεθος** γειτονιάς για αυτό το σύστημα συνεργατικού φιλτραρίσματος. Σε αυτή την τιμή, το **MAE** είναι το **χαμηλότερο (0,653)** τόσο για το Cosine όσο και για το Pearson), υποδεικνύοντας την **υψηλότερη ακρίβεια πρόβλεψης**. Η **ανάκληση (Recall)** **βελτιώνεται** σημαντικά (έως και **0,757** για το Pearson), αν και η **ακρίβεια μειώνεται ελαφρώς** σε σύγκριση με το N=5. Για μεγαλύτερες τιμές από N=20, η απόδοση **μειώνεται** καθώς οι μεγαλύτερες γειτονιές εισάγουν **θόρυβο**, αυξάνοντας το MAE (π.χ. 0,678 στο N=100). Αυτό υποστηρίζει την ιδέα ότι ένα μέτριο N **εξισορροπεί** την **ακρίβεια** και τη **συνάφεια**, καθιστώντας το N=20 τη **βέλτιστη** επιλογή.

Δεύτερο πείραμα

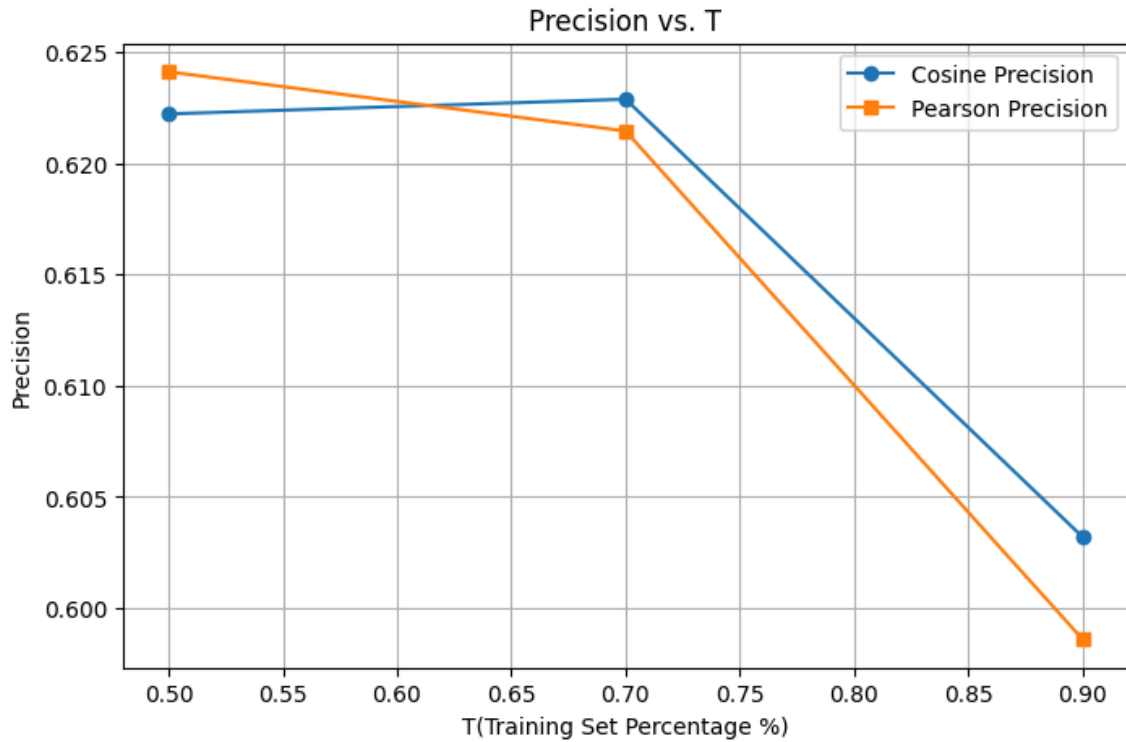
Το **δεύτερο πείραμα** υλοποιείται με την χρησιμοποίηση της συνάρτησης **run_experiment_1** του αρχείου **RecommendationSystem.py** και την **καλύτερη τιμή N = 20** που βρέθηκε στο προηγούμενο πείραμα. Η συνάρτηση αυτού του πειράματος έτρεξε με σταθερό το N και με εναλλασσόμενα σύνολα εκπαίδευσης (trainsets) τα οποία ορίζονται στον πίνακα **T[0.5, 0.7, 0.9]**. Το πρόγραμμα έτρεξε **5 φορές** με τον παρακάτω πίνακα να περιέχει τον **μέσο όρο** των αποτελεσμάτων και **μέσο χρόνο** εκτέλεσης **8 λεπτά και 56 δευτερόλεπτα**. Αναπτύχθηκαν και σχετικά **διαγράμματα** για να αναδειχθεί το **καλύτερο T**.

T	Similarity	Favor Popular	MAE	Precision	Recall
0,5	Cosine	-	0,684135	0,622216	0,67916
0,5	Cosine	TRUE	0,683702	0,612729	0,719059
0,5	Cosine	FALSE	0,687806	0,634723	0,625256
0,5	Pearson	-	0,683095	0,624108	0,685614
0,5	Pearson	TRUE	0,682864	0,610807	0,722994
0,5	Pearson	FALSE	0,686626	0,630218	0,630922
0,7	Cosine	-	0,662682	0,622877	0,710093
0,7	Cosine	TRUE	0,663526	0,60829	0,743898
0,7	Cosine	FALSE	0,663983	0,637598	0,680165
0,7	Pearson	-	0,661905	0,621447	0,720274
0,7	Pearson	TRUE	0,663036	0,605826	0,748452
0,7	Pearson	FALSE	0,66305	0,637963	0,690692
0,9	Cosine	-	0,651434	0,603205	0,686376
0,9	Cosine	TRUE	0,653135	0,594363	0,708947
0,9	Cosine	FALSE	0,651527	0,60552	0,657691
0,9	Pearson	-	0,65228	0,598607	0,690607
0,9	Pearson	TRUE	0,653829	0,594792	0,711146
0,9	Pearson	FALSE	0,652505	0,605535	0,666418

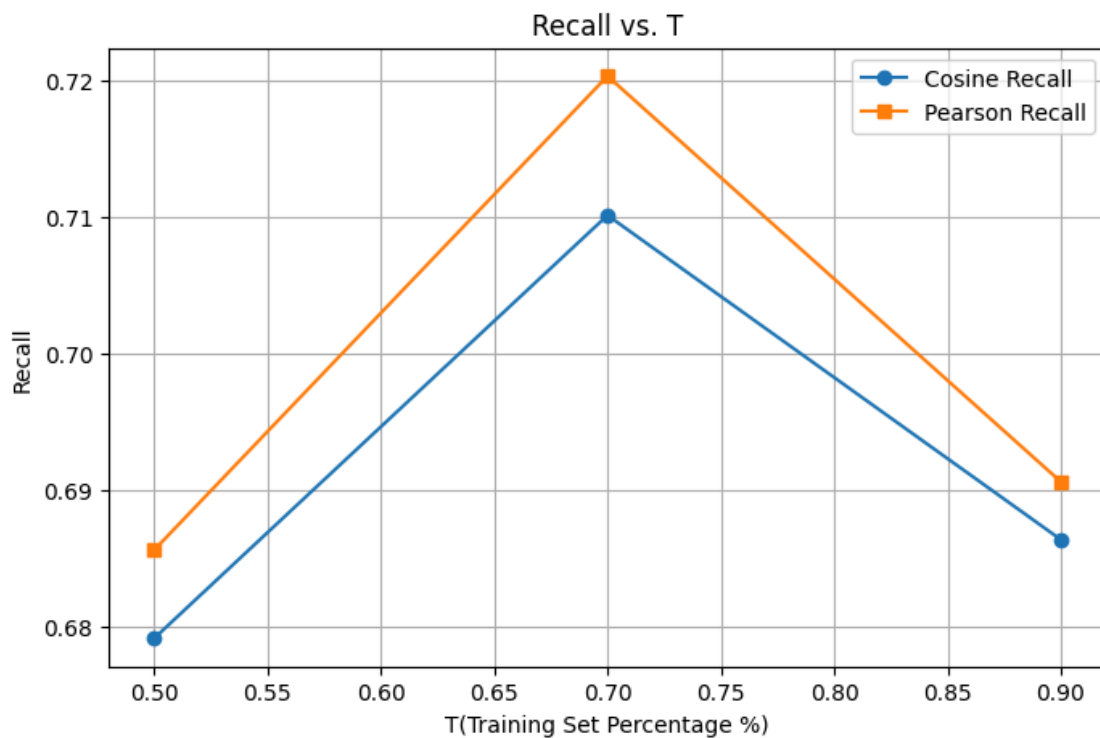
Πίνακας 2: Πίνακας μέσων όρων αποτελεσμάτων δεύτερου πειράματος.



Διάγραμμα 2.1: Μέσο απόλυτο σφάλμα (MAE) σε σχέση με το μέγεθος του συνόλου εκπαίδευσης (T) για τα μέτρα ομοιότητας Cosine και Pearson.



Διάγραμμα 2.2: Ακρίβεια(Precision) σε σχέση με το μέγεθος του συνόλου εκπαίδευσης (T) για τα μέτρα ομοιότητας Cosine και Pearson.



Διάγραμμα 2.3: Ανάκληση (Recall) σε σχέση με το μέγεθος του συνόλου εκπαίδευσης (T) για τα μέτρα ομοιότητας Cosine και Pearson.

Συμπεράσματα δευτέρου πειράματος

Ο καλύτερος διαχωρισμός εκπαίδευσης είναι **T = 70% (0,7)**, επειδή εξισορροπεί την **ακρίβεια** και την **ανάκληση**. Διατηρεί το **MAE** σε **χαμηλά επίπεδα (~0,66)** και δίνει την **υψηλότερη ανάκληση** (0,748 με στάθμιση ομοιότητας Pearson και μεγαλύτερο βάρος στα πιο δημοφιλή αντικείμενα). Ενώ το **T = 90%** **μειώνει** λίγο περισσότερο το **MAE** (~0,65), **βλάπτει** την **ανάκληση**, η οποία είναι **σημαντική** για **καλές συστάσεις**. Το **T = 70%** έχει επίσης **αξιοπρεπή Precision** (~0,60-0,63). Για τα καλύτερα αποτελέσματα, προτείνεται η χρησιμοποίηση της **ομοιότητας Pearson** με μεγαλύτερο βάρος στα πιο **δημοφιλή αντικείμενα** σε **T = 70%**. Αυτή η προσαρμογή δίνει **ακριβείς** και **σχετικές** συστάσεις.

Τρίτο πείραμα

Το **τρίτο πείραμα** υλοποιείται με την χρησιμοποίηση της συνάρτησης **run_experiment_1** του αρχείου **RecommendationSystem.py** και την **καλύτερη τιμή N = 20** που βρέθηκε στο πρώτο πείραμα. Η συνάρτηση του πειράματος έτρεξε με **T = 0.8 (80%)** και **N = 20** και εναλλασσόταν τα **M** και **M'** με τις τιμές του παρακάτω πίνακα [(20, 30), (40, 60), (80, 120)] όπου το πρώτο μέλος κάθε παρένθεσης είναι το **M** και το δεύτερο το **M'**. Το πρόγραμμα έτρεξε **5 φορές** και κρατήθηκαν οι **μέσοι όροι** των αποτελεσμάτων. Παρακάτω θα παρουσιαστούν **ξεχωριστά** τα αποτελέσματα για **κάθε ζευγάρι (M, M')**. Το πρόγραμμα είχε διάρκεια **44 δευτερόλεπτα** κατά μέσο όρο.

M = 20 & M' = 30

M	M'	Similarity	Favor Popular	MAE	Precision	Recall	Density (%)
20	30	cosine	-	0,641561	0,619985	0,738686	8,602%
20	30	cosine	TRUE	0,643812	0,618900	0,762869	8,602%
20	30	cosine	FALSE	0,639909	0,623523	0,714952	8,602%
20	30	pearson	-	0,642398	0,617997	0,759844	8,602%
20	30	pearson	TRUE	0,645776	0,611047	0,779088	8,602%
20	30	pearson	FALSE	0,639764	0,622554	0,739993	8,602%

Πίνακας 3.1: Πίνακας μέσων όρων αποτελεσμάτων τρίτου πειράματος για **M = 20 & M' = 30**.

M = 40 & M' = 60

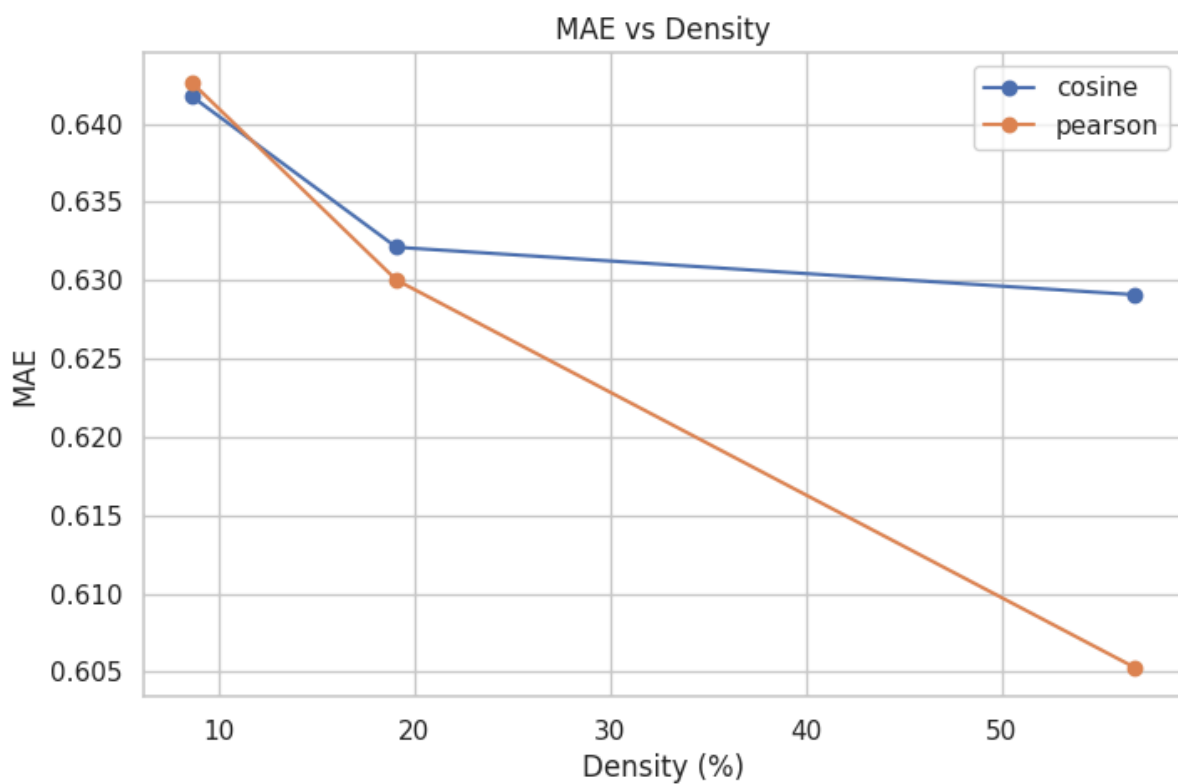
M	M'	Similarity	Favor Popular	MAE	Precision	Recall	Density (%)
40	60	cosine	-	0,631994	0,614749	0,791571	19,072%
40	60	cosine	TRUE	0,634732	0,607775	0,797911	19,072%
40	60	cosine	FALSE	0,629605	0,618135	0,778987	19,072%
40	60	pearson	-	0,629897	0,616117	0,804384	19,072%
40	60	pearson	TRUE	0,633045	0,611834	0,812841	19,072%
40	60	pearson	FALSE	0,627103	0,617202	0,791799	19,072%

Πίνακας 3.2: Πίνακας μέσων όρων αποτελεσμάτων τρίτου πειράματος για **M = 40 & M' = 60**.

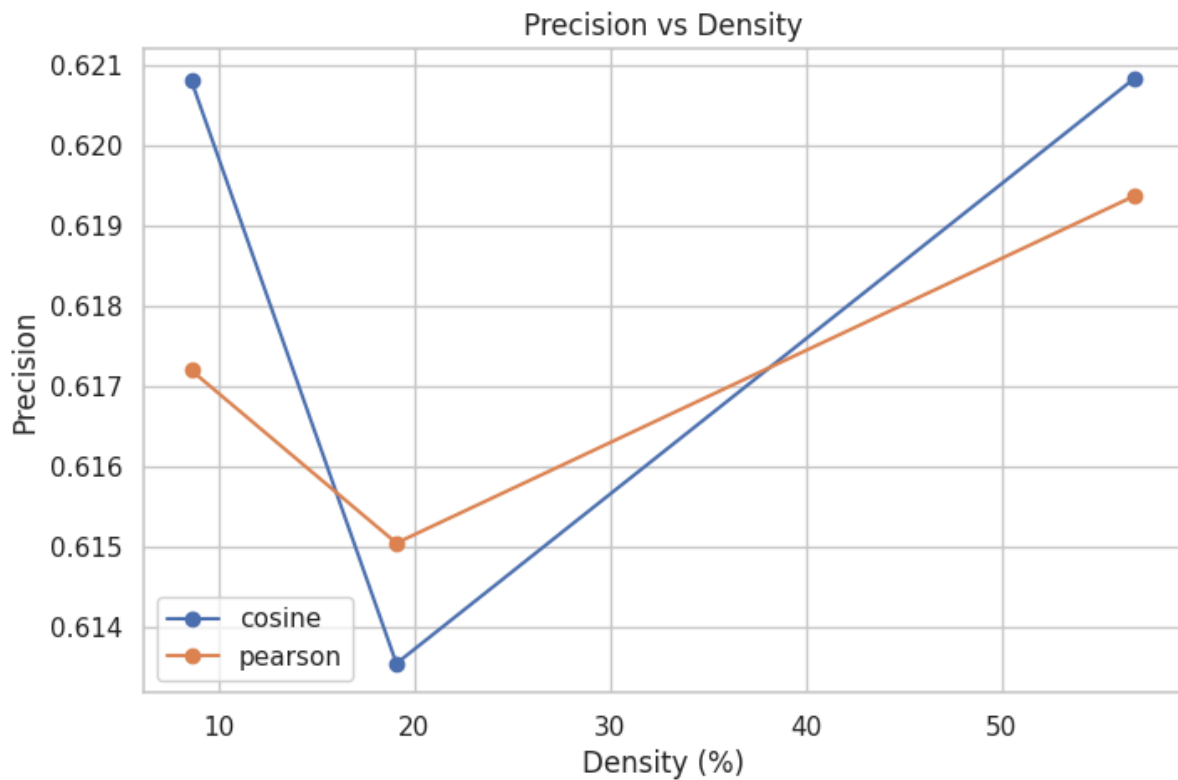
$M = 80$ & $M' = 120$

M	M'	Similarity	Favor Popular	MAE	Precision	Recall	Density (%)
80	120	cosine	-	0,629059	0,619730	0,683260	56,777%
80	120	cosine	TRUE	0,629422	0,622348	0,691106	56,777%
80	120	cosine	FALSE	0,628758	0,620427	0,675613	56,777%
80	120	pearson	-	0,605273	0,618527	0,849472	56,777%
80	120	pearson	TRUE	0,605838	0,616607	0,855722	56,777%
80	120	pearson	FALSE	0,604738	0,622981	0,846347	56,777%

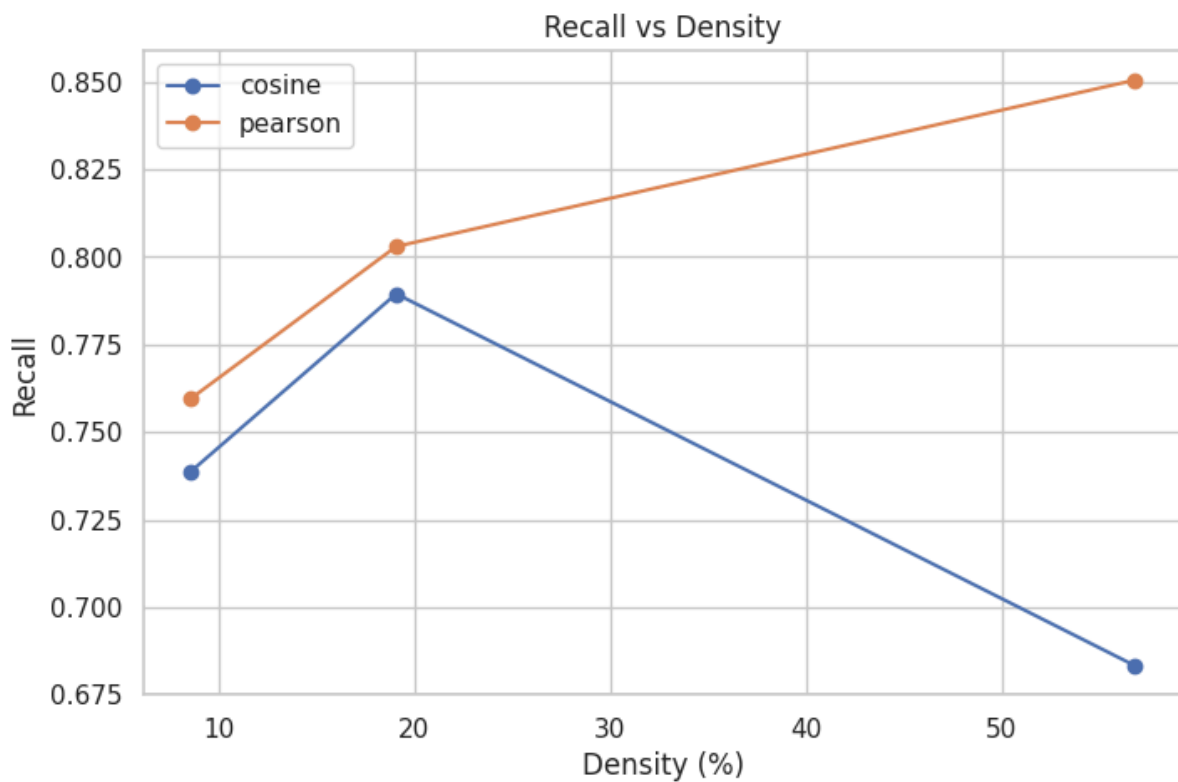
Πίνακας 3.3: Πίνακας μέσων όρων αποτελεσμάτων τρίτου πειράματος για $M = 80$ & $M' = 120$.



Διάγραμμα 3.1: Μέσο απόλυτο σφάλμα (MAE) σε σχέση με την πυκνότητα (Density) για τα μέτρα ομοιότητας Cosine και Pearson.



Διάγραμμα 3.2: Ακρίβεια (*Precision*) σε σχέση με την πυκνότητα (*Density*) για τα μέτρα ομοιότητας *Cosine* και *Pearson*.



Διάγραμμα 3.3: Ανάκληση (*Recall*) σε σχέση με την πυκνότητα (*Density*) για τα μέτρα ομοιότητας *Cosine* και *Pearson*.

Συμπεράσματα τρίτου πειράματος

Η αύξηση των **M** (ελάχιστες βαθμολογίες ταινίας) και **M'** (ελάχιστες βαθμολογίες χρήστη) μείωσε δραστικά τη σπανιότητα του πίνακα, **αυξάνοντας** την **πυκνότητα** από **8,6%** ($M=20$, $M'=30$) σε **56,8%** ($M=80$, $M'=120$). Αυτή η **πυκνότερη δομή** οδήγησε σε:

- **Βελτίωση του MAE** (από $\sim 0,64$ σε $0,605$ για Pearson), καθώς **περισσότερες αλληλεπιδράσεις** χρηστών-ταινιών επιτρέπουν **ακριβέστερες** προβλέψεις.
- **Σημαντική αύξηση του Recall** (από $\sim 0,76$ σε $0,855$ για Pearson), λόγω της **ικανότητας** του συστήματος να **ανακαλύπτει** περισσότερες σχετικές ταινίες σε μεγάλο πλήθος δεδομένων.
- **Σταθερότητα της Precision** ($\sim 0,62$), υποδηλώνοντας ότι η **αναλογία αληθινά σχετικών συστάσεων** διατηρείται **αμετάβλητη** ακόμα και με **πυκνότερους πίνακες**.

Κύριο Συμπέρασμα: Η πυκνότητα ενισχύει την **αξιοπιστία του μοντέλου**, ιδίως για μετρικές όπως η **ομοιότητα Pearson**, που εκμεταλλεύονται συσχετισμούς βαθμολογιών. Ωστόσο, η επιλογή **υψηλών M και M'** πρέπει να εξισορροπείται με τον **κίνδυνο** να **αποκλειστούν** χρήστες/ταινίες με **λιγότερες αλληλεπιδράσεις**, ιδιαίτερα σε **μικρά datasets**. Η βέλτιστη απόδοση επιτυγχάνεται με **M = 80, M' = 120, Pearson, και T=80%**, όπου το σύστημα επιδεικνύει υψηλό Recall ($0,855$) και χαμηλό MAE ($0,605$), ιδανικά για συστάσεις που προτεραιοποιούν την **ολοκληρωμένη κάλυψη σχετικού περιεχομένου**.

Γενικά συμπεράσματα

Από τα τρία πειράματα προκύπτει ότι η **βέλτιστη απόδοση** του συστήματος επιτυγχάνεται με **N=20, T=70%**, και **M=80, M'=120**, σε συνδυασμό με την **ομοιότητα Pearson** και τη **στάθμιση προς δημοφιλή αντικείμενα**. Το μέγεθος γειτονιάς **N=20** εξισορροπεί την **ακρίβεια** (MAE $\sim 0,65$) και την κάλυψη **σχετικών** ταινιών (Recall $\sim 0,75$), **αποφεύγοντας** τον **θόρυβο** μεγαλύτερων γειτονιών. Το **T=70%** διατηρεί **ισορροπία** μεταξύ εκπαίδευσης και ελέγχου, με ικανοποιητικό Recall ($0,748$) και χαμηλό MAE ($\sim 0,66$). Η **αύξηση των M και M'** μείωσε δραστικά τη **σπανιότητα**, **αυξάνοντας** την **πυκνότητα** του πίνακα σε **56,8%**, με αποτέλεσμα την **βελτίωση** του **Recall** έως $0,855$ και τη **μείωση** του **MAE** σε $0,605$ για **Pearson**, χωρίς να θυσιάζεται η Precision ($\sim 0,62$). Η **Pearson επικρατεί σε πυκνά δεδομένα** λόγω της ικανότητάς της να αναγνωρίζει συσχετισμούς βαθμολογιών, ενώ το **Cosine εμφανίζει μικρότερη προσαρμοστικότητα**. Παρά τις βελτιώσεις, η επιλογή υψηλών M και M' μπορεί να **αποκλείει** χρήστες/ταινίες με **λιγότερες αλληλεπιδράσεις**, υπογραμμίζοντας την **ανάγκη ισορροπίας** ανάμεσα σε **πυκνότητα** και **αντιπροσωπευτικότητα δεδομένων**. Συνολικά, το σύστημα με **Pearson, T=70%, N=20, και πυκνό πίνακα (M=80, M'=120)** προσφέρει ακριβείς και **ολοκληρωμένες συστάσεις**, ιδανικές για εφαρμογές που προτεραιοποιούν την **ανακάλυψη σχετικού περιεχομένου**.